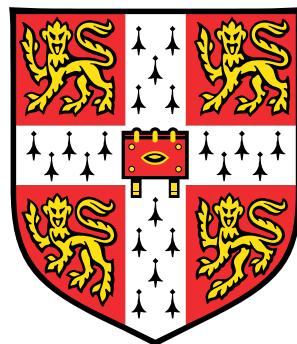


# Multi-scale multiphase modelling of granular flows



**Krishna Kumar Soundararajan**

Department of Engineering  
University of Cambridge

This dissertation is submitted for the degree of

*Doctor of Philosophy*



## Abstract

Geophysical hazards usually involve flow of dense granular solids and water as a single-phase system. Understanding the mechanics of granular flow is of particular importance in predicting the run-out behaviour of debris flows. The dynamics of a homogeneous granular flow involve three distinct scales: the microscopic scale, the meso-scale, and the macroscopic scale. Conventionally, granular flows are modelled as a continuum because they exhibit many collective phenomena. 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. In the present study, a multi-scale approach is adopted, using both DEM and continuum techniques, to better understand the rheology of granular flows and the limitations of continuum models.

The collapse of a granular column on a horizontal surface is a simple case of granular flow, however a proper model that describes the flow dynamics is still lacking. In the present study, the generalised interpolation material point method (GIMPM), a hybrid Eulerian – Lagrangian approach, is implemented with Mohr-Coloumb failure criterion to describe the continuum behaviour of granular flows. The granular column collapse is also simulated using DEM to understand the micro-mechanics of the flow. The limitations of MPM in modelling the flow dynamics are studied by inspecting the energy dissipation mechanisms. The lack of collisional dissipation in the Mohr-Coloumb model results in longer run-out distances for granular flows in dilute regimes (where the mean pressure is low). However, the model is able to capture the rheology of dense granular flows, such as run-out evolution of slopes subjected to impact loading, where the inertial number  $I < 0.1$ .

The initiation and propagation of submarine flows depend mainly on the slope, density, and quantity of the material destabilised. Certain macroscopic models are able to capture simple mechanical behaviours, however the complex physical mechanisms that occur at the grain scale, such as hydrodynamic instabilities and formation of clusters, have largely been ignored. In order to describe the mechanism of submarine granular flows, it is important to consider both the dynamics of the solid phase and the role of the ambient fluid. In the present study, a two-dimensional coupled Lattice Boltzmann LBM – DEM technique is developed

to understand the micro-scale rheology of granular flows in fluid. Parametric analyses are performed to assess the influence of initial configuration, permeability, and slope of the inclined plane on the flow. The effect of hydrodynamic forces on the run-out evolution is analysed by comparing the energy dissipation and flow evolution between dry and immersed conditions.

# Table of contents

<b>Table of contents</b>	<b>5</b>
<b>List of figures</b>	<b>7</b>
<b>List of tables</b>	<b>11</b>
<b>Nomenclature</b>	<b>11</b>
<b>1 Numerical modelling of granular flows</b>	<b>13</b>
1.1 Introduction . . . . .	13
1.2 Continuum modelling of granular flows . . . . .	13
1.3 Material Point Method (MPM) . . . . .	16
1.3.1 Discrete formulation of the governing equations . . . . .	18
1.3.2 Boundary conditions . . . . .	20
1.3.3 Solution scheme . . . . .	22
<b>2 Particulate modelling of granular flows</b>	<b>27</b>
2.1 Introduction . . . . .	27
2.2 Discrete Element Method . . . . .	28
2.2.1 The Forces . . . . .	30
2.2.2 Numerical algorithm and integration scheme . . . . .	34
2.2.3 Boundary conditions . . . . .	41
2.2.4 Particle Assembling Methods . . . . .	43
2.2.5 Voronoi Tesselation . . . . .	46
<b>3 Multi-scale modelling of dry granular flows</b>	<b>47</b>
3.1 Introduction . . . . .	47
3.2 Granular column collapse . . . . .	48
3.2.1 Numerical set-up . . . . .	49

3.2.2	Deposit morphology . . . . .	54
3.2.3	Flow evolution and internal flow structure . . . . .	55
3.2.4	Energy dissipation mechanism . . . . .	64
3.2.5	Role of initial grain properties . . . . .	66
3.3	Slopes subjected to impact loading . . . . .	70
3.3.1	Numerical procedures . . . . .	78
3.3.2	Evolution of pile geometry and run-out . . . . .	79
3.3.3	Decay of kinetic energy . . . . .	81
3.3.4	Effect of friction . . . . .	89
3.3.5	Effect of material points . . . . .	92
3.3.6	Comparison with granular column collapse . . . . .	92
3.4	Summary . . . . .	92
<b>4</b>	<b>Numerical modelling of fluid–grain interactions</b>	<b>101</b>
4.1	Fluid simulation using lattice Boltzmann method . . . . .	101
4.1.1	Formulation . . . . .	103
4.1.2	Lattice Boltzmann - Multi-Relaxation Time (LBM-MRT) . . . . .	107
4.1.3	Boundary conditions . . . . .	110
4.2	Validation of the lattice Boltzmann method . . . . .	113
4.3	Turbulence in lattice Boltzmann method . . . . .	117
4.4	Coupled LBM and DEM for fluid-grain interactions . . . . .	120
4.4.1	Draft, kiss and tumbling: Sedimentation of two grains . . . . .	127
4.5	GP-GPU Implementation . . . . .	130
<b>5</b>	<b>Underwater granular flows</b>	<b>137</b>
5.1	Submarine granular flows down incline plane . . . . .	137
5.1.1	Effect of initial density . . . . .	138
5.1.2	Effect of permeability . . . . .	146
5.1.3	Summary . . . . .	151
<b>References</b>		<b>153</b>

# List of figures

1.1	Multi-scale modelling of granular materials . . . . .	14
1.2	Typical discretization of a domain in MPM . . . . .	17
1.3	Cell crossing noise . . . . .	21
1.4	Illustration of the MPM algorithm for particles occupying a single cell in the background grid . . . . .	24
2.1	Calculation of normal force . . . . .	31
2.2	Verlet list . . . . .	36
2.3	Checking the validity of Verlet list . . . . .	38
2.4	Verlet integration scheme . . . . .	40
2.5	A 2D simulation cell $\omega$ with its basis vectors in an absolute frame . . . . .	42
2.6	(a)Ballistic deposition: first contact followed by steepest descent; (b) small-scale periodic sample (Radjai and Dubois, 2011) . . . . .	45
2.7	Voronoi tessellation to average bulk properties . . . . .	46
3.1	Schematic of experimental configuration for 2-D collapse in a rectangular channel, (Lajeunesse et al., 2004) . . . . .	49
3.2	DEM sample characteristics . . . . .	51
3.3	Periodic shear test . . . . .	53
3.4	Normalised final run-out distance for columns with different initial aspect ratio	56
3.5	Normalised final collapse height for columns with different initial aspect ratio	57
3.6	Velocity profile of a granular column collapse (' $a' = 0.4 \& t = \tau_c$ ) . . . . .	59
3.7	Velocity profile of a granular column collapse (' $a' = 0.4 \& t = 3 \times \tau_c$ ) . . . . .	60
3.8	Velocity profile of a granular column collapse (' $a' = 6 \& t = \tau_c$ ) . . . . .	62
3.9	Velocity profile of a granular column collapse (' $a' = 6 \& t = 3 \times \tau_c$ ) . . . . .	63
3.10	Flow evolution of granular column collapse . . . . .	65
3.11	Energy evolution of granular column collapse . . . . .	67
3.12	Flow evolution of granular column collapse using $\mu(I)$ rheology . . . . .	68

---

3.13	Evolution of inertial number with time for columns with ‘ $a' = 0.4$ and ‘ $a' = 6$	69
3.14	Effect of density on run-out evolution ‘ $a' = 0.8$	71
3.15	Effect of density on energy evolution $a = 0.8$	72
3.16	Evolution of local packing density ‘ $a' = 0.8$	73
3.17	Effect of density on run-out evolution ‘ $a' = 0.8$ (poly-dispersity ‘ $r$ ’ = 6)	74
3.18	Snapshots of granular column collapse $t = 6\tau_c$	75
3.19	Effect of density on energy and packing fraction evolution ‘ $a' = 0.8$ (poly-dispersity ‘ $r$ ’ = 6)	76
3.20	Effect of density on run-out behaviour and energy evolution ‘ $a' = 0.6$	77
3.21	Periodic shear test using CD (?).	80
3.22	Snapshots of MPM simulations of the evolution of granular pile subjected to a gradient impact energy.	82
3.23	Snapshots of CD simulations of the evolution of granular pile subjected to a gradient impact energy (?).	83
3.24	Run-out behaviour of a pile subjected a gradient impact energy	84
3.25	Evolution of kinetic energy with time	86
3.26	Evolution of vertical and horizontal kinetic energy with time	87
3.27	Evolution of kinetic energy in the $x$ component of the velocity field normalized by the available kinetic energy at the end of the transient as a function of normalized time.	88
3.28	Decay time and run-out time as a function of the normalised kinetic energy $E_{kx0}$ .	90
3.29	Effect of friction on the run-out behaviour	91
3.30	Effect of input velocity distribution on the run-out behaviour	93
3.31	Snapshots of MPM simulations of the evolution of granular pile subjected to a gradient impact energy $E_0 = 61 \text{ mgd}$ .	94
3.32	Snapshots of DEM simulations of the evolution of granular pile subjected to a gradient impact energy $E_0 = 61 \text{ mgd}$ .	95
3.33	Evolution of run-out with time for varying material points per cell.	96
3.34	Evolution of kinetic with time for varying material points per cell	97
3.35	Evolution of run-out and duration of flow for varying material points per cell.	98
3.36	Comparison of column collapse with slope subjected to impact loading.	99
4.1	The lattice Boltzmann discretisation and D2Q9 scheme	103
4.2	Illustration of the streaming process on a $D2Q9$ lattice	105
4.3	Illustration of the collision process on a $D2Q9$ lattice	106
4.4	Half-way bounce back algorithm for the $D2Q9$ model	111
4.5	Velocity profile: LBM Simulation of a laminar flow through a channel.	114

4.6	Finite Volume mesh used in the CFD analysis of laminar flow through a channel.	115
4.7	Velocity profile: CFD analysis of laminar flow through a channel. . . . .	115
4.8	Development of the Poiseuille velocity profile in time: comparison between LBM simulation, CFD simulation and the analytical solution. Time is made dimensionless by $H/U_0$ . . . . .	116
4.9	LBM simulation of velocity profile for a laminar flow through a pipe with an obstacle at $L/4$ . . . . .	117
4.10	CFD simulation of velocity contour for a laminar flow through a pipe with an obstacle at $L/4$ . . . . .	117
4.11	LBM and CFD simulation of a flow around an obstacle. . . . .	118
4.12	Kármán vortex street . . . . .	121
4.13	Bounce back boundaries for different values of $\delta$ . . . . .	125
4.14	Time series of draft, kiss and tumble of two grains during sedimentation in a viscous fluid. . . . .	128
4.15	Time history of two circular grains during sedimentation. . . . .	129
4.16	Schematics of a heterogeneous CPU/GPU system. . . . .	131
4.17	GPU scalability with increase in LBM nodes . . . . .	135
5.1	Underwater granular collapse set-up . . . . .	138
5.2	Evolution of run-out with time (dense) . . . . .	139
5.3	Evolution of Kinetic Energy with time (dense case) . . . . .	140
5.4	Flow morphology at critical time for different slope angles (dense) . . . . .	141
5.5	Evolution of run-out with time (loose) . . . . .	142
5.6	Flow morphology at critical time for different slope angles (loose) . . . . .	143
5.7	Evolution of Kinetic Energy with time (loose) . . . . .	144
5.8	Evolution of packing density with time . . . . .	145
5.9	Evolution of run-out with time for different permeability (loose slope 5°) . . .	147
5.10	Evolution of height with time for different permeability (loose slope 5°) . . .	148
5.11	Evolution of Kinetic Energy with time for different permeability (loose slope 5°)	149
5.12	Evolution of packing density with time for different permeability (loose slope 5°)	150



# List of tables

3.1	Material properties of glass ballotini (Lajeunesse et al., 2004) . . . . .	49
3.2	Micro-mechanical parameters used in DEM simulations . . . . .	52
3.3	Parameters used in continuum simulations . . . . .	54
3.4	DEM simulation of simple shear test (?) . . . . .	78
4.1	LBM parameters used in simulating laminar flow through a circular pipe. . .	113
4.2	Computed Strouhal number for fluid flows with different Reynolds number .	122
4.3	GPU vs CPU parallelisation . . . . .	134



# Chapter 1

## Numerical modelling of granular flows

### 1.1 Introduction

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 (see figure 1.1). 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. In the present study, a multi-scale approach is adopted, using both DEM and continuum techniques, to better understand the rheology of granular flows and the limitations of continuum models.

### 1.2 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 the outset, it may even appear for some reasons why such a treatment is objectionable. Most obvious is the fact that the micro-constituents of granular matter, i.e. the individual grains are not small enough to warrant a continuum description (Kamrin et al., 2007). Typical continuum laws are only expected to apply when there is a strong separation of scales, i.e. separation of the micro-scale from the macro-scale, in the flow geometry. Continuum mechanics relies

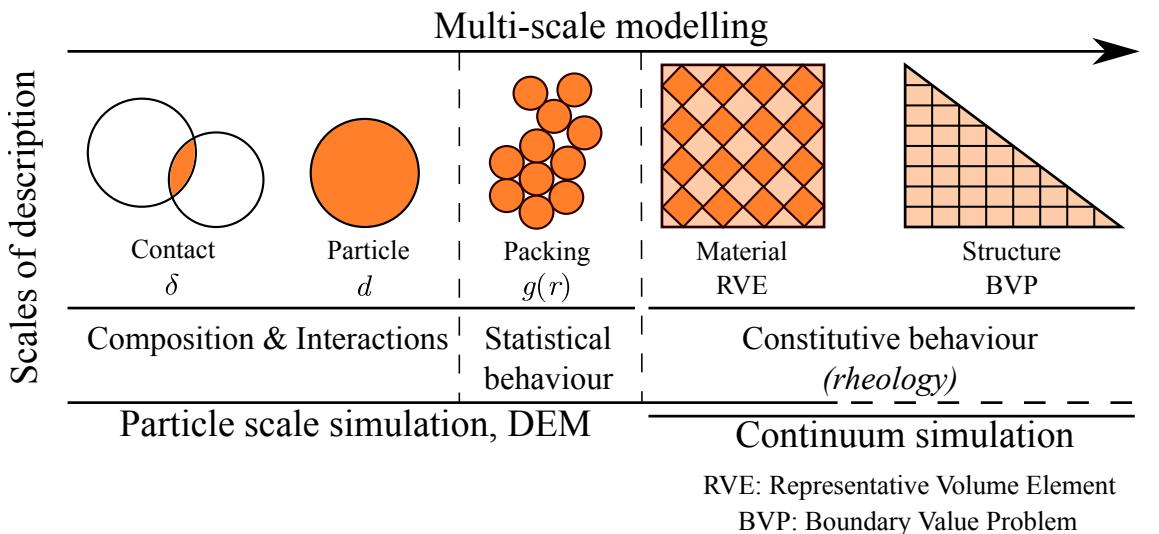


Figure 1.1 Multi-scale modelling of granular materials

1 on the fundamental notion of a representative volume element, in which properties averaged  
 2 over discrete particles exhibit deterministic relationships. Recent works on granular materials  
 3 suggest that a continuum law may be incapable of revealing in-homogeneities at the particle-  
 4 scale level, such as force chains (?). Granular materials exhibit many collective phenomenon  
 5 (Jaeger et al., 1996). However, no continuum model is still capable of describing the parabolic  
 6 flow, the plug flow (Rycroft et al., 2006) and the occurrence of localized shear bands in the  
 7 granular materials. This is in contrast to the hydrodynamics of dilute granular materials and  
 8 the molecular fluids for which accurate continuum models can be systematically derived, by  
 9 averaging over particle collisions in an idealized element (Jenkins and Savage, 1983). The  
 10 fundamental question is how to model granular materials which exhibit complex phenomenon,  
 11 meaningfully.

12 The oldest approach involves modelling the granular material as a rigid solid, which behaves  
 13 as an ideal Coulomb material and undergoes failure if the ratio of the shear stress to the normal  
 14 stress in any plane reaches a critical value of the Coulomb internal friction coefficient  $\mu$ .  
 15 The stress is determined based on the mechanical equilibrium of the system along with the  
 16 hypothesis of *incipient yield*, i.e. the yield criterion is attained everywhere at all times. In limit-  
 17 state Mohr-Coulomb plasticity, these conditions are assumed to hold even if the wall allows  
 18 for a plastic yielding, due to the assumption of coaxiality (?). The fundamental assumption  
 19 of a limit-state stress field at incipient yield everywhere is questionable. Granular flows can  
 20 contain regions lying within the yield surface. For example, in the case of a granular column  
 21 collapse the central cone remains stagnant, and thus cannot be considered as yielded. In fact,  
 22 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 *critical state* theories 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.

Granular materials are composed of distinct grains, which displace independently from one another and interact only at contact points. It is assumed that the deformations of individual grains are negligible in comparison with the deformation of the granular assembly as a whole. The latter deformation is primarily due to the movement of the grains as a rigid body. Therefore, it can be argued that precise modelling of particle deformation is not necessary to obtain a good approximation of the overall mechanical behaviour. An Eulerian grain-level continuum model describes the response of individual grains to the applied loads. However, continuum mechanics solves over the whole domain using initial and boundary conditions appropriate for the problem. Hence, continuum models are still widely used to solve engineering problems associated with granular materials and flows. Conventional mesh based approaches, such as Finite Element Method or Finite Difference Method involves complex re-meshing and remapping of variables, which causes additional errors in simulating large deformation problems (Li and Liu, 2002). Mesh free methods such as Material Point Method are not constrained by the mesh size and

- <sup>1</sup> mesh distortion, and hence can be effectively used in simulating large deformation problems,  
<sup>2</sup> such as debris flow and submarine landslides.

### <sup>3</sup> 1.3 Material Point Method (MPM)

<sup>4</sup> Material Point Method (MPM) ([Sulsky et al., 1994, 1995](#)) is a particle based method that  
<sup>5</sup> represents the material as a collection of *material points*, and their deformations are deter-  
<sup>6</sup> mined by *Newton's laws of motion*. [Sulsky et al. \(1994\)](#) extended the Particle-in-Cell (PIC)  
<sup>7</sup> method ([Harlow, 1964](#)) to computational solid mechanics by taking advantage of the combined  
<sup>8</sup> Eulerian-Lagrangian approach. Material Point Method is a hybrid Eulerian-Lagrangian ap-  
<sup>9</sup> proach, which uses moving material points, and computational nodes on a background mesh.  
<sup>10</sup> This approach is very effective particularly in the context of large deformations (??). Although,  
<sup>11</sup> not derived directly from what is classically considered as mesh-free or mesh-less methods,  
<sup>12</sup> MPM is still considered as a mesh-free approach, primarily because the initial discretization of  
<sup>13</sup> the material does not involve a polygonal tessellation, as in Finite Element Method. However,  
<sup>14</sup> MPM utilizes a background mesh to perform differentiation, integration, and to solve equations  
<sup>15</sup> of motions ([Steffen et al., 2008](#)). The background mesh can be of any form, however for  
<sup>16</sup> computational efficiency a Cartesian lattice is adopted. A typical 2D discretization of a solid  
<sup>17</sup> body is shown in figure 1.2.

<sup>18</sup> The grey-shaded circles are the material points ( $X_p$ , where 'p' represents a material point)  
<sup>19</sup> and the computational nodes are the points of intersection of the grid (denoted as  $X_i$ , where  
<sup>20</sup>  $i$  represents a computational node). The Material Point Method involves discretizing the  
<sup>21</sup> domain  $\Omega$  with a set of material points. The material points are assigned an initial value of  
<sup>22</sup> position, velocity, mass, volume, and stress denoted as  $\mathbf{x}_p$ ,  $v_p$ ,  $m_p$ ,  $\mathbf{V}_p$  and  $\sigma_p$ . Depending on  
<sup>23</sup> the material being simulated, additional parameters, like pressure, temperature, etc., should be  
<sup>24</sup> specified at the material points. The material points are assumed to be within the computational  
<sup>25</sup> grid, which is assumed to be a Cartesian lattice for convenience (see figure 1.2). At every  
<sup>26</sup> time step  $t_k$ , the MPM computation cycle involves projecting the data, such as position, mass,  
<sup>27</sup> and velocity from the material points to the computational grid using standard nodal basis  
<sup>28</sup> functions, called *shape functions*, derived based on the position of particle with respect to the  
<sup>29</sup> grid. Gradient terms are calculated in the computational grid and the governing equation, i.e.  
<sup>1</sup> the equation of motion, is solved and the updated position and velocity values are mapped back  
<sup>2</sup> to the material points. The mesh is reinitialized to its original state and the computational cycle  
<sup>3</sup> is repeated.

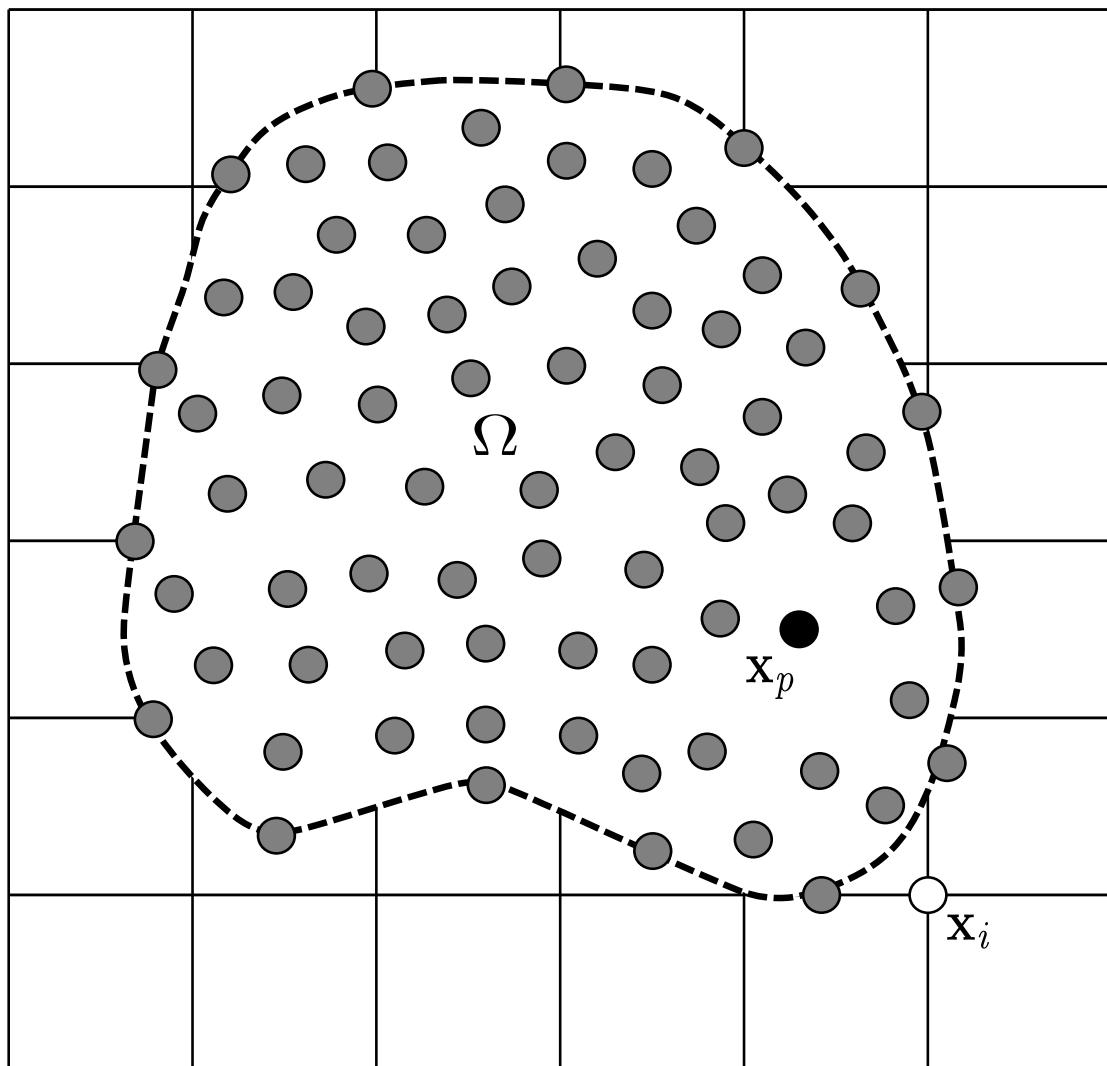


Figure 1.2 Typical discretization of a domain in MPM. The dotted line represents the boundary of the simulated object  $\Omega$  and each closed point represents a material point used to discretize  $\Omega$ . The square mesh represents the background grid. Each square in the background grid is a grid cell, and grid nodes are located at the corners of grid cells.

#### 4 1.3.1 Discrete formulation of the governing equations

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

7

$$\frac{\partial \rho}{\partial t} \rho \Delta \cdot v = 0 \quad (1.1)$$

8 and

9

$$\rho a = \Delta \cdot \sigma + \rho b \quad (1.2)$$

10

11 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  
 12 the Cauchy's stress tensor, and  $\mathbf{b}(\mathbf{x}, t)$  is the body force. The vector  $\mathbf{x}$  represents the current  
 13 position of any material point in the continuum, at time  $t$ . MPM discretizes the continuum body  
 14 into finite number of material points  $N_p$ . Let  $\mathbf{x}_p^t$  ( $p = 1, 2, \dots, N_p$ ) denote the current position of  
 15 material point  $p$  at time  $t$ . Each material point, at any given time  $t$ , has an associated mass  $m_p^t$ ,  
 16 density  $\rho_p^t$ , velocity  $v_p^t$ , Cauchy stress tensor  $\sigma_p^t$ , strain  $\epsilon_p^t$ , and other necessary internal state  
 17 variables based on the adopted constitutive model. These material points provide a Lagrangian  
 18 description of the continuum body, since material points have a fixed mass at all times, eq. 1.1  
 19 is satisfied. The data from the material points are mapped on to the nodes of the computational  
 20 grid, where the discrete form of eq. 1.2 is described. The weak form of eq. 1.2 is obtained by  
 21 multiplying eq. 1.2 with a test function  $w(\mathbf{x}, t)$ .

22

$$\int_{\Omega} \rho \mathbf{w} \cdot a d\Omega = - \int_{\Omega} \rho \sigma^s : \Delta w d\Omega + \int_{\partial\Omega_t} w \cdot \tau dS + \int_{\Omega} \rho w \cdot \mathbf{b} d\Omega \quad (1.3)$$

23

24 where  $\sigma^s$  is the specific stress (i.e. stress divided by mass density,  $\sigma^s = \sigma/\rho$ ),  $\Omega$  is the current  
 25 configuration of the continuum,  $\tau$  is the traction. eq. 1.3 is obtained by applying the divergence  
 26 theorem, similar to the standard procedure adopted in Finite Element Methods (Chen and  
 27 Brannon, 2002; Sulsky et al., 1994, 1995). The differential volume and the surface elements  
 28 are denoted by  $d\Omega$  and  $dS$ , respectively.

As the whole continuum is discretized into a finite set of material points, the mass density can  
 1  
 2 be written as:

3

$$\rho(\mathbf{x}, t) = \sum_{p=1}^{N_p} M_p \delta(x - x_p^t) \quad (1.4)$$

4

where  $\delta$  is the Dirac delta function. Substituting [eq. 1.4](#) in [eq. 1.3](#), the sum of quantities of material points can be evaluated as:

$$\sum_{p=1}^{N_p} M_p [w(x_p^t, t) \cdot \mathbf{a}(x_p^t, t)] = \sum_{p=1}^{N_p} M_p [-\boldsymbol{\sigma}^s(x_p^t, t) : \Delta w|_{x_p^t} + w(x_p^t, t) \cdot \boldsymbol{\tau}^s(x_p^t, t) h^{-1} + w(x_p^t, t) \cdot b(x_p^t, t)] \quad (1.5)$$

where  $h$  is the thickness of the boundary layer. It can be noted from [eq. 1.5](#) that the interactions between different material points are reflected only through the gradient terms. In MPM, a background computational mesh is used to calculate the gradient terms. The computational mesh is constructed using 2-node cells for 1-D, 4-node cells for 2-D, and 8-node cells for 3-D problems. These elements are used to define the standard nodal basis functions,  $N_i(\mathbf{x})$ , associated with the spatial nodes  $x_i(t), i = 1, 2, \dots, N_n$ , where  $N_n$  represents the total number of mesh nodes. The nodal basis functions are assembled by using the conventional finite-element shape functions ([Chen and Brannon, 2002](#)). The co-ordinates of any material point in a cell can be represented by

$$x_p^t = \sum_{i=1}^{N_n} x_i^t N_i(\mathbf{x}_p^t) \quad (1.6)$$

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:

$$w_p^t = \sum_{i=1}^{N_n} w_i^t N_i(\mathbf{w}_p^t) \quad (1.7)$$

[eq. 1.6](#) and [1.7](#) 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. 1.6](#) and [1.7](#) into [eq. 1.5](#), the weak form of the equation of motion reduces to:

$$\sum_{j=1}^{N_n} m_{ij}^t \mathbf{a}_j^t = \mathbf{f}_i^{int,t} + \mathbf{f}_i^{ext,t} \quad (1.8)$$

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

$$m_{ij}^t = \sum_{p=1}^{N_p} M_p N_i(x_p^t) N_j(x_p^t) \quad (1.9)$$

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

$$7 \quad \mathbf{f}_i^{int,t} = - \sum_{p=1}^{N_p} M_p \mathbf{G}_{ip}^t \cdot \boldsymbol{\sigma}_p^{s,t}$$

$$8 \quad \mathbf{f}_i^{int,t} = - \sum_{p=1}^{N_p} M_p \mathbf{b}_p^t N_i(\mathbf{x}_p^t) + \sum_{p=1}^{N_p} M_p N_i(\mathbf{x}_p^t) \tau_p^{s,t} h^{-1} \quad (1.10)$$

10 where  $\mathbf{G}_{ip}^k = \Delta N_i(x)|_{x=X_p^t}$ . The nodal accelerations are obtained by explicit time integration of  
 11 eq. 1.8. To obtain stable solutions, the time step used in the analysis should be less than the  
 12 critical time step, which is defined as the ratio of the smallest cell size to the wave speed (Chen  
 13 and Brannon, 2002). The boundary conditions are enforced on the cell nodes and the nodal  
 14 velocities are obtained by solving the equation of motion at each node. The strain increment  
 15 for each material point is determined using the gradients of the nodal basis functions. The  
 16 corresponding stress increments are computed using the adopted constitutive law. After all the  
 17 material points have been completely updated, the computational mesh is discarded, and a new  
 18 mesh is defined for the next time step.

### 19 1.3.2 Boundary conditions

20 The Material Point Method uses standard shape functions, similar to those used in the Finite  
 21 Element Methods, hence the essential and the natural boundary conditions can be applied  
 22 to the background grid nodes in the same way as in the traditional FEM. The free surface  
 23 boundary conditions are satisfied, as MPM is formulated in the weak form. Implementation  
 24 of traction boundary conditions requires a set of material points to represent the boundary  
 25 layer. ? proposed a friction interaction for the planar boundary condition, using Coulomb's  
 26 friction criterion. The nodal accelerations were considered to include the frictional effects  
 27 instead of the forces, as the forces are proportional to the corresponding accelerations. The  
 28 static and kinematic frictions are applied tangential to the nodal boundary. Friction forces  
 29 are applied, only if the particles are in contact. The normal velocity and acceleration on the  
 30 boundary plane is zero. The shape functions used in the MPM are continuous, and hence  
 31 penetration between bodies are handled automatically without the need for any supplemental  
 1 contact algorithm (Chen and Brannon, 2002). In the MPM, the continuum body deforms and  
 2 moves in an arbitrary computation grid and all the boundary conditions are carried by the  
 3 boundary particles. If a boundary particle is present in a cell, then the cell boundary becomes a  
 4 part of the continuum body, and the cell size represents the thickness of the boundary. However,  
 5 in certain cases both the boundary particle and an interior particle can be found in a cell, in

which case the cell is still treated as a boundary cell, and the interior particle temporarily acts as a boundary particle. To avoid numerical errors, it is essential to consider smaller cell size along the boundary (Chen and Brannon, 2002).

In the MPM simulations, numerical noises are observed when the material points crosses the cell boundaries during deformation of the material, this is termed as cell crossing noise. If a material point is located very close to the cell boundary, it results in discontinuous nature of 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 1.3 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).

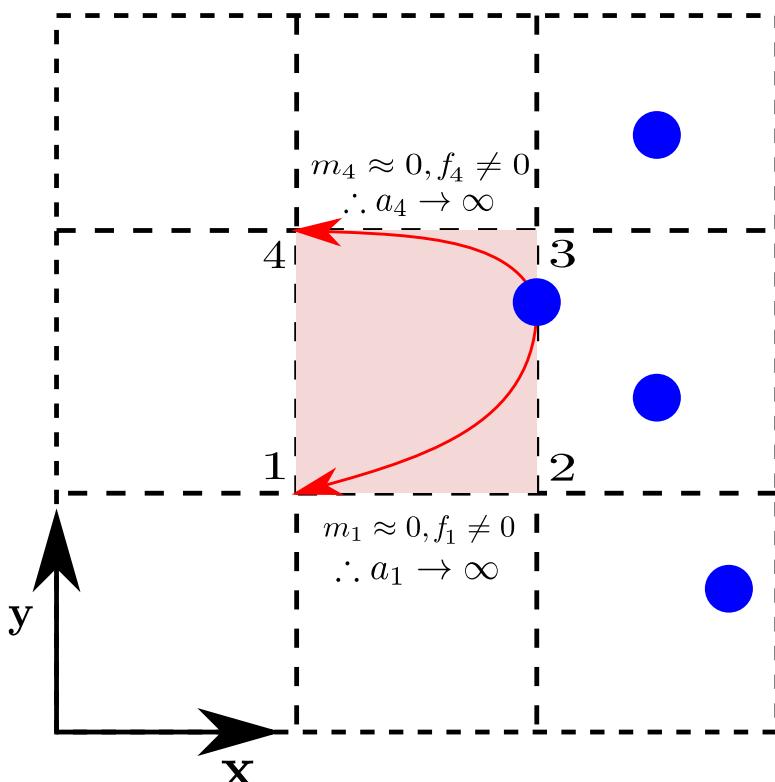


Figure 1.3 Cell crossing noise

**8 1.3.3 Solution scheme**

9 A step-by-step solution scheme for Material Point Method (?) is described below. It involves  
 10 three stages: preprocessing, computation, and post-processing:

**11 Preprocessing**

- 12 • A continuum body is discretized into a finite set of material points corresponding to  
 13 the original configuration of the body. The number of material points corresponds to  
 14 the resolution of the mesh size adopted in Finite Element Method. The material points  
 15 are followed throughout the deformation of the material, which gives a Lagrangian  
 16 description of the motion.
- 17 • An arbitrary computational grid is initialized to describe the natural coordinates of the  
 18 material points. For the purpose of simplicity, a Cartesian grid is usually adopted.
- 19 • The state variables (mass/density, velocity, strain, stress, other material parameter corre-  
 20 sponding to the adopted constitutive relation) are initialized at every material point.

**21 Computation scheme**

- 22 • For each material point, the mapping of the properties from the particles to the cell nodes  
 23 is accomplished using the shape functions computed from the particle position. The  
 24 nodal mass matrix is obtained as:

$$25 m_i^k = \sum_{p=1}^{N_p} M_p N_{ip}^k \quad (1.11)$$

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

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

$$v_i^k = \sum_{p=1}^{N_p} m_p v_p^t N_{ip}^t / m_i^t \quad (1.12)$$

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

$$\Delta\epsilon_p^{t+1} = \frac{\Delta t}{2} \sum_{i=1}^{N_n} \mathbf{G}_{ip}^t \mathbf{v}_i^t + (\mathbf{G}_{ip}^t \mathbf{v}_i^t)^T \quad (1.13) \quad \begin{matrix} 9 \\ 10 \end{matrix}$$

- 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

$$\Delta\sigma_p^{t+1} = \mathbf{D} : \Delta\epsilon_p^{t+1} \quad (1.14) \quad \begin{matrix} 13 \\ 14 \end{matrix}$$

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

$$\begin{aligned} \sigma_p^{t+1} &= \sigma_p^t + \Delta\sigma_p^{t+1} \\ \epsilon_p^{t+1} &= \epsilon_p^t + \Delta\epsilon_p^{t+1} \end{aligned} \quad (1.15) \quad \begin{matrix} 16 \\ 17 \\ 18 \end{matrix}$$

- The material point density is then updated as:

$$\rho_p^{t+1} = \frac{\rho_p^t}{\{1 + \text{tr}(\Delta\epsilon_p^{t+1})\}} \quad (1.16) \quad \begin{matrix} 20 \\ 21 \end{matrix}$$

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

$$\mathbf{a}_i^t = (\mathbf{f}_i^{int,t} + \mathbf{f}_i^{ext,t}) / m_i^t \quad (1.17) \quad \begin{matrix} 23 \\ 24 \end{matrix}$$

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

$$v_i^L = \mathbf{v}_i^k + a_i^k \Delta t \quad (1.18) \quad \begin{matrix} 26 \\ 1 \end{matrix}$$

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

$$\begin{aligned} \mathbf{x}_p^{t+1} &= \mathbf{x}_p^t + \Delta t \sum_{i=1}^{N_n} v_i^L N_{ip}^t \\ \mathbf{v}_p^{t+1} &= \mathbf{v}_p^t + \Delta t \sum_{i=1}^{N_n} a_i^t N_{ip}^t \end{aligned} \quad (1.19) \quad \begin{matrix} 3 \\ 4 \\ 5 \end{matrix}$$

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

figure 1.4 illustrates the steps involved in a MPM analysis.

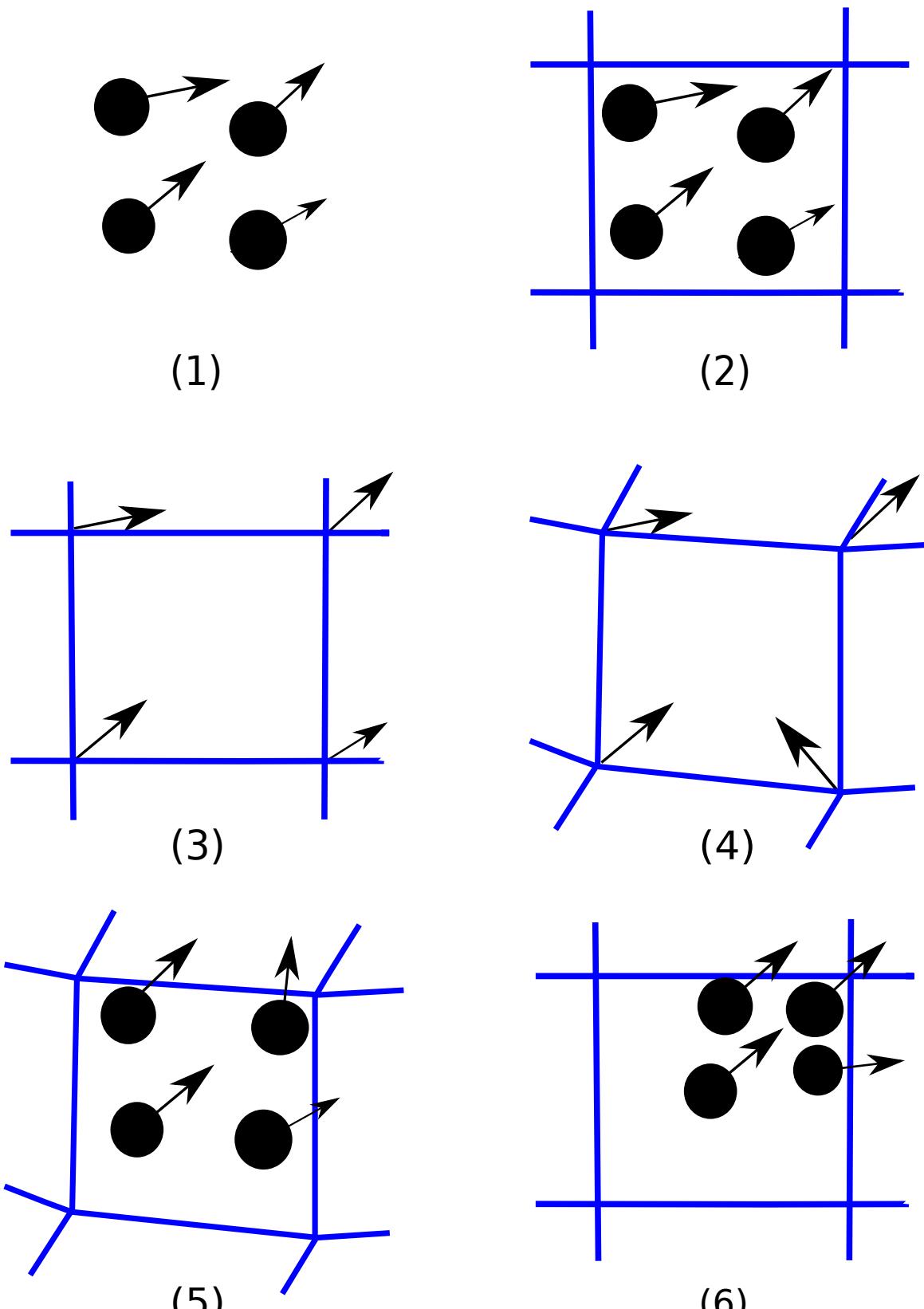


Figure 1.4 Illustration of the steps in the MPM algorithm for particles occupying a single cell in the background grid. (1). A representation of four material points (filled circles), overlaid with the computational grid (solid lines). Arrows represent displacement vectors. (2). The material point state 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)

**Post processing**

The final step in any analysis is post-processing. It involves visualization and extraction of the data from the analysis. In mesh-less methods, like the Material Point Method, structures are generally represented as points which represent a discrete region of the body. The MPM facilitates representation of arbitrarily complex geometries and is robust in computing large deformation problems. It has advantages over strictly grid based methods in simulations involving contact between multiple objects (Bardenhagen et al., 2000). The MPM poses a whole new set of visualization problem, it is essential to visualize the general configuration of the body as well as to observe the finer details like development of cracks or separation of chunk of material from the body. The body is discretized into conceptual material points, which carry all the relevant information of the corresponding segment. The unique qualities of the MPM necessitate the need to visualize the particle data in a way that is informative and appropriate. In the MPM, the particle data represents the finite portion of a larger continuum, and the ability to see and interpret the macroscopic structure created by these particles is vital (Bigler et al., 2006). There are two vital aspects in visualizing the MPM data: (1) visualization of the structure represented by material points, and (2) understanding the qualitative trend associated with the material points like mass, velocity or stress. The MPM output data contains both the material point and the grid data, one approach in visualizing the MPM data is by rendering the interpolated particle values on grid nodes using *iso-surfacing* (Lorensen and Cline, 1987) or *volume rendering* (Levoy, 1988) technique. In regions where the material points are sparse, it is necessary that the grid resolution to be sufficiently fine to compensate for the missing features. This results in storing large amount of unnecessary data in regions where sufficient material points are present. Thus, it is advantageous to visualize the MPM data of the material points as particles (Bigler et al., 2006). Particle visualization involves rendering the particles as a sphere or an ellipsoid representing the size and location of the fraction of the continuum (Gumhold, 2003; Krogh et al., 1997; Kuester et al., 2001). Colour mapping of scalar quantities such as mass, velocity, or stress of a material point provide additional qualitative understanding of data.

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



# Chapter 2

9

## Particulate modelling of granular flows

10

### 2.1 Introduction

11

Granular materials often exhibit different behaviour under different circumstances. Fluidized granular material often resembles a liquid, and reveals surface waves. In certain situations, granular materials behave more like a solid exhibiting plastic deformations. Despite the wide variations in the physical and chemical properties of the grains, the discrete granular structure has a rich generic phenomenology, which motivates us to understand the fundamental behaviour of these materials. A granular material can be considered as a continuous material if it is viewed at a macroscopic scale, ignoring the fact that it is composed of grains. On a macroscopic scale, the behaviour of the granular material could be approximately defined using continuum mechanics. However, On a particle level, the granular materials exhibit complex solid-like and/or fluid like behaviour depending on the way the grains interact with each other. The Analytical and Finite Element models, which consider granular materials as continuum cannot take into account the local geometrical processes that govern the mechanical behaviour of a non-homogeneous soil. The application of continuum models to describe granular flow poses subtle problems for statistical analysis ([Mehta and Barker, 1994](#)). The particle level description of the granular material enriches the macro-scale variables which poorly account

- 1 for the local rheology of the materials. Numerical models based on the Discrete Element
- 2 approach proposed by ? are capable of simulating the granular material as a discontinuous
- 3 system. Although, modern measurement techniques can probe into the local granular variables,
- 4 like particle position, velocities, contact forces, etc., they have inherent limitations in acquiring
- 5 those variables. The *discrete-element* approach is a powerful and reliable research tool to study
- 6 the behaviour granular materials at the grain-scale. This approach involves applying Newton's
- 7 equation of motion simultaneously to all particles described as rigid solid bodies by considering
- 8 the contact forces and the external forces acting on the particles. For a given boundary condition,

the collective mechanical response of particles to the external force leads to the relative motion between particles constrained in a dense state and/or by in-elastic collisions in the loose state. ? applied this method to granular geomaterials, and called it the *Distinct Element Method*, to differentiate from the existing *Finite Element method* used in geomechanics. The attribute “distinct” refers to the degrees of freedom of individual particles, but it was later replaced by “discrete” to underline the discrete nature of the system. A similar method called *Discrete Element Method* (DEM) was used at the same time for the simulation of molecular systems with classical schemes that could be directly applied to granular materials. Hence, the acronyms DEM and DEM are used synonymously to describe the methods of simulating granular materials at particle scale. In-spite of the formal analogy (particles and force laws) between granular and molecular systems, the physics is fundamentally different. The interactions between individual grains are governed by unilateral contact laws, and the mechanism of energy dissipation is through friction and inelastic collisions. Moreover, granular materials have a wide variation in their particle shape and size distribution that require appropriate numerical treatments. In the Molecular Dynamics approach, the normal reaction force, which prevents the interpenetration of two grains is proportional to the depth of penetration. Thus, frictional contact between grains can be expressed as a function of the configuration variables, which describe the positions and velocities of the grains ([Radjai and Dubois, 2011](#)). Discrete-Element methods, which describe interactions between grains based on the explicit overlap between the grains are termed as *smooth methods*. Another approach is the *non-smooth approach* ([Jean, 1999](#)), which describes the behaviour of discrete elements using the main features of uni-laterality and Coulomb friction, and by neglecting the finer details such as interpenetration and overlap between grains. The fundamental difference between the non-smooth method and the common DEM or Discrete Element Method (DEM) approach lies in the treatment of small length and time scales involved in the dynamics of granular media. In DEM-type DEM, the particles are treated as rigid bodies but the contacts between particles 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 particle dynamics is described at a larger scale than the elastic response time and displacement scales ([Jean, 1999; Radjai and Richefeu, 2009](#)).

## 2.2 Discrete Element Method

Discrete Element Method computes the equilibrium and the trajectories of a classical multi-body system. The Discrete Element Method is a simple and flexible discrete-element approach, which

involves applying Newton's second law of motion to each grain to describe the deformation of the granular assembly.

$$m_i \frac{d^2 \vec{x}_i}{dt^2} = \mathbf{F}_i, (i = 1, \dots, N) \quad (2.1)$$

where  $N$  is the number of grains in the simulation,  $m_i$  is the mass of grain  $i$ ,  $\vec{x}_i$  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. 2.1. 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 particles 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  $N$  particles and solving the Newton's equation of motion for the system until the properties of the system no longer change with time (equilibration of the system). After equilibration, the actual analysis is performed.

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

$$\frac{\partial^2 \vec{r}_i}{\partial t^2} = \frac{1}{m_i} \vec{\mathbf{F}}_i(\vec{r}_j, \vec{v}_j, \vec{\varphi}_j, \vec{\omega}_j) \quad (2.2)$$

$$\frac{\partial^2 \vec{\varphi}_i}{\partial t^2} = \frac{1}{\hat{J}_i} \vec{\mathbf{M}}_i(\vec{r}_j, \vec{v}_j, \vec{\varphi}_j, \vec{\omega}_j), (j = 1, \dots, N) \quad (2.3)$$

- <sup>1</sup> The force  $\vec{\mathbf{F}}_i$  and the torque  $\vec{\mathbf{M}}_i$ , which act on particle  $i$  of mass  $m_i$  and the tensorial moment
- <sup>2</sup> of inertia  $\hat{J}_i$  are (sometimes complicated) functions of the particle positions  $\vec{r}_j$ , their angular
- <sup>3</sup> orientations  $\vec{\varphi}_j$ , and their corresponding velocities  $\vec{v}_j$  and  $\vec{\omega}_j$ . In a two-dimensional system,
- <sup>4</sup> the angular orientation of a particle is described by a single (scalar) quantity  $\varphi_i$  and the moment
- <sup>5</sup> of inertia reduces to a scalar value  $J_i$ . The Newton's equation of motion can be written as:

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

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

9 For granular particles in the absence of long range fields, the force  $\vec{\mathbf{F}}_i$  and the torque  $\vec{\mathbf{M}}_i$   
 10 acting upon the particle  $i$  are given as sum of the pairwise interaction of particle ‘ $i$ ’ with all  
 11 other particles of the system:

$$12 \quad \vec{\mathbf{F}}_i = \sum_{j=1, j \neq i}^N \vec{\mathbf{F}}_{ij}, \quad \vec{\mathbf{M}}_i = \sum_{j=1, j \neq i}^N \vec{\mathbf{M}}_{ij} \quad (2.2)$$

$$13$$

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

### 23 2.2.1 The Forces

24 The force  $\mathbf{F}_i$  in eq. 2.1 represents both the grain to grain interaction force, and other external  
 25 forces acting on the system. Therefore, the force  $\mathbf{F}_i$  is expressed as:

$$26 \quad \mathbf{F}_i = \sum_{j \neq 1} \mathbf{F}_{ij} + \mathbf{F}_{ext,i} \quad (2.3)$$

$$27$$

28 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  
 29 of gravity,  $\mathbf{F}_{ext,i} = m_i \mathbf{g}_i$ . The methodology to incorporate any other external forces in the  
 30 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 2.1). The contact force can be  
 decomposed into two components, as the normal ( $F_n$ ) and the tangential ( $F_t$ ) components:

$$\mathbf{F}_{ij} = F_n \mathbf{n} + F_t \mathbf{t} \quad (2.4)$$

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.

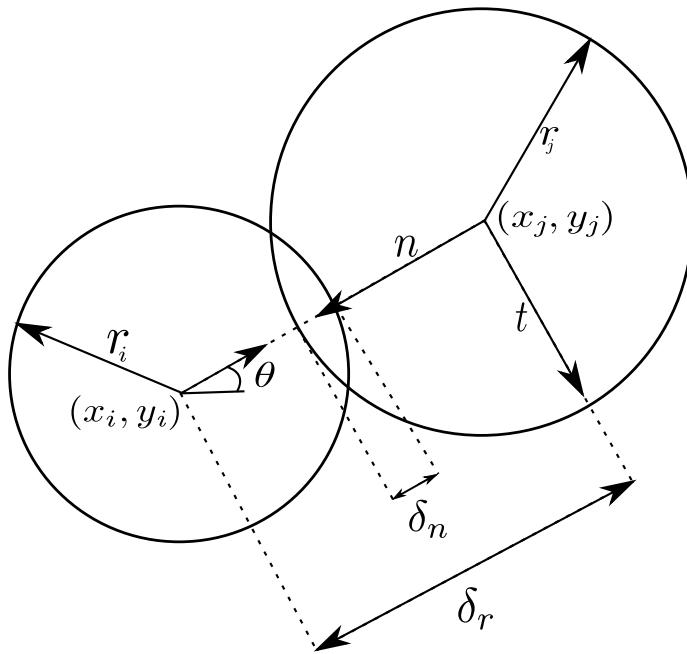


Figure 2.1 Grains *i* and *j* in contact, and the separation  $\delta_n$  is used to calculate the normal force

## Normal force

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

$$\delta_n = |x_i - x_j| - r_i - r_j \quad (2.5)$$

<sup>1</sup> where  $x_i$  and  $x_j$  are the centres of the grains and  $r_i$  and  $r_j$  are their radii (see Figure 2.1). When <sup>2</sup>  $\delta_n > 0$ , the two grains are not in contact, and there is no interaction. When  $\delta_n < 0$ , the two <sup>3</sup> grains overlap, and there is a repulsive normal force that pushes the two grains apart. The <sup>4</sup> simplest model is to consider the contact as a linear spring with damping. The repulsive force <sup>5</sup> depends linearly on  $\delta_n$ , and is controlled by the stiffness of the grain. The energy dissipation <sup>6</sup> due to the interaction between grains is an intrinsic characteristic of the granular material and <sup>7</sup> is incorporated by adding a damping force that opposes the relative velocity for the duration of <sup>8</sup> the contact. The interaction force at the contact is idealized as a simple spring-dashpot system,

- 9 with elastic and dissipative constants ([Luding et al., 1994](#)).

10

$$F_n = \begin{cases} 0, & \delta_n > 0 \\ -k_n \delta_n - \gamma_n \frac{d\delta_n}{dt}, & \delta_n < 0 \end{cases} \quad (2.6)$$

11

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

18

$$\hat{F}_n = -k_n \delta_n - \gamma_n \frac{d\delta_n}{dt}, \quad F_n = \begin{cases} 0, & \hat{F} \leq 0 \\ \hat{F}_n, & \hat{F} > 0 \end{cases} \quad (2.7)$$

19

20 For pairwise collisions, the normal force ( $F_n$ ) represented as  $k_n \delta_n + \gamma_n$  causes a decrease in  
 21 the relative normal velocity of the particles by a factor  $\varepsilon$ . This factor is the *coefficient of*  
 22 *restitution*, and is defined as  $\varepsilon \approx g'/g$ , where  $g$  is the absolute normal relative velocity before  
 23 the collision and  $g'$  corresponds to the post-collision value. The relative velocity,  $d\delta_n/dt > 0$ ,  
 24 can be obtained by differentiating [eq. 2.5](#). Thus we obtain:

25

$$\frac{d\delta_n}{dt} = (\mathbf{v}_i - \mathbf{v}_j) \cdot \mathbf{n} \quad (2.8)$$

26 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  
 27 numerical integration of [eq. 2.8](#) yields the separation  $\delta_n$  and permits us to generalise the model  
 28 and treat the tangential forces as well, as explained in the next section. By integrating Newton's  
 29 equation of motion it is found that the linear force corresponds to the co-efficient of restitution,  
 which is defined as:

1

2

$$\varepsilon = \exp\left(-\frac{\pi \gamma_n}{2m^{eff}} / \sqrt{\frac{Y}{m^{eff}} - \frac{\gamma^2}{2m^{eff}}}\right) \quad (2.9)$$

3

### Tangential force

4

5 Granular particles are not perfect spheres, but have a complicated surface texture, therefore  
 6 at oblique collisions, besides the normal force there is a tangential force too. Even perfectly  
 7 smooth spheres exert a tangential force due to their bulk viscosity (?). To build a heap of spheres  
 8 on a flat surface, the particles 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 particle surfaces at the point of contact. The point of contact is an approximation, as the description of the normal force assumes a compression  $\delta_n$ , which implies a contact surface in 3-D or a contact line in 2-D. Assuming a tangential spring of length  $\delta_t$  exerts an opposing force to the relative tangential displacements (ignoring the effect of relative rolling between the particles), the tangential force can be postulated similar to the normal force ( eq. 2.8) as:

$$\frac{d\delta_t}{dt} = (\mathbf{v}_i - \mathbf{v}_i) \cdot \mathbf{t} \quad (2.10)$$

This equation must also be numerically integrated, just like eq. 2.1. 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,  $\delta_t$  can be calculated similar to the normal force. The tangential force is assumed to be governed by Coulomb's friction law.

$$|F_t| \leq \mu F_n \quad (2.11)$$

where  $F_t$  is the tangential force and  $\mu$  is the friction coefficient. It is therefore necessary to constrain the tangential force to remain less than or equal to  $\mu F_n$ . To impose the condition in eq. 2.11, two-stages similar to the normal force computation is adopted. The first step is to evaluate the candidate force, and is then accepted if it obeys the condition in eq. 2.11.

$$\hat{F}_t = -k_t \delta_t - \gamma \frac{d\delta_t}{dt}, \quad F_t = \begin{cases} sgn(F_t), & |\hat{F}| \geq \mu F_n \\ \hat{F}_t, & |\hat{F}| < \mu F_n \end{cases} \quad (2.12)$$

where  $k_t$  is the stiffness of the tangential spring and  $\gamma$  is the damping constant. If  $|F_t| = \mu F_n$ , the contact is sliding, otherwise, it is non-sliding. It can be noted that the normal force ( eq. 2.7) and the tangential force ( eq. 2.12) are handled in the same way in Discrete Element Method. When the grains slide against each other, they do not retain any memory of their initial position, and hence do not return to its original position. In order to model this behaviour, a limiting value of  $\delta_t$  is imposed. When the contact slides  $\delta_t = \pm \mu F_n / k_t$  is imposed.

In addition to sliding, the grains can roll relative to one another about their centre of mass due to the tangential force acting at their contact surface. In this case,  $d\delta_t/dt = 0$ . It is important to assume that the grains touch at a single point instead of overlapping, i.e.  $\delta_n = 0$ . This point

9 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  
 10  $v_i + r_i(\boldsymbol{\omega} \times \mathbf{n})$ . If it belongs to grain  $j$ , its velocity is  $v_j + r_j(\boldsymbol{\omega} \times \mathbf{n})$ . The relative velocity is the  
 11 difference between these two velocities.

$$12 \quad \frac{d\delta_t}{dt} = (\mathbf{v}_i - \mathbf{v}_j) \cdot \mathbf{t} - (r_i \boldsymbol{\omega}_i + r_j \boldsymbol{\omega}_j) \times \mathbf{n} \quad (2.13)$$

13 It should be noted that the eq. 2.13 is only an approximation, as the grains in Molecular  
 14 Dynamics do not touch at points, but overlap. It is therefore an approximation that produces  
 15 an error of order  $O(\delta_n/r)$  (Radjai and Dubois, 2011). It is assumed that the contact forces are  
 16 exerted at the point of contact. It implies that the tangential force is accompanied by torque  
 17 acting on two grains. If the overlap is zero, these torques are:

$$18 \quad \tau_{ij} = -(a_i \mathbf{n}) \times (F_t \mathbf{t}), \tau_{ji} = -(a_j \mathbf{n}) \times (F_t \mathbf{t}), \quad (2.14)$$

20 The torques modify the angular velocities of the grains. It is therefore necessary to incorporate  
 21 the equation for the angular coordinates of the grains in eq. 2.1:

$$22 \quad I_j \frac{d\boldsymbol{\omega}_i}{dt} = \sum_{j \neq i} \tau_{ij} \quad (2.15)$$

23 where  $I_j$  is the moment of inertia of grain  $j$ . The eq. 2.14 is only valid when  $\delta_n = 0$ . The  
 24 torque is a vector product of the force and its lever arm. It is assumed that the lever arms  
 25 have lengths equal to  $r_i$  and  $r_j$ , which is true only when the grains do not overlap, hence in  
 26 this case they produce an error of order  $O(\delta_n/r)$ . It is nevertheless desirable to damp this  
 27 type of motion (Radjai and Dubois, 2011). The interaction between two solid bodies is much  
 28 more complex than that is described by the simple linear model. Nevertheless, the linear  
 29 force law has several advantages. It is simple to implement, and its harmonic behaviour is  
 30 well understood, which makes it easier to interpret the results. The most common non-linear  
 1 interaction law is the Hertz law (Hertz, 1882). In certain situations, such as a quasi-static  
 2 packing, a non-linear law can have significant influence on the acoustic properties (?), and on  
 3 the global stiffness (?). However, in case of rapid granular flows, the interaction force between  
 4 the particles has almost no effect on the phenomenon, and a linear law can be used to describe  
 5 this kind of behaviour (Radjai and Dubois, 2011).

## 2.2.2 Numerical algorithm and integration scheme

The efficiency of a Discrete Element Method program is mainly determined by its efficiency  
 7 to compute the interaction forces between particles. If we consider a model system with  
 8

pairwise interactions, we have to consider the contribution of the force on particle  $i$  due to all its neighbours. If we consider only the interaction between a particle and the nearest image of another particle, then for a system of  $N$  particles, we must evaluate  $N \times (N - 1)/2$  pair distances. Consider a system of 1000 particles, at every time step all possible pairs of particles have to be considered to compute the interaction forces, hence,  $N(N - 1)/2 \approx 500,000$  force computations are required. For short-range particle interactions, the majority of these force evaluations is unnecessary as the corresponding particles are located far apart and do not necessarily touch each other. For a dense system of equally sized particles, the particles can have contacts with not more than 6 particles, 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 *lattice* algorithm. The Verlet algorithm described in Grubmuller et al. (1991) is implemented in the present study.

<sup>10</sup> **Verlet list algorithm**

<sup>11</sup> The Verlet list algorithm assumes a cut-off value, so that only neighbouring particles that  
<sup>12</sup> contribute to the energy of a particle  $i$  are considered. It is advantageous to exclude the particles  
<sup>13</sup> that do not interact in the memory expensive energy computation. [Verlet \(1967\)](#) developed  
<sup>14</sup> a book-keeping technique, commonly referred to as the Verlet list or neighbour list, which  
<sup>15</sup> is illustrated in [2.2](#). In this method a second cut-off radius  $r_v > r_c$  is introduced, and before  
<sup>16</sup> we calculate the interactions, a list is made (the Verlet list) of all particles within a radius  
<sup>17</sup>  $r_v$  of the particle  $i$ . In the subsequent calculations of the interactions, only those particles in  
<sup>18</sup> this list will be considered. The idea of the Verlet algorithm is based on a simple property of  
<sup>19</sup> particle dynamics: neighbourhood relation between particles can only change slowly, i.e. two  
<sup>20</sup> particles which are close to each other at a given time step will remain as neighbours, at least  
<sup>21</sup> in the following few time steps. During initialization the neighbourhood relations between  
<sup>22</sup> the particles, i.e. the distance of all close pairs of particles are computed. Two particles are  
<sup>23</sup> considered as neighbours if the distance of their surface is smaller than a predefined distance  
<sup>24</sup> *Verlet distance*:

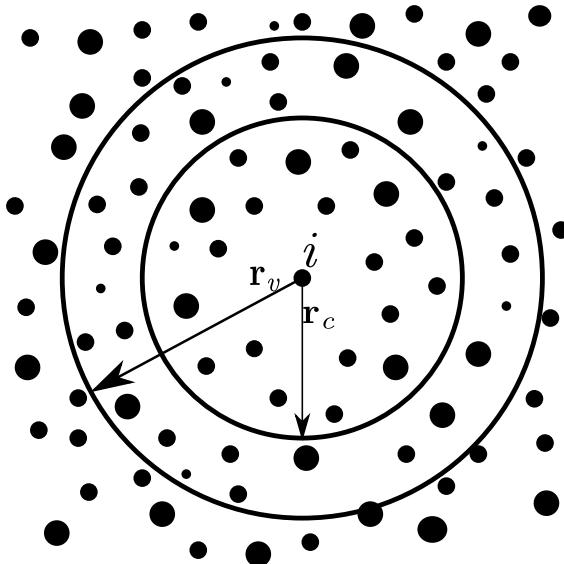


Figure 2.2 The Verlet list: a particle  $i$  interacts with those particles with the cut-off radius  $r_c$ , the Verlet list contains all the particles within a sphere with radius  $r_v > r_c$

$$(|\vec{r}_i - \vec{r}_j| - R_i - R_j) < \text{Verlet distance} \quad (2.16)$$

For each particle 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 particle. For construction of the lists only pairs whose particles reside in the same or adjacent grids are considered. This procedure guarantees the detection of all close pairs

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

$$|\vec{r}_i - \vec{r}_j| - R_i - R_j > 0 \quad (2.17)$$

The above condition is required for all pairs  $(i,j)$  of particles which are *not* known as neighbours. This condition is a criterion to update the Verlet lists (?). Assume at the instant when the Verlet lists are constructed, the surfaces of the particles have the distance  $|\vec{r}_i - \vec{r}_j| - R_i - R_j >$  Verlet distance, i.e. they are not classified as neighbours. If the Verlet lists are updated before one of these particles has travelled the distance  $verletdistance/2$  since the lists were constructed, they can never collide without being recognized as neighbours first. This is explained in figure 2.3. The impact of optimisation of the Verlet list algorithm has negligible effect on the computation time, as the algorithm is quite efficient already and only consumes a few percent of the total computation time in construction of the Verlet lists. The implementation of the Verlet list algorithm in force computation drastically reduces the computation time in comparison to the linear algorithm. The performance of the Verlet list algorithm is controlled by two crucial parameters: the number of cells  $N_c$ , for the construction of the Verlet lists and the Verlet distance  $r_v$  (?).

#### 4 Leap frog or Verlet integration algorithm

- 5 Discrete Element Method involves numerically solving Newton's equation of motion eq. 2.1,
- 6 which is an ordinary differential equation. Choosing an integration algorithm is important, as
- 7 the forces are not always differentiable in time, and the temporary derivative of the force is
- 8 discontinuous when the contact splits. It is also very essential to numerically integrate eq. 2.13
- 9 with the same precision as eq. 2.1. At first, computational speed seems important. It is usually
- 10 not very relevant because the fraction of time spent on integrating the equation of motions

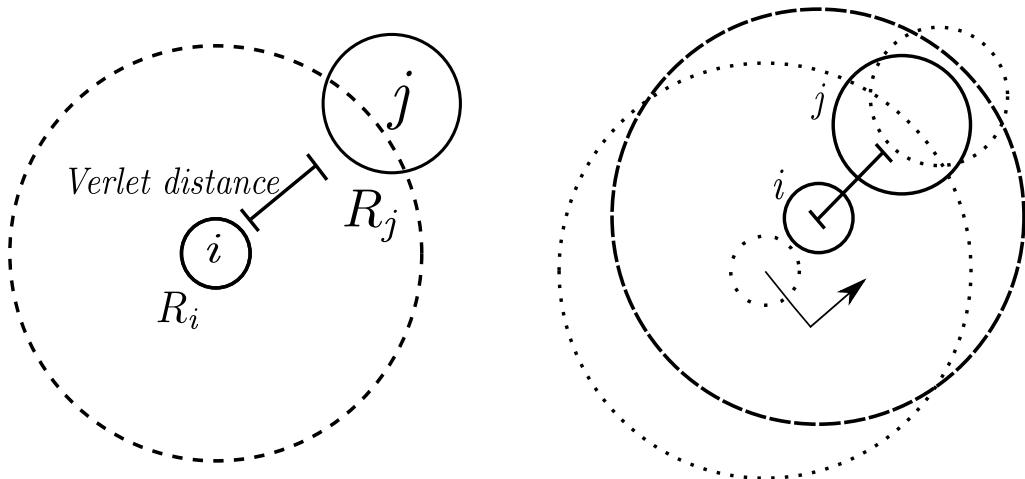


Figure 2.3 Checking the validity of Verlet lists. *Left:* the particles  $i$  and  $j$  are not recognized as neighbours since the distance of their surface is larger than the Verlet distance. The radius of the dashed circle is  $R_i + R_{max} + \text{Verlet distance}$ . *Right:* in the most unfortunate case the particles approach each other directly, travelling at the same velocity. As soon as one of the particles has travelled the distance  $\text{Verlet distance}/2$  (arrows), the Verlet lists have to be rebuilt. The particles  $i$  and  $j$  are now recognized as neighbours. Figure redrawn after ?.

11 (as opposed to computing the interactions) is small. Accuracy for large time steps is more  
 12 important, because the larger the time step that we can use, the fewer evaluation of the forces  
 13 are needed per unit of simulation time. Hence, this would suggest that it is advantageous to use  
 14 a sophisticated algorithm that allows use of larger time step. Algorithms that allow the use of  
 15 large time steps, achieve this by storing information on increasingly higher-order derivatives of  
 16 the particle coordinates. Consequently, they tend to require more memory storage. However,  
 17 the most important aspect to consider is the energy conservation. It is important to distinguish  
 18 between two kinds of energy conservation: the short time and long time. The sophisticated  
 19 higher-order algorithms tend to have very good energy conservation at short times. However,  
 20 they often have undesirable feature that results in drifting of the overall energy for longer  
 1 times. In contrast, the Verlet style algorithms tend to have only moderate short term energy  
 2 conservation, but little long-term drift (?). In this case, such algorithms are not useful. They  
 3 are more complicated to program, and do not yield a more precise solution (?). It might seem  
 4 important to have an algorithm that accurately predicts the trajectories of all particles for both  
 5 short and long durations, however no such algorithm exists. In certain cases, two trajectories  
 6 that are initially very close may diverge exponentially as time progresses. Any integration error,  
 7 however small it may be, would always diverge the predicted trajectory exponentially from the  
 8 true trajectory. This phenomenon is called the Lyapunov instability, and it is devastating to  
 9 the whole idea of Discrete Element Method simulation, but we have good reasons to assume  
 10 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 particles 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 particles 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 particle co-ordinates.

$$t(t + \Delta t) = rt + \mathbf{v}(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 + \dots \quad (2.18)$$

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

$$\mathbf{v}_{k+1/2} = \mathbf{v}_{k-1/2} + \mathbf{a}_k \Delta t, \quad x_{k+1} = x_k + \mathbf{v}_{k+1/2} \Delta t \quad (2.19)$$

This algorithm determines the new grain position with an error of order  $O(\Delta t^4)$ . But [eq. 2.19](#) 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

<sup>12</sup> write:

<sup>13</sup>  $\mathbf{v}_k = \mathbf{v}_{k-1/2} + \mathbf{a}_{k-1}\Delta t/2$  (2.20)

<sup>14</sup> The equation uses the acceleration of the preceding time step to estimate the velocity. This  
<sup>15</sup> approximation does not diminish the order of the algorithm. eq. 2.20 estimates  $\mathbf{v}_k$  with an  
<sup>16</sup> error of order  $\mathbf{O}(\Delta t^2)$ , which produces an error of the same order in the calculation of the  
<sup>17</sup> force in eq. 2.19. But this causes only an error of order  $\mathbf{O}(\Delta t^3)$  in the velocity and an error  
<sup>18</sup> of order  $\mathbf{O}(\Delta t^4)$  in the position. However, this problem does not exist in energy conservation  
<sup>19</sup> systems, because the computed forces do not depend on the velocities of the grains. The  
<sup>20</sup> heaviest computational task is the evaluation of forces and not the integration of equations. The  
<sup>21</sup> Verlet integration scheme is summarized in eq. 2.21 and figure 2.4. To calculate the forces and  
<sup>22</sup> acceleration, it requires the positions and velocities at time t:

<sup>23</sup>  $\mathbf{v}(t + \Delta t/2) = \mathbf{v}(t - \Delta t/2) + \mathbf{a}(t)\Delta t$   
<sup>24</sup>  $x(t + \Delta t) = x(t) + \mathbf{v}(t + \Delta t/2)\Delta t$   
<sup>25</sup>  $\mathbf{v}(t) = \mathbf{v}(t - \Delta t/2) + \mathbf{a}(t - \Delta t)\Delta t/2$  (2.21)  
<sup>26</sup>

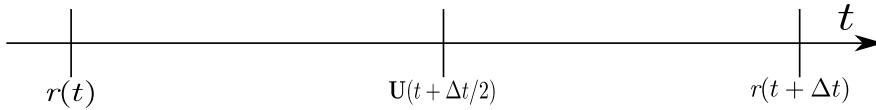


Figure 2.4 Verlet integration scheme

<sup>27</sup> The analysis of the Discrete Element Method formulation reveals that the linear force law  
<sup>1</sup> gives the model a harmonic character, showing that it is very closely related to simple models  
<sup>2</sup> widely used in physics and mechanics. The shortest time scales often arise from the oscillations  
<sup>3</sup> of one or two grains. The integration algorithm must resolve these movements with sufficient  
<sup>4</sup> precision. Thus, the time steps used must be smaller than these time scales, the most rapid  
<sup>5</sup> frequency is usually  $\omega_N$ , the characteristic oscillation frequency of very short waves. This  
<sup>6</sup> frequency is proportional to  $\omega_0$ , which is easier to estimate. Therefore, it is essential to choose  
<sup>7</sup> a time step  $\Delta t \approx \epsilon/\omega_0$ , where  $\epsilon$  is a constant that depends on the integration algorithm. Values  
<sup>8</sup> such as  $\epsilon \approx 0.01$  are often a reasonable choice (?). In the case of rapid granular flows, the time  
<sup>9</sup> step must be small enough so that the fastest grains move only by a small fraction of their size  
<sup>10</sup> during one time step. The grains must be stiff enough so that violent collisions do not lead to  
<sup>11</sup> large overlaps between particles.

### 2.2.3 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 particles, which obey the same rules of interaction as the particles of granular material. By varying the size and position of the wall particles, 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 particles and the wall. The interaction force is computed in a similar fashion to that of a pair of particles in contact and is divided into the normal and tangential components. The compression of the particle upon collision with the wall is calculated along the normal direction to the wall and the particle contact.

### Periodic boundary

The effect of a wall on the response of particles is very critical, especially in numerical simulation where the number of particles is relatively fewer in comparison to the experiments.

- <sup>1</sup> The undesired effect of a wall can be eliminated using periodic boundary conditions, i.e. a
- <sup>2</sup> periodic extension of the simulation area in one or more dimensions. Any particle leaving the
- <sup>3</sup> system at one side is reintroduced at the opposite side, and correspondingly the interaction
- <sup>4</sup> forces between particles at opposite sides of the simulation area are taken into account. In this
- <sup>5</sup> framework, the simulation domain becomes a unit area containing particles with periodic copies
- <sup>6</sup> paving the whole system. The periodic boundary conditions extend the system boundaries
- <sup>7</sup> to infinity, so that the simulation cell simply plays the role of a coordinate system to locate
- <sup>8</sup> particle positions 2.5.

- <sup>9</sup> The external stresses or displacements are applied on the simulation box by constraining the
- <sup>10</sup> degrees of freedom of the wall, which are alternatively kept free or fixed depending on whether
- <sup>11</sup> 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 particle 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.

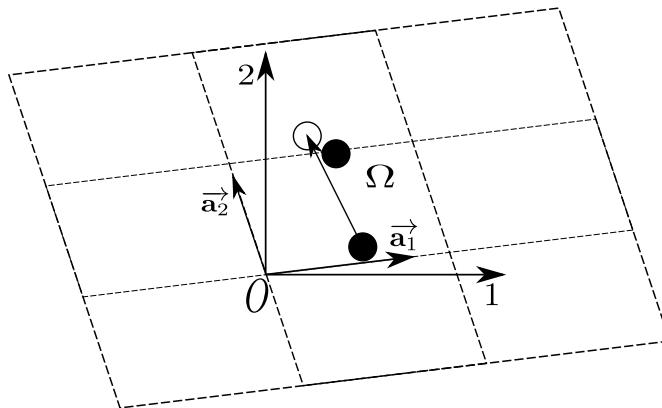


Figure 2.5 A 2D simulation cell  $\omega$  with its basis vectors in an absolute frame. A particle located at the right boundary interacts with the image of another particle located at the left boundary.

Let us consider a collection of  $N_p$  particles 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 2.5. 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 particle  $i \in \Omega$  of coordinates  $\vec{r}(i)$  are given by:

$$\vec{r}(i') = \vec{r}(i) + \sum_{k=1}^2 i_k \vec{a}_k \quad (2.22)$$

The particles belonging to the cell  $\Omega$ , characterized by  $i_1 = i_2 = 0$ , can interact with the particles of the same cell but also with image particles 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 particle. The distance between two particles  $i$  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 particle  $i$  may leave but one of its images  $i'$  enters at the same moment. In order to keep all original particles in the cell, the status “original” should be reserved to the particles whose

centres belongs to  $\Omega$ . Hence, whenever a particle  $i$  leaves the simulation cell, it becomes an image of  $i'$ , which then becomes the original. This means that a particle crossing a border of the simulation cell, returns to the cell by crossing another border.

#### 2.2.4 Particle Assembling Methods

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

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

12 and connectivity is another important property, which depends on the assembling rules. In the  
 13 present study, the initial grain packing is obtained using ballistic deposition technique.

14 **Ballistic deposition**

15 Initially a random arrangement of particles which do not touch each other is generated ( ??).  
 16 The radii of the particles are chosen from the interval of  $(R_{min}, R_{max})$  in such a way that the  
 17 total mass of all particles from a certain size interval is the same for all sizes, thus ensuring that  
 18 neither larger nor smaller particles dominate the system. This distribution can be obtained, if  
 19 the radii are chosen according to the probability distribution:

$$20 \quad p(R) = \frac{R_{min}R_{max}}{R_{max} - R_{min}} \frac{1}{R^2} \quad (2.23)$$

$$21$$

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

$$24 \quad R = \frac{R_{min}R_{max}}{R_{max} - z(R_{max} - R_{min})} \quad (2.24)$$

$$25$$

26 This transformation is applied to initialise the particle radii and the particles are arranged  
 27 randomly on a regular lattice. The configuration of particles obtained after this step is presented  
 28 in ???. In the second step, the particles arranged in a regular lattice are allowed to fall down  
 29 and are packed using the *random deposition with relaxation method*, a ballistic deposition  
 30 technique. The geometrical methods help in this way to improve numerical efficiency in the  
 31 preparation phase. For example, gravitational deposition of particles located initially on a  
 32 regular grid can require hours of computation whereas a nearly similar result may be obtained  
 33 by means of a geometrical method in only a few minutes. The drawback is that the resulting  
 1 sample will not be in mechanical equilibrium and no information is available on the contact  
 2 forces. Nevertheless, depending on the relaxation rule, the sample may still be sufficiently close  
 3 to equilibrium to be considered as a good starting point for mechanical simulations. Hence,  
 4 a combined approach of ballistic deposition and Discrete Element Method is adopted in the  
 5 present study to generate mechanically stable samples. The random deposition and relaxation  
 6 method, first proposed by [Vold \(1959\)](#); ? and developed by [Jullien et al. \(1992\)](#) and [Meakin and Jullien \(1985\)](#),  
 7 is adopted in the present study. The general principle of this method is  
 8 quite simple (see figure 2.6); it consists of placing the particles consecutively on a substrate or  
 9 a layer of already deposited particles. Each particle first touches the substrate or a deposited  
 10 particle, then undergoes a relaxation process (single-particle restructuring) along the steepest  
 11 descent (steepest-descent model) until a more stable position according to a stability criterion

is reached. The construction of the packing proceeds layer by layer from the substrate, hence this deposition model is also known as bottom-to-top restructuring model. The first step is to release a particle from a random position above the substrate. Upon contact with the first deposited particle, the particle rolls following the steepest descent until a new contact is formed with a second particle. In 2D, two contacts are sufficient to balance a particle if its centre of gravity lies between the two contacts. This corresponds to a position of local stable equilibrium. If this criterion is not met, the particle continues to roll and the procedure is iterated until a local stable position is reached. The wall effects are eliminated by adopting periodic boundaries (Sec 2.2.3) in the horizontal direction (perpendicular to that of deposition). figure 2.6 shows a small sample prepared by this method (the grey particles are periodic images of the black ones). In this method, the order of deposited particles is generally random and independent of their sizes. The mechanically stable sample obtained from equilibrating the random deposited and relaxed sample is presented in ??

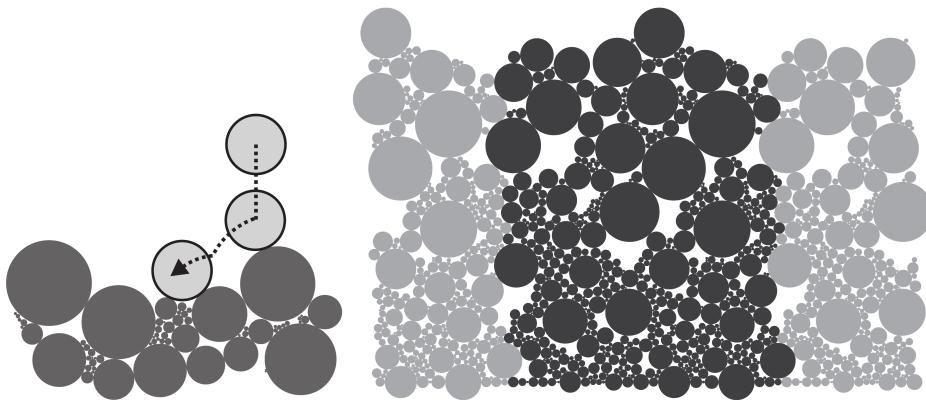


Figure 2.6 (a)*Ballistic deposition*: first contact followed by steepest descent; (b) small-scale periodic sample ([Radjai and Dubois, 2011](#))

1

## 2 Statistical analysis of prepared sample

3 In order to ensure the homogeneity of the sample prepared using the above technique, a  
4 statistical analysis of the sample is performed. Various parameters such as particle size,  
5 coordination number, contact normal direction, contact normal force, etc., can be used in the  
6 statistical analysis to verify the homogeneity of the prepared sample. Of the various parameters  
7 available, the coordination number, i.e. the average number of contacts per particle is chosen to  
8 study the homogeneity of the sample, because of its simplicity and its physical significance in  
9 representing the density of the sample. With increase in the coordination number, the density  
10 of the granular assembly increases. The mechanically stable sample prepared by ballistic  
11 deposition technique presented in ?? is used for the statistical analysis. Representative elements

of size  $0.02 \text{ m} \times 0.01 \text{ m}$  (highlighted segments in ??) were considered. Each representative element has approximately 40 particles. A histogram is plotted showing the number of particles having a particular coordination number for all the representative elements ( ??). A normal distribution of the number of particles and the coordination number can be observed from ??.

Most particles are found to have 2 or 3 contacts with its neighbours. The number of particles 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 particles.

### 2.2.5 Voronoi Tessellation

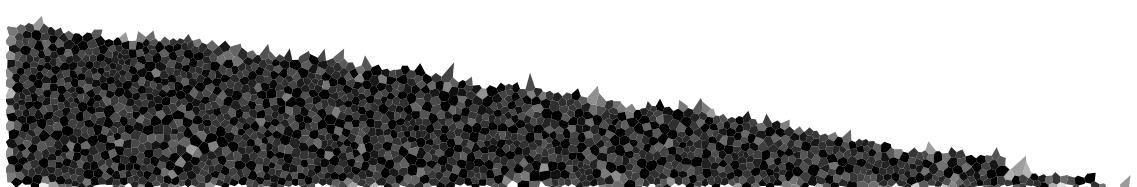
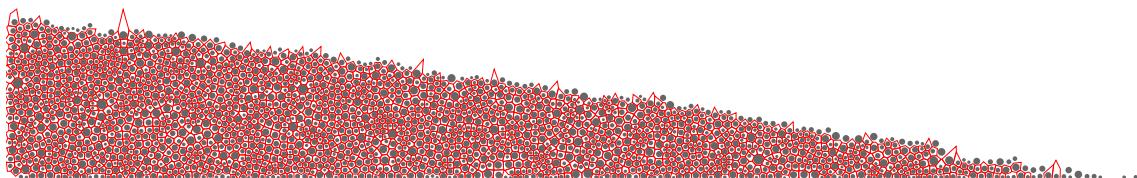


Figure 2.7 Voronoi tessellation to average bulk properties

# Chapter 3

13

## Multi-scale modelling of dry granular flows

14

15

### 3.1 Introduction

16

In nature, instabilities of slopes or cliffs are dramatic events involving sudden release of a large mass of soil. However, the prediction of catastrophic events still represents challenge, one difficulty being our incomplete understanding of the dynamics of granular flows (Rondon et al., 2011). Understanding the mechanics is of particular importance for risks assessment. Small scale laboratory experiments are usually unable to properly capture the dynamics of geophysical events. However, they can be useful to precisely study physical mechanisms, which may play a role in real flows (Iverson, 1997).

17

18

19

20

21

22

23

24

Conventionally, granular materials such as soils are modelled as a continuum. On a macroscopic scale, granular materials exhibit many collective phenomena and the use of continuum mechanics to describe the macroscopic behaviour can be justified. However, on a grain scale, the granular materials exhibit complex solid-like and/or fluid-like behaviour depending on how the grains interact with each other. Numerical studies at grain scale allows a precise understanding of the internal flow structure. Recent works on granular materials suggest that a continuum law may be incapable of revealing in-homogeneities at the grain-scale level, such as orientation of force chains, which are purely due to micro-structural effects Rycroft et al. (2009). Discrete Element approaches are capable of simulating the granular material as a discontinuous system allowing one to probe into local variables such as position, velocities, contact forces, etc. The fundamental question is how to model granular materials which exhibit complex phenomenon. It is important to understand the mechanics of granular flows and the ability and limitations of continuum methods in capturing the flow dynamics.

### <sup>13</sup> 3.2 Granular column collapse

<sup>14</sup> The collapse of a granular column, which mimics the collapse of a cliff, has been extensively  
<sup>15</sup> studied in the case of dry granular material (Hogg, 2007; Kerswell, 2005; Lajeunesse et al.,  
<sup>16</sup> 2004; Lo et al., 2009; Lube et al., 2005; Staron and Hinch, 2007; Zenit, 2005). The granular  
<sup>17</sup> column collapse experiment involves filling a rectangular channel of height  $H_0$  and width  $L_0$   
<sup>18</sup> with a granular material of mass ‘m’ (see figure 3.1). The granular column is then released  
<sup>19</sup> *en masse* by quickly removing the gate, thus allowing the granular material to collapse onto  
<sup>20</sup> the horizontal surface, forming a deposit having a final height  $H_f$  and length  $L_f$ . Despite the  
<sup>21</sup> complexity of the intermediate flow dynamics, experimental investigations have shown that the  
<sup>22</sup> flow evolution, the spreading velocity, the final extent of the deposit, and the energy dissipation  
<sup>23</sup> can be scaled in a quantitative way independent of the substrate properties, grain size, density,  
<sup>24</sup> and shape of the granular material and released mass (Lajeunesse et al., 2005; Lube et al., 2005;  
<sup>25</sup> Staron and Hinch, 2007). The granular collapse has also been studied using discrete element  
<sup>26</sup> method, which allows precise measurement of the internal flow structure (Lo et al., 2009;  
<sup>27</sup> Staron and Hinch, 2006, 2007; Utili et al., 2014). Power laws relating the final run-out and  
<sup>28</sup> height to the initial aspect ratio of the column were observed. These findings immediately pose  
<sup>29</sup> the question: are these simple scaling fortuitous, an oversimplification, or in fact indicative of a  
<sup>30</sup> simple dynamical balance?

<sup>31</sup> Granular flows are conventionally modelled as a frictional dissipation process in continuum  
<sup>32</sup> mechanics but the lack of influence of inter-particle friction on the energy dissipation and  
<sup>33</sup> spreading dynamics (Lube et al., 2005) is surprising. However, Kerswell (2005) showed the run-  
<sup>34</sup> out behaviour has a clear material dependence, which corroborates the conclusion of Lajeunesse  
<sup>35</sup> et al. (2004) and softens that of Lube et al. (2005). The collapse of a granular column on a  
<sup>1</sup> horizontal surface is a simple case of granular flow, however a proper model that describes  
<sup>2</sup> the flow dynamics is still lacking. Simple mathematical models based on conservation of  
<sup>3</sup> horizontal momentum capture the scaling laws of the final deposit, but fail to describe the initial  
<sup>4</sup> transition regime. From a theoretical point of view, the spreading has been described using  
<sup>5</sup> depth averaged equations (Kerswell, 2005; Larrieu et al., 2006). The depth-averaged and Saint-  
<sup>6</sup> Venant equations struggle to recover the precise dynamic behaviour of the system (Warnett  
<sup>7</sup> et al., 2013) and only succeeds in predicting the scaling observed for aspect ratio less than one.  
<sup>8</sup> However, describing larger aspect ratio and capturing the initial stage of the collapse, when  
<sup>9</sup> the grains experience a rapid change of direction from vertical to horizontal, remain an open  
<sup>10</sup> challenge,

<sup>11</sup> In the present study, multi-scale numerical modelling, i.e. grain-scale modelling and  
<sup>12</sup> continuum analyses, of the quasi-two dimensional collapse of granular columns are performed  
<sup>13</sup> using Discrete Element (DEM) approach and Generalised Interpolation Material Point Method

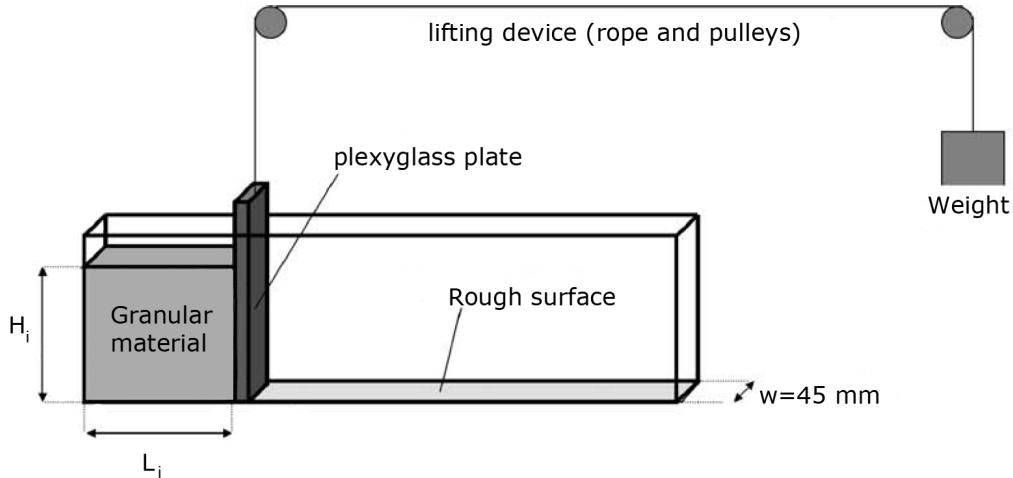


Figure 3.1 Schematic of experimental configuration for 2-D collapse in a rectangular channel, ([Lajeunesse et al., 2004](#))

(GIMPM). GIMPM, a hybrid Eulerian – Lagrangian approach, with Mohr-Coloumb failure criterion is used to describe the continuum behaviour of granular column collapse. While the micro-mechanics of the flow is captured using DEM simulations. Comparing the grain scale behaviour with the continuum simulations highlights the limitations of continuum approaches in modelling the dense granular flows and their ability (or lack thereof) in capturing the complex micro-scale rheology.

14

### 3.2.1 Numerical set-up

In this study, numerical simulations of granular columns are analogous to the experimental investigation of column collapse performed by [Lajeunesse et al. \(2004\)](#). The experimental configuration of [Lajeunesse et al. \(2004\)](#) is shown in figure 3.1. Granular material of mass ' $M$ ' was poured into a container to form a rectangular heap of length ' $L_0$ ', height ' $H_0$ ' and thickness ' $W$ '. The internal friction angle and the wall friction between the wall and the glass beads measured by [Lajeunesse et al. \(2004\)](#) are listed in table 3.1. The gate was then quickly removed to release the granular mass that spreads in the horizontal channel until it comes to rest. The final run-out distance ' $L_f$ ' and the collapsed height ' $H_f$ ' were measured. The run-out distance and collapse height exhibit a power law relation with the initial aspect ratio ' $a$ ' ( $= H_0/L_0$ ) of the column.

Granular materials when released suddenly on a horizontal surface exhibit transient flow. In this study, the mechanism of flow initiation, spreading dynamics and energy dissipation are studied for varying initial aspect ratios of the granular column. DEM soil grain characteristics match that of the experiment. The particle size distribution (PSD) is one of the most impor-

Table 3.1 Material properties of glass ballotini ([Lajeunesse et al., 2004](#))

Parameter	Value
Mean diameter	1.15 mm
Repose angle	$22 \pm 0.5^\circ$
Avalanche angle	$27.4 \pm 0.5^\circ$
Wall friction angle	$24.8 \pm 0.2^\circ$

tant factors controlling landslide initiation and soil permeability. Cumulative  $\beta$  distribution (described in ??) is used to generate a graded sample with a mean grain diameter of 1.15mm (see figure 3.2b). The DEM sample is composed of  $\sim 3000$  disks with a uniform distribution of diameters by volume fractions in the range  $[d_{min}, d_{max}] = 0.92 - 1.38$  mm with polydispersity  $r = \frac{d_{max}}{d_{min}} = 1.5$ . The granular column is prepared by allowing the randomly placed grains to undergo ballistic deposition with a constant potential head between layers of soil grains. A snapshot of the sample generated is shown in figure 3.2a. A DEM sample with soil grains arranged in a regular hexagonal lattice is also used to study the influence of crystallisation and jamming on the run-out behaviour.

The overlap between grains are determined by the stiffness  $k_n$  of the spring in the normal direction. Typically, an average overlap in the range 0.1 to 1.0% is desirable [Zenit \(2005\)](#) and the spring constant is chosen to produce grain overlaps in this range. The stiffness is determined as

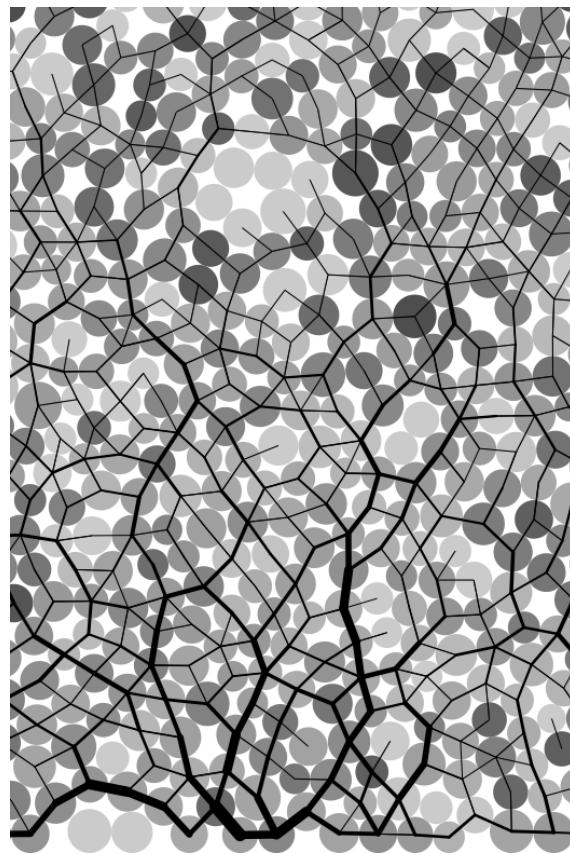
$$k_n = \frac{2\pi G}{(1-\nu)[2\ln(\frac{2r}{A}) - 1]} \quad (3.1)$$

$$A = \left[ \frac{2r(1-\nu)f_n}{\pi G} \right]^{\frac{1}{2}}, \quad (3.2)$$

where  $f_n$  is the normal contact force;  $G$  is the shear modulus;  $\nu$  is the Poisson's ratio and  $r$  is the radius of the grain. A simpler form of stiffness for a spherical grain is defined as

$$k_n = 4ER, \quad (3.3)$$

where  $E$  is the Young's modulus of the material and  $R$  is the radius of the grain. [Cambou et al. \(2009\)](#) observed that the contact model has negligible influence on the run-out behaviour of rapid granular flows. The granular collapse simulations performed using non-linear Hertz-Mindlin contact model and the linear-elastic contact model showed no significant difference on the granular flow behaviour [Utili et al. \(2014\)](#). Linear-elastic contact model is used in the



(a) DEM sample prepared using ballistic deposition

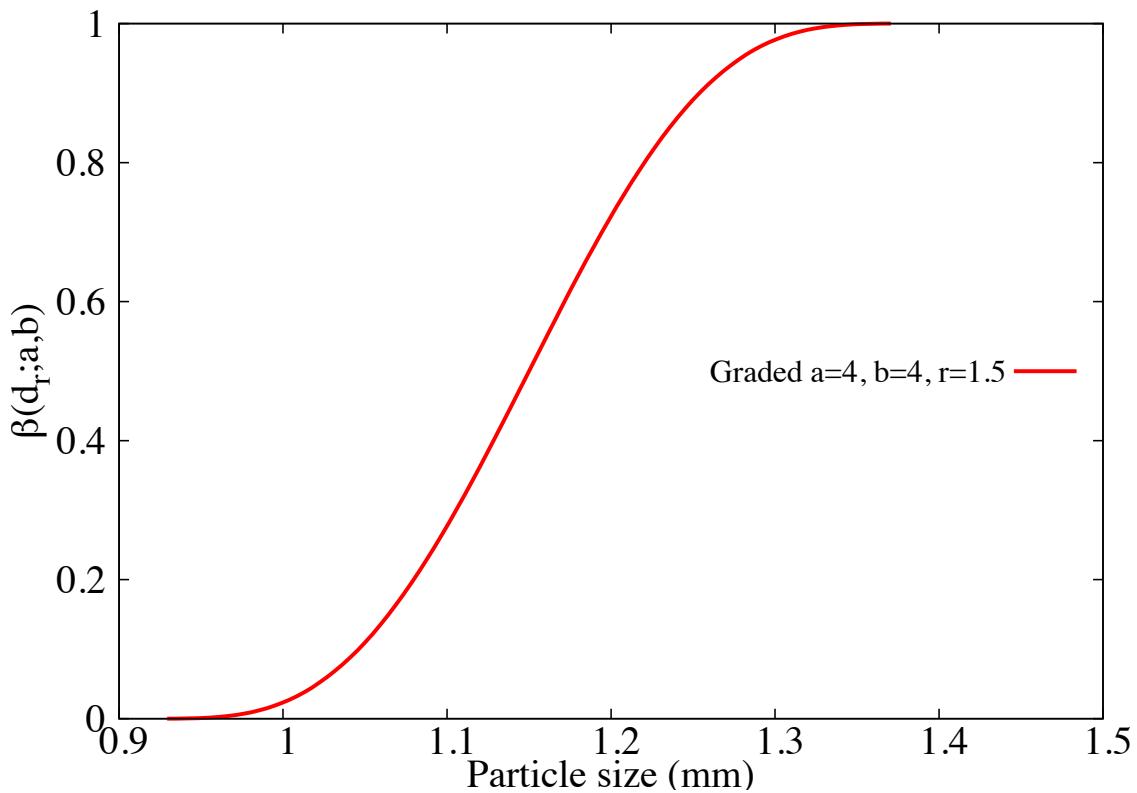
(b) DEM grains generated using the cumulative  $\beta$  distribution

Figure 3.2 DEM sample characteristics

Table 3.2 Micro-mechanical parameters used in DEM simulations

Parameter	Value
Young's modulus of glass bead	$70 \times 10^9 \text{ N m}^{-2}$
Poisson's ratio	0.22 - 0.24
Diameter of glass beads	0.92 to 1.38 mm
Normal and shear stiffness of grains	$1.6 \times 10^8 \text{ N m}^{-1}$
Normal and shear stiffness of wall	$4 \times 10^8 \text{ N m}^{-1}$
Inter-particle friction coefficient, $\mu$	0.53
Wall friction coefficient	0.466
Coefficient of restitution, $\epsilon$	0.755

present study due to its simplicity and lower computation time requirement. The maximum tangential force is limited by the Mohr-Coloumb criterion.

Staron and Hinch (2006) observed that the coefficient of restitution  $\epsilon$  was dramatically changing the behaviour of the systems for  $\epsilon \rightarrow 1$ ; in particular, this dramatic change is expected to become more important for increasing values of  $a$ . On the contrary, for  $\epsilon \leq 0.8$ , the influence of the coefficient of restitution becomes negligible. In the present study, a value of 0.75 is adopted as the coefficient of restitution, similar values of restitution coefficient was adopted by Girolami et al. (2012); Zenit (2005). The normal damping coefficient  $C_n$  is appropriately chosen to achieve the required coefficient of restitution  $\epsilon$ :

$$C_n = 2\gamma\sqrt{m_{ij}k_n} \quad (3.4)$$

$$\text{where } \gamma = -\frac{\ln(\epsilon)}{\sqrt{\pi^2 + \ln^2(\epsilon)}}, \quad \text{and} \quad m_{ij} = \frac{m_i m_j}{m_i + m_j}. \quad (3.5)$$

The micro-mechanical parameters used in this study are presented in table 3.2. Due to the unsteady nature of the flow, the grains get dispersed on the horizontal plane as discrete bodies start to separate from the main mass, hence the run-out distance is calculated as the position of the farthest grain which has at least one contact with the main mass.

GIMPM with Mohr-Coloumb constitutive model is used to simulate plane strain collapse of granular columns. Crosta et al. (2009) observed that the Mohr-Coloumb with non-associate flow rule is able to capture granular collapse dynamics and models the strong vertical motion components, but it does not suffer the limitations of typical shallow water equation methods. In order to understand the ability and limitations of continuum approaches in capturing the local rheology, it is important to scale the grain scale properties, such as inter-particle friction and stiffness, to the continuum scale (macroscopic friction and Young's modulus). Crosta et al. (2009) observed that the friction angle plays a significant role on the run-out behaviour. In

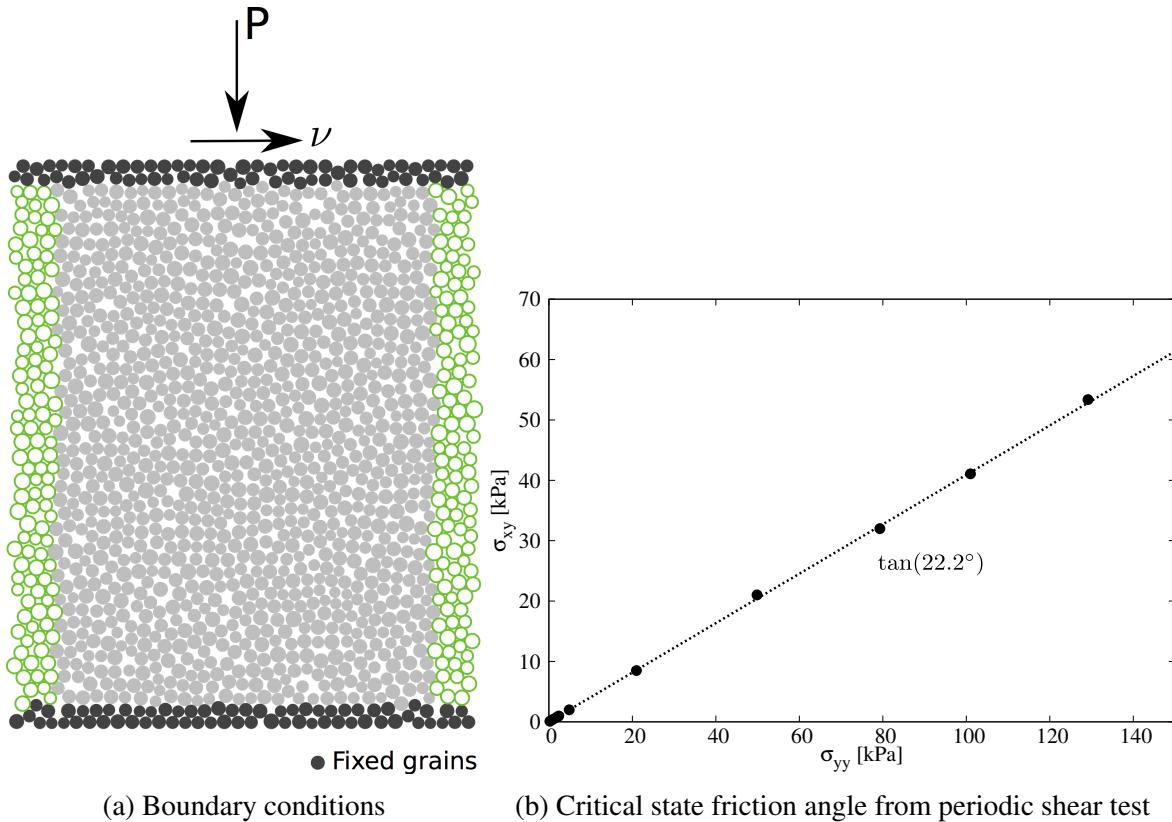


Figure 3.3 Periodic shear test

MPM simulations, the granular flow is assumed to be in critical state and the critical state friction angle is used in the Mohr-Coloumb model. In order to obtain the critical state friction angle of the granular sample, a shear test is performed using 1078 DEM grains. A bi-periodic

<sub>1</sub> boundary condition is adopted on the sides of the sample (see figure 3.3a). Two layers of fixed  
<sub>2</sub> grains (shown in black) is placed at the top and the bottom of the shear sample. A normal  
<sub>3</sub> pressure 'P' and a horizontal velocity  $v$  is applied to the fixed grains at the top of the shear  
<sub>4</sub> sample. The normal effective stress is varied in the sample and the average shear stress of the  
<sub>5</sub> sample is measured. The sample was sheared until critical state was reached. The slope of  
<sub>6</sub> shear stress versus normal effective stress gives the critical state friction angle. A critical state  
<sub>7</sub> friction angle of  $22.2^\circ$  is obtained. The macroscopic friction angle is in the range observed  
<sub>8</sub> by Estrada et al. (2008); Mitchell and Soga (2005). The Young's modulus of the granular  
<sub>9</sub> assembly is obtained as the initial slope of the stress-strain plot of a uni-axial compression of a  
<sub>10</sub> granular column using DEM.

<sub>11</sub> Guilkey et al. (2003) suggests using at least four material points per cell for large deforma-  
<sub>12</sub> tion problems. In the present study, 16 material points per cell is adopted. If the mesh is too  
<sub>13</sub> fine and the number of particles is too large, the particle size  $2l/p$  decreases, and the GIMPM

12

13

14

Table 3.3 Parameters used in continuum simulations

Parameter	Value
Material point spacing	0.575 mm
Number of material points per cell	16
Young's Modulus, E	$1.98 \times 10^6$ Pa
Poisson's ratio, $\nu$	0.22 to 0.24
Friction angle, $\phi$	$23.2 \pm 0.2^\circ$
Dilatancy angle, $\Phi$	$0^\circ$
Density, $\rho$	$1800 \text{ kg m}^{-3}$
Wall friction	0.466
Time step increment	$1.0 \times 10^{-6}$ s

<sup>14</sup> interpolation function tends to approach the original MPM function, as shown by ?. Hence  
<sup>15</sup> GIMPM loses the merit that it reduces the numerical noise due to material points crossing  
<sup>16</sup> the background mesh. In addition, the probability of particles crossing the background mesh  
<sup>17</sup> increases with decrease in mesh size, hence, more noise can be produced [Abe et al. \(2013\)](#). The  
<sup>18</sup> effect of number of material points per cell on the run-out behaviour is discussed in section [3.3.5](#).  
<sup>19</sup> Each material point represents one-fourth of a DEM soil grain. The parameters used for the  
<sup>20</sup> continuum analyses are presented in table [3.3](#).

### <sup>21</sup> 3.2.2 Deposit morphology

<sup>22</sup> MPM and DEM simulations of granular column collapse are performed by varying the initial  
<sup>23</sup> aspect ratio of the column. The normalized final run-out distance,  $\Delta L = (L_f - L_0)/L_0$ , as a  
<sup>24</sup> function of the initial aspect ratio 'a' of the column is presented in figure [3.4](#). Similar to the  
<sup>25</sup> experimental behaviour a power law relation between the run-out and the initial aspect ratio  
<sup>26</sup> of the column is observed. Two distinct flow regimes can be seen: (a) for 'a' < 1.7 a linear  
<sup>27</sup> relation between the spread and aspect ratio can be observed, and (b) for 'a' > 1.7 a power-law  
<sup>28</sup> relationship exists. In the present study, the following scaling law for the run-out (using DEM)  
<sup>29</sup> is observed:

$$\frac{L_f - L_0}{L_0} \approx \begin{cases} 1.67a, & a \lesssim 2.3 \\ 2.5a^{2/3}, & a \gtrsim 2.3 \end{cases} \quad (3.6)$$

Both, MPM and DEM simulations are able to capture the linear relationship for 'a' < 1.7,  
<sup>10</sup> and the simulation results agree with the experimental investigation [Lajeunesse et al. \(2005\)](#).  
<sup>11</sup> This shows that a simple frictional dissipation model is able to capture the flow dynamics for  
<sup>12</sup>

columns with smaller aspect ratio. For ‘a’ < 1.7, the normalised run-out distance predicted using DEM simulations are very close to the run-out observed in the experiments. DEM simulations with hexagonal packing shows shorter run-out distances in comparison to randomly packed sample. This difference in the run-out behaviour might be due to the crystallisation and jamming effects in hexagonal packing. The small difference in the final run-out between DEM and experimental results can be attributed to the variation in the packing of grains. Also, the experimental data corresponds to granular column collapse in a rectangular channel, the collapse is not a pure two-dimensional collapse as in the case of numerical simulations.

Significant difference in the final run-out between MPM, which is based on a simple frictional model for dissipation of potential energy, and DEM simulations for ‘a’ > 1.7 indicates a change in the mechanism of energy dissipation for columns with large aspect ratios (‘a’ > 1.7). [Staron and Hinch \(2005\)](#) observed that a constant frictional dissipation model cannot describe a power-law relation observed at large aspect ratio. A transition in the run-out behaviour at an aspect ratio of 1.7 indicates a change in flow dynamics. Similar behaviour in the run-out distance was observed by [Bandara \(2013\)](#) for columns with large the aspect ratio  $\geq 2$ .

The longer run-out distance in MPM simulations at large aspect ratios might be influenced by the amount of material mobilised during the collapse. In tall columns, the entire column participates in the flow, in contrast to short columns where the collapse is due to avalanching of flanks, [Lajeunesse et al. \(2004\)](#). It is possible that MPM simulations collapses more resulting in longer run-out distance. Figure 3.5 shows the normalized final height as a function of the initial aspect ratio of the column. Similar to the run-out behaviour, the normalised-height also shows two distinct regimes. The scaling of final height of the column with the initial aspect ratio of the column can be written as

$$\frac{H_f}{L_i} \propto \begin{cases} a, & a \lesssim 0.7 \\ a^{2/3}, & a \gtrsim 0.7 \end{cases} \quad (3.7)$$

The final height predicted by both DEM and MPM simulations match the experimental data for columns with smaller aspect ratio (‘a’  $\leq 0.7$ ). Linear relationship between the final height and the aspect ratio indicates that only a part of the granular column is mobilised during the collapse. For tall columns, both approaches predict similar normalised height. However, the normalised height observed in MPM is higher than in DEM simulations, which is in contrast to the idea of increase in the amount of material mobilised during the collapse in MPM simulations resulting in longer run-out distance. Hence, the longer run-out observed in MPM simulations is due a change in the flow dynamics at higher aspect ratios, which is not captured in MPM simulations. The final height of a column is controlled by the amount of static region in the

granular column collapse, while the run-out distance is essentially a function of the flowing mass. Hence, it is essential to compare the evolution of flow and the internal flow structure in DEM and MPM simulations.

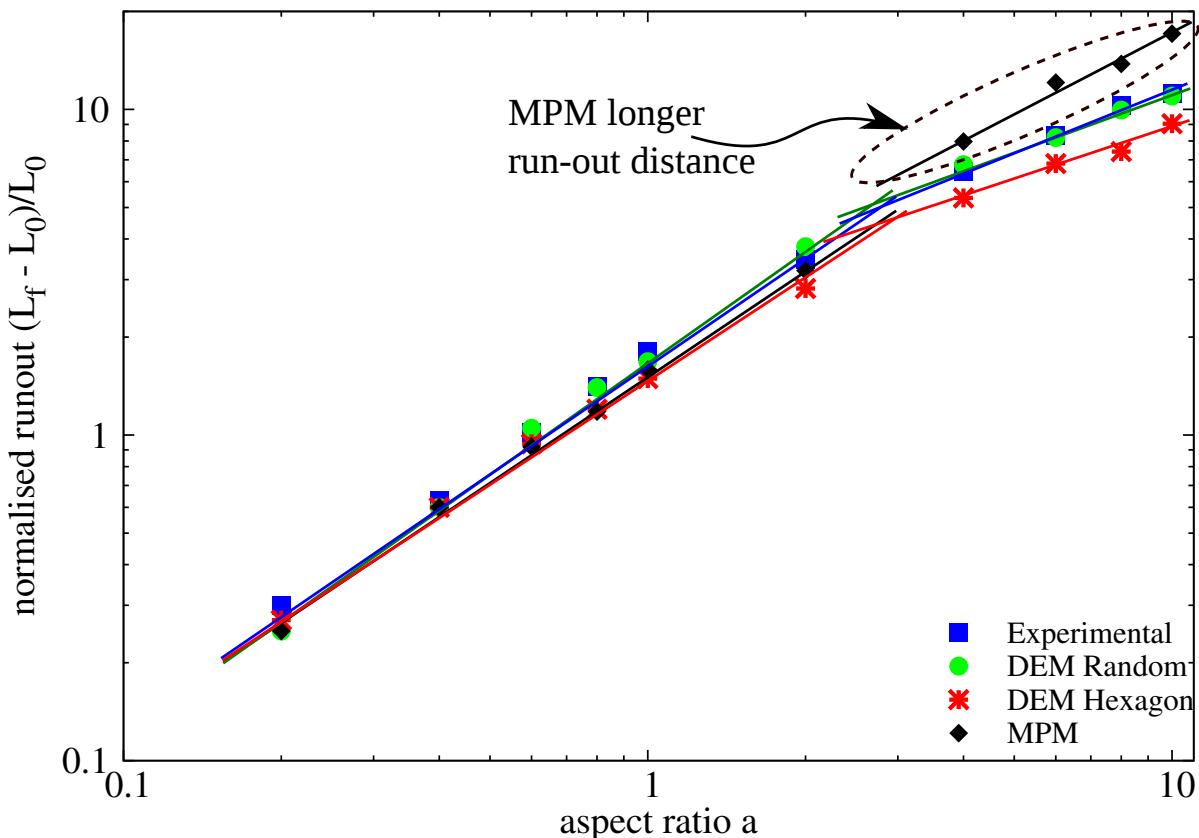


Figure 3.4 Normalised final run-out distance for columns with different initial aspect ratio

### 3.2.3 Flow evolution and internal flow structure

- 1 The normalised run-out and height as a function of the aspect ratio indicates that, for a given
- 2 granular material and substrate properties, the flow dynamics and the final deposit morphology
- 3 are independent of the volume of granular material released, but depend only on the geometry
- 4 of the column. A power law relationship is observed between the run-out distance and the
- 5 initial aspect ratio of the column. A transition in the run-out behaviour at an aspect ratio of 2.3
- 6 indicates a change in the flow dynamics.

For smaller aspect columns ('a' < 2.3), the flow is initiated by a failure at the edge of the pile along a well-defined fracture surface. The granular mass fails through avalanching of flanks producing a truncated cone-like deposit ('a' < 0.7) or conical deposit ('a' > 0.7). The

## 3.2 Granular column collapse

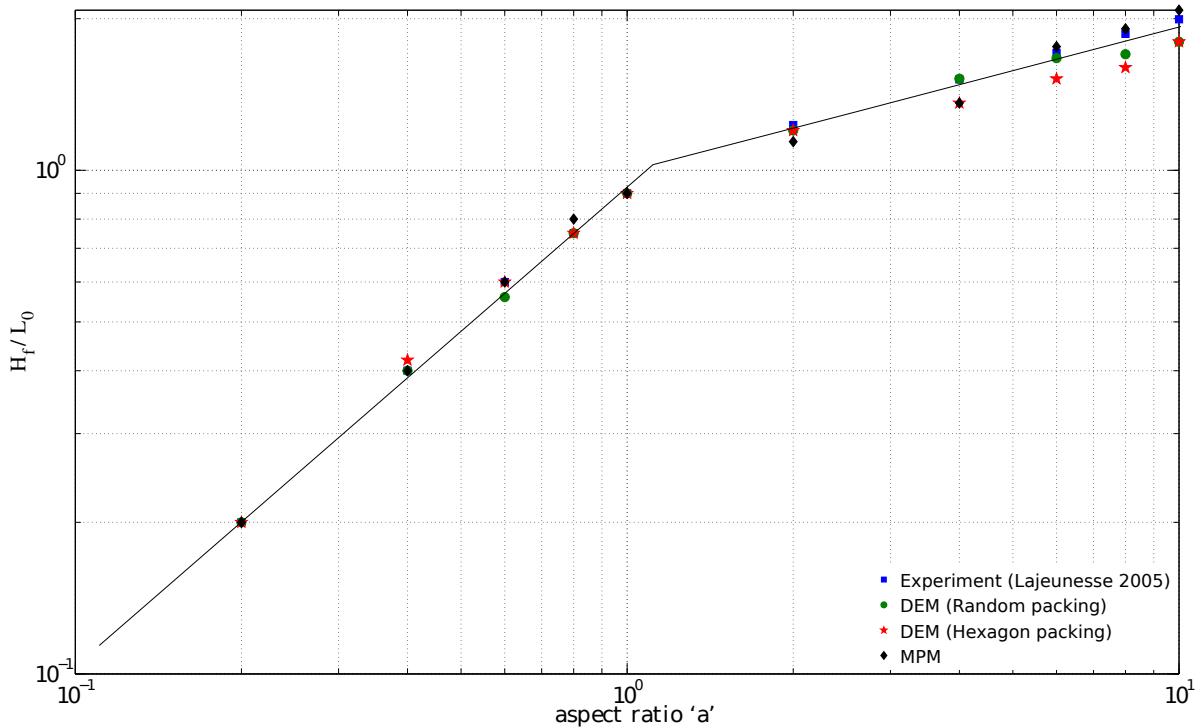


Figure 3.5 Normalised final collapse height for columns with different initial aspect ratio

10 grains located above the failure surface move “*en masse*” leaving a static region underneath the  
11 failure surface.

12 Dimensional analysis of granular column collapse reveals an intrinsic time defined as  
13  $\sqrt{H_i/g}$ . This intrinsic time is a transient time of order  $\tau_c$ , at which the flow is fully developed,  
14 i.e., the potential energy available at the initiation of collapse is now fully converted to kinetic  
15 energy. Numerical simulation of the velocity profile of a granular column ( $'a'=0.4$ ) at critical  
16 time  $\tau_c$  is presented in figure 3.6. At critical time, the velocity field depends only on the  
17 position of the grain along the sliding mass. The maximum velocity is observed at the front of  
18 the flowing mass corresponding to that of a plug flow in horizontal direction. Particulate and  
19 continuum simulations show similar run-out distance at the critical time. Both approaches show  
20 similar quantity of material destabilised above the failure surface. However, the crystalline  
21 arrangement of soil grains in a hexagonal packing results in a different flow mechanics, which  
22 also shows the effect of jamming at the flow front. The continuum nature of MPM results in a  
23 slightly different geometry of the material destabilised above the failure surface in comparison  
24 to DEM simulations. The velocity profile is similar to a steady granular surface flow observed  
25 by [Lajeunesse et al. \(2004\)](#).

26 For columns with lower initial aspect ratios, the run-out distance is proportional to the mass  
27 flowing above the failure surface. The spreading results from a Coulomb-like failure of the edges

and implies no free fall of the column. [Daerr and Douady \(1999\)](#) also observed active Coulomb yielding in transient granular surface flows. In this case, the effective friction properties of the flow can be simply predicted from the shape of the final deposit. The amount of mass mobilized during the collapse is significantly affected by the angle of the failure surface. Figure 3.6 shows that both numerical techniques predict a distinct failure surface when the flow is fully developed at critical time  $\tau_c$ . The angle of the failure surface is found to be about  $55^\circ$ . The failure surface begins from the toe of the column and protrudes inwards at an angle of 50 to  $55^\circ$ . The formation of the “truncated conical deposit” or “conical deposit” depends only on the initial length of the column, as the angle of the failure surface is found to be independent of the aspect ratio. The failure angle is consistent with the interpretation in terms of *active Coulomb failure* ([Lajeunesse et al., 2004](#)), which leads to a predicted failure angle  $\theta_y = 45^\circ + \delta/2$ , where  $\delta$  is the internal friction angle of the granular material. In the present study, the friction angle of the glass beads is  $22^\circ$ , which leads to  $\theta_y = 45^\circ + 22^\circ/2 = 56^\circ$ , which is in good agreement with the numerical simulations and experimental observations by [Lajeunesse et al. \(2004\)](#). The fracture angle has a direct effect on the transition between the truncated cone and the conical deposit occurring at an aspect ratio of 0.7. [Schaefer \(1990\)](#) observed the onset of instabilities in a narrow wedges of  $56$  to  $65^\circ$  for Cambridge-type constitutive models that describes granular flows, which is in-line with the failure angle observed in the present study.

The final profile of the granular column with an initial aspect ratio of 0.4 is shown in figure 3.7. Both MPM and DEM show similar run-out behaviour. The continuum approach is able to capture the flow dynamics of short columns, wher the failure mechanism is active Coulomb failure. In dense hexagonal packing, the failure surface is steep due to crystallisation effect. The variation in the angle of the failure surface causes a difference in the amount of material destabilised, and in turn in the run-out distance. This crystallisation phenomenon is found to have a significant influence on the final deposit of the granular column. [Lacaze and Kerswell \(2009\)](#) observed that poly-disperse grains have lesser tendency to crystallize especially in the case of tall columns.

For tall columns ( $'a' > 2.3$ ), the flow is still initiated by a well defined failure surface as can be seen in figure 3.8. However, in this case the initial granular column is much higher than the top of the failure surface. Due to gravity most of the grains in the column experience free-fall consuming the column along their way. When they reach the vicinity of the failure surface, the flow gets deviated along the horizontal direction releasing a huge amount of kinetic energy gained during the free fall. For larger aspect ratio ( $a > 0.7$ ), the resulting static region is a cone, the final height of the cone, i.e,  $H_f$  lies above the summit of the failure surface. Hence, a different evolution is observed from that of the axis-symmetric geometry ([Lube et al., 2005](#)), where the final height coincides with the summit of the failure surface forming a truncated

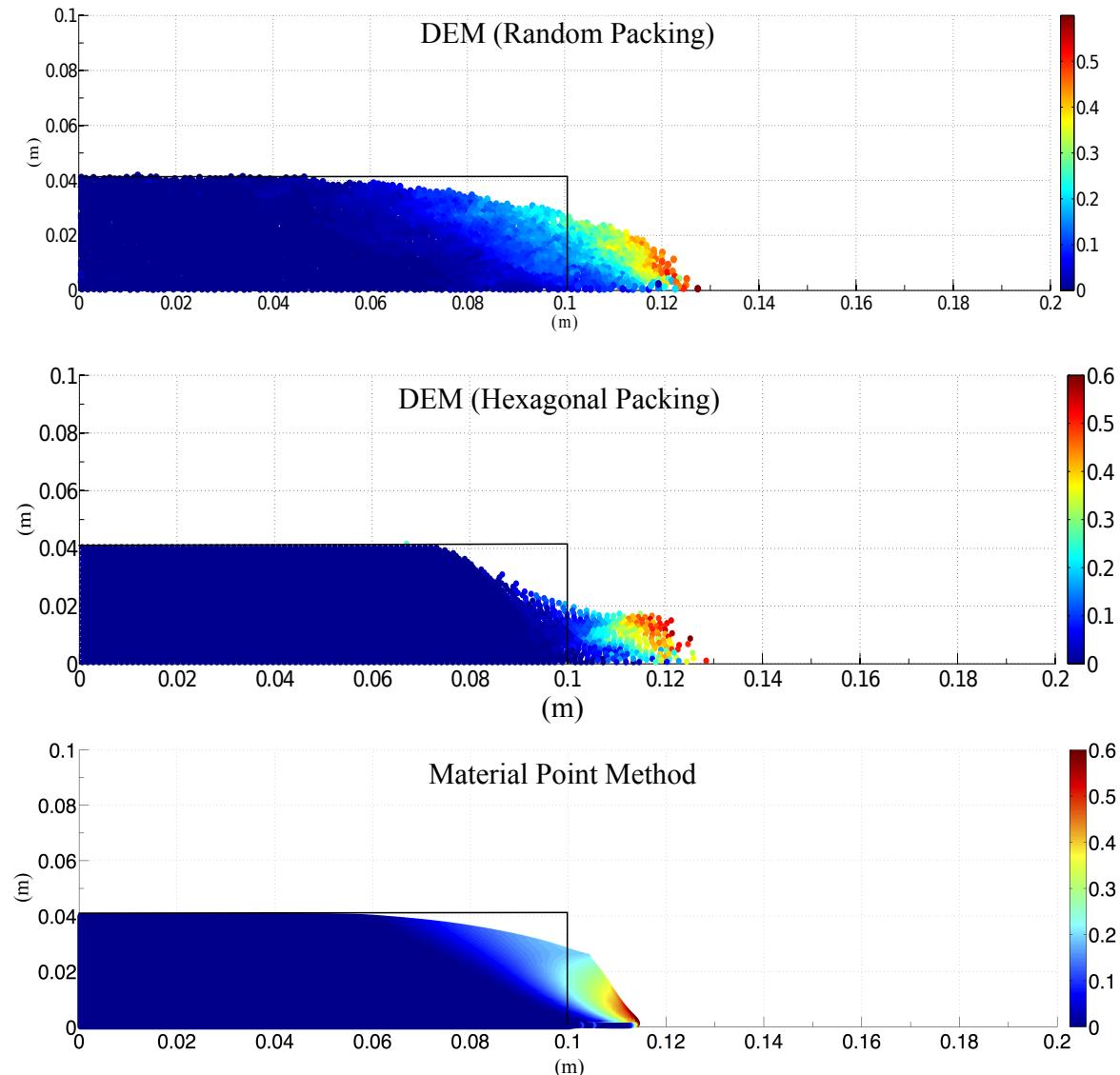


Figure 3.6 Velocity profile of a granular column collapse ( $a' = 0.4$  &  $t = \tau_c$ )

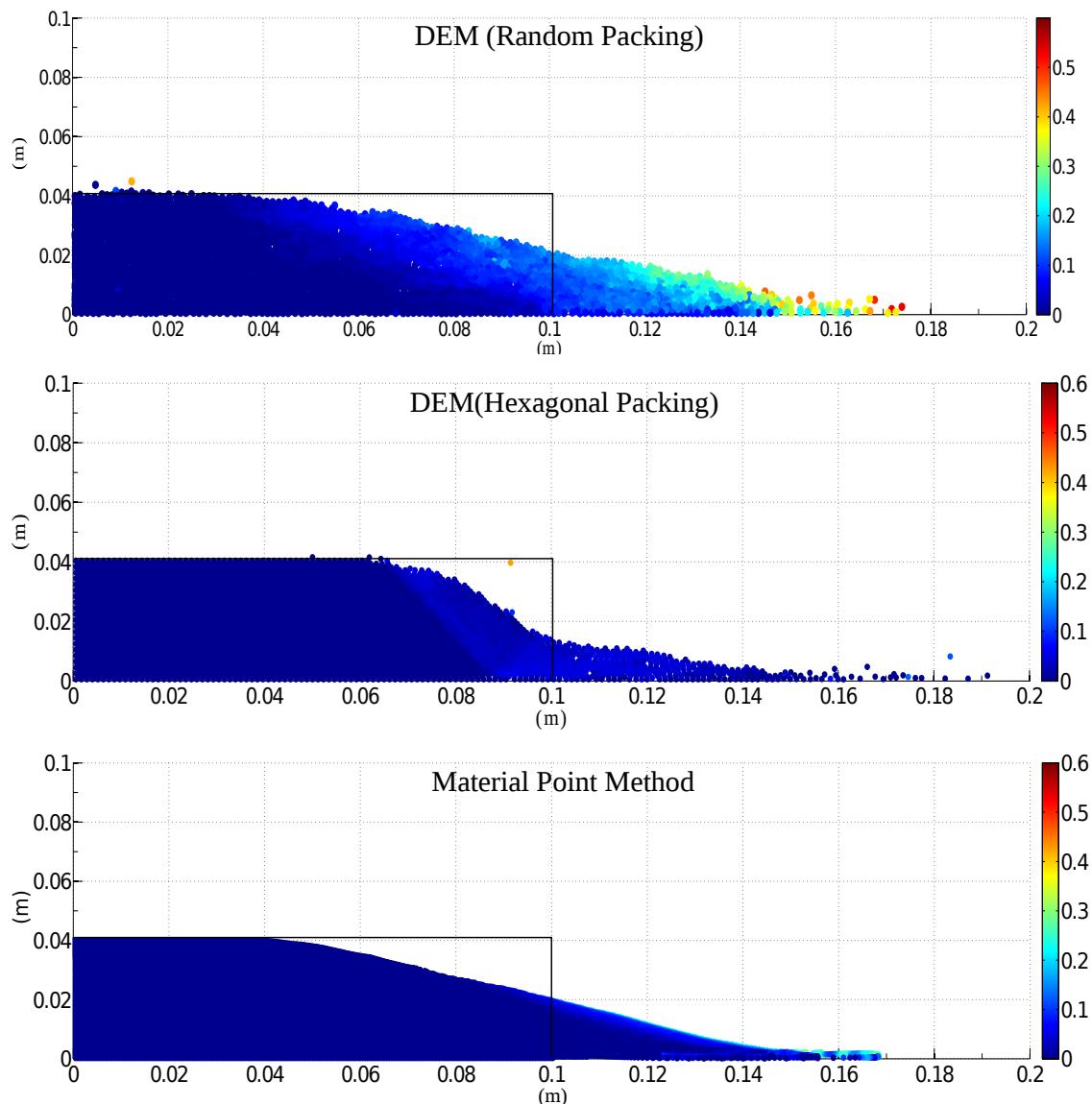


Figure 3.7 Velocity profile of a granular column collapse ( $a' = 0.4$  &  $t = 3 \times \tau_c$ )

conical deposit. Lajeunesse et al. (2004) observed that the variation in the deposit morphology between the axis-symmetric case and the rectangular collapse to be a geometrical effect rather than as an experimental artefact.

An initial failure surface starting from the toe end of the column at an angle of about 55° can be observed at the critical time  $\tau_c$ . As the collapse of the granular collapse progresses, successive failure planes parallel to the initial failure surface are formed and shear failure occurs along these planes. The presence of several shear bands in the final profile of the collapsed granular column confirms this hypothesis. Crystallisation in hexagonal packing has a significant effect on the run-out distance by forming series of parallel shear bands. However, the Material

- 1 Point Method fails to capture the formation of shear bands during the collapse. This observation
- 2 throws light on the mechanics of propagation of shear bands in massive landslides such as
- 3 the Storegga submarine landslide. The flow behaviour becomes similar to that of columns
- 4 with lower aspect ratio as the flow starts descending along the failure plane. The final profile
- 5 of the collapsed granular column with an initial aspect ratio of 6 is presented in Figure 3.9.
- 6 For tall columns, the dissipation process is more complex due to the free-fall dynamics. The
- 7 vertical acceleration of the grains induces a non-trivial mass distribution in the flow while
- 8 spreading. This mass distribution plays a dominant role in the power-law scaling law obeyed
- 9 by the run-out (Staron and Hinch, 2006).

10 Regardless of the experimental configuration and the initial aspect ratio of the columns, the  
 11 flow is initiated by a well-defined rupture surface, above which the material slides down leaving  
 12 a static region underneath the failure plane. Depending on the aspect ratio of the column, two  
 13 asymptotic behaviours are observed. For smaller aspect ratios, the flow is dominated by friction  
 14 where as large aspect ratio columns are influenced by the pressure gradient.

15 To study the influence of aspect ratio on the flow dynamics of granular columns, the flow  
 16 front  $L(t)$  and the maximum height of column  $H(t)$  are tracked. The evolution of scaled height  
 17 ( $H_f/L_0$ ) and the run-out distance  $(L_f - L_0)/L_0$  with time for granular columns with an initial  
 18 aspect ratio of 0.4 and 6 are presented in figure 3.10. Three distinct regions can be observed  
 19 in the flow evolution of granular column collapse regardless of the initial aspect ratio of the  
 20 column. An initial transient acceleration phase is observed for a time  $0.8\tau_c$ . This phase is  
 21 followed by a heap movement of granular materials at the foot with a constant spreading  
 22 velocity  $V$  for about  $2\tau_c$ . When time ‘ $t$ ’ >  $\tau_c$ , the velocity varies linearly with depth in the  
 23 flowing layer and decreases exponentially with depth near the static layer. This velocity profile  
 24 is similar to those observed in steady granular surface flows (Lajeunesse et al., 2004). Most of  
 25 the run-out happens during this phase. The final phase involves deceleration of the flow front  
 26 and the flow comes to rest after  $0.6\tau_c$ . The spreading of the granular column ceases after a time  
 27 in the order of about  $3\tau_c$ , however some motion still persists along the free surface behind the

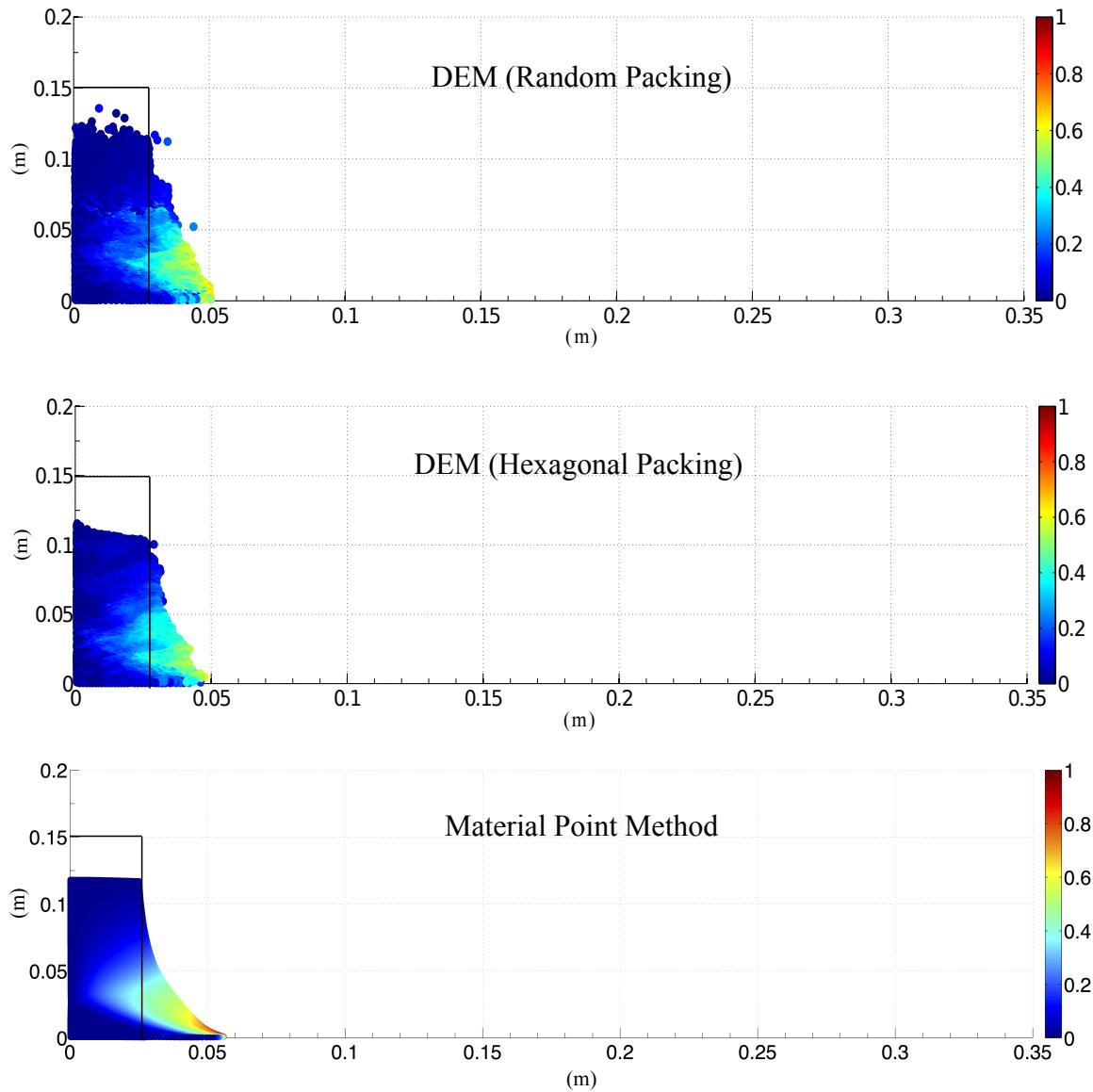


Figure 3.8 Velocity profile of a granular column collapse ( $a' = 6$  &  $t = \tau_c$ )

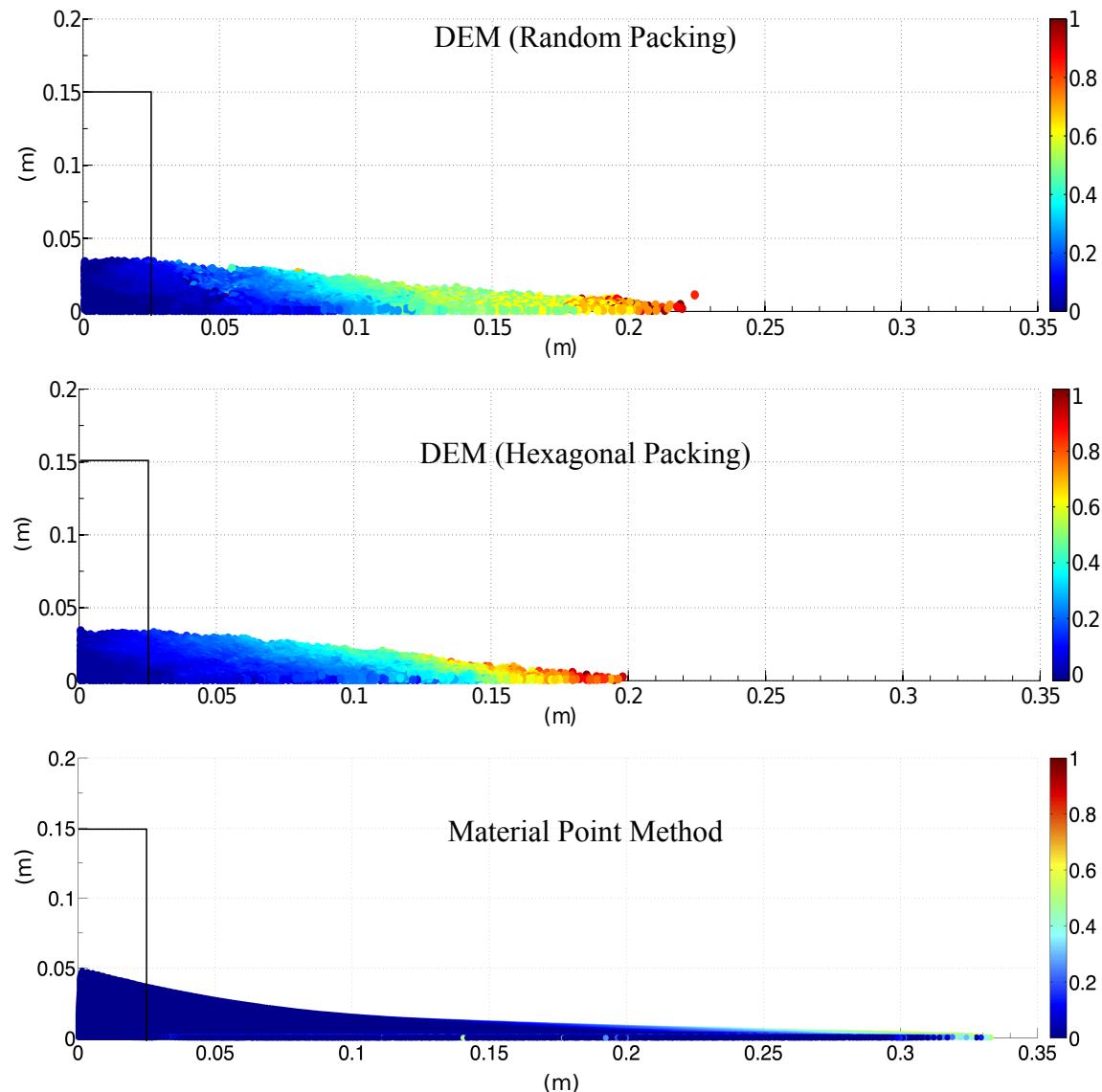


Figure 3.9 Velocity profile of a granular column collapse ( $a' = 6$  &  $t = 3 \times \tau_c$ )

<sup>28</sup> flow front for a much longer time due to internal rearrangement, the duration of which can last  
<sup>29</sup> up to  $t \approx 6\tau_c$ .

<sup>30</sup> For smaller aspect ratios, the critical time is evaluated as the point of intersection of  
<sup>31</sup> the scaled run-out and height. The critical time predicted for both hexagonal and random  
<sup>1</sup> packing of grains matches the experimental observations. However, the Material Point Method  
<sup>2</sup> overestimates the critical time by a factor of 1.25, which means that it takes longer for the  
<sup>3</sup> flow to be fully mobilized. However, the actual run-out duration is short and the granular  
<sup>4</sup> materials comes to rest abruptly at about  $t = 3\tau_c$ . For columns with larger aspect ratios, the  
<sup>5</sup> continuum and particulate approaches simulate similar flow evolution behaviour for times  
<sup>6</sup> up to  $3\tau_c$ , beyond which particulate simulations stabilise and come to rest, while the flow  
<sup>7</sup> continues to evolve in MPM simulations resulting in larger run-outs than expected. The flow  
<sup>8</sup> tends to come to rest at time  $t = 6\tau_c$ . The three phases in a granular flow can be distinctly  
<sup>9</sup> observed in the flow evolution plot for a granular column with initial aspect ratio of 6 (see  
<sup>10</sup> Figure figure 3.10b). For larger aspect ratios, the flow evolution behaviour observed in the  
<sup>11</sup> case of random packing matches the experimental observation by Lajeunesse et al. (2004).  
<sup>12</sup> Hexagonal packing predicts longer time for the flow to evolve, which can be attributed to the  
<sup>13</sup> increase in the internal resistance due to crystallisation of grains. MPM overestimates the  
<sup>14</sup> critical time by 50%, however has the same value of run-out as the particulate simulations, at  
<sup>15</sup> time  $t = 3\tau_c$ , beyond which the material continues to flow until it ceases at  $6\tau_c$ . In order to  
<sup>16</sup> understand the flow dynamics in the case of Material Point Method it is important to study the  
<sup>17</sup> effect of different parameters on the deposit morphology.

### <sup>18</sup> 3.2.4 Energy dissipation mechanism

<sup>19</sup> The time evolution of the flow exhibited three distinct stages during the collapse of a granular  
<sup>20</sup> column. Studying the energy dissipation mechanism provides useful insight into the flow  
<sup>21</sup> dynamics. shows the time evolution of potential energy ( $E_p$ ) and kinetic energy ( $E_k$ ) normalized  
<sup>22</sup> by the initial potential energy  $E_o$ .

$$\text{23} \quad E_p = \sum_{p=1}^{N_p} m_p g h_p \quad (3.8)$$

$$\text{24} \quad E_{ki} = \frac{1}{2} \sum_{p=1}^{N_p} m_p v_p^2 \quad (3.9)$$

<sup>26</sup> where  $N_p$  is the total number of particles,  $m_p$  is the mass of a particle ‘ $p$ ’,  $h_p$  is the height and  
<sup>27</sup>  $v_p$  is the velocity of the particle ‘ $p$ ’. It can be observed from the figure that the initial potential  
<sup>28</sup> energy stored in the particle is converted to kinetic energy which is dissipated as the granular

## 3.2 Granular column collapse

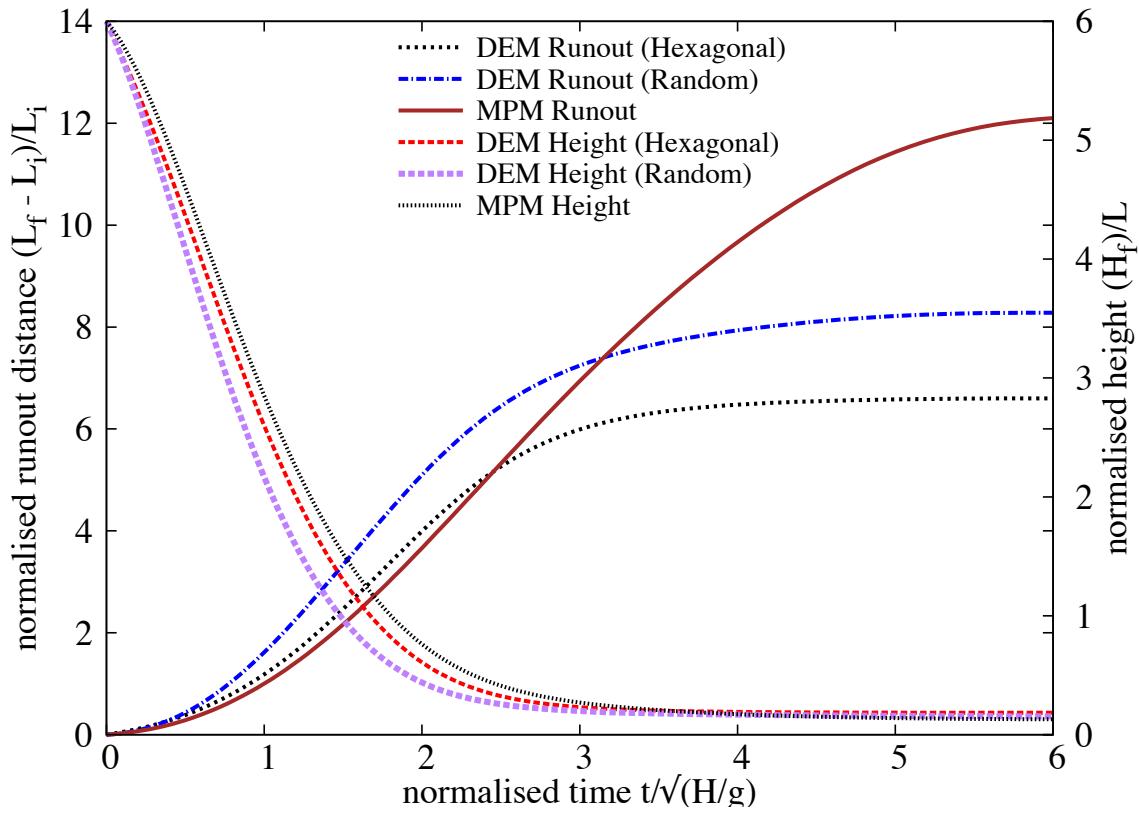
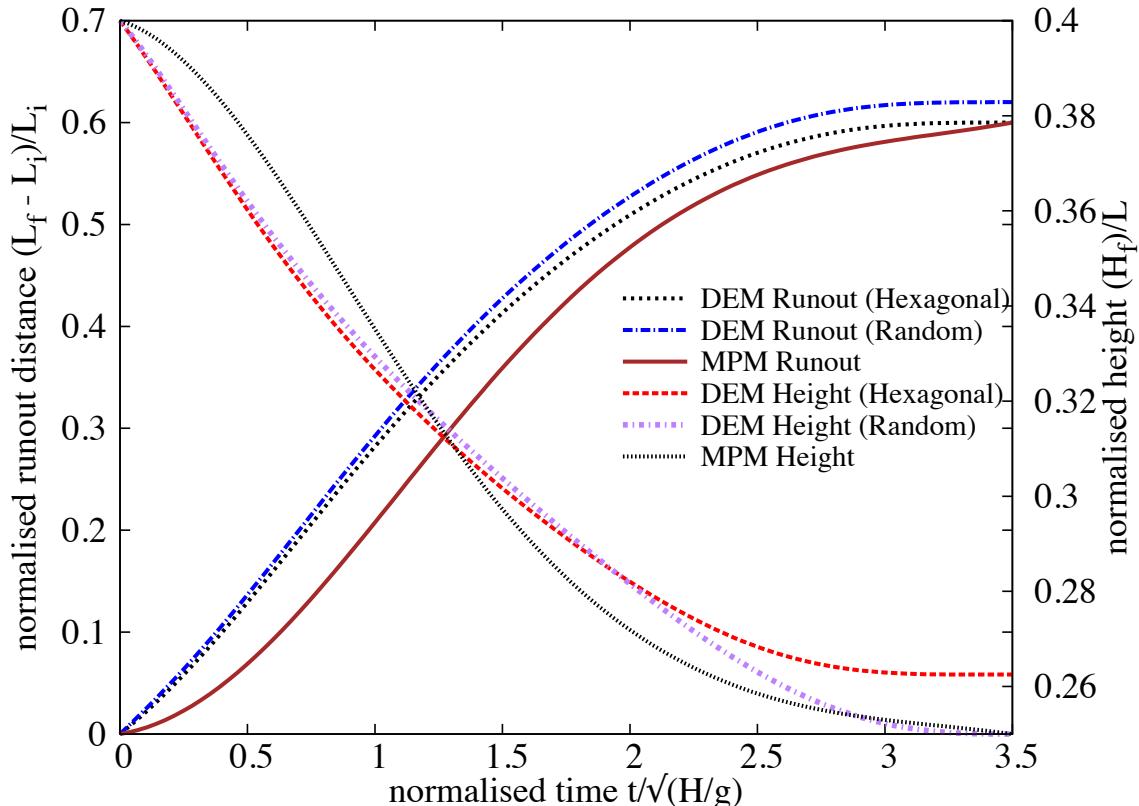


Figure 3.10 Flow evolution of granular column collapse

material flows down. Three successive stages can be identified in the granular column collapse. In the initial acceleration stage ( $t < 0.8\tau_c$ ), the initial potential energy stored in the grains is converted into vertical motion. In the second stage, the grains undergo collisions with the bottom plane and/or with neighbouring grains, and the stored potential energy is converted into horizontal motion. In the third stage, the grains eventually leave the base area of the column and flow sideways. As the process involves collective dynamics of all the particles, it is difficult to predict the exact trajectory of a grain, however, the overall dynamics can be explained. To explain the dissipation of energy during the collapse, [Staron et al. \(2005\)](#) assumed that the total initial potential energy stored in the system is completely dissipated through friction over the entire run-out distance as:

$$\mu m_0 g \times (L_f - L_i) = m_0 g H_o \quad (3.10)$$

where  $\mu$  is the friction coefficient. The model predicts well the flow dynamics for columns with larger aspect ratios, as most of the initial potential energy is dissipated during the collapse involving the entire column. However, for columns with smaller aspect ratios, only a portion of the mass above the failure surface is involved in the flow. Hence, the energy dissipation should involve only the grains lying above the failure surface. A mathematical model, which considers the grains lying above the failure surface, will be derived to predict the flow dynamics of the granular column collapse for different aspect ratios.

### 3.2.5 Role of initial grain properties

[Lube et al. \(2005\)](#) observed that the run-out distance scales with the initial aspect ratio of the column, independent of the material properties. The run-out evolution after the initial transition regime is a frictional dissipation process, and the lack of influence of material properties on the run-out behaviour is inconsistent with continuum modelling of granular flow behaviour. [Balmforth and Kerswell \(2005\)](#) observed that the material properties has almost no influence on the exponent of the normalised run-out as a function of the initial aspect ratio. The numerical constant of proportionality, however, showed clear material dependence. This corroborates the conclusions of [Lajeunesse et al. \(2004\)](#) and softens that of [Lube et al. \(2005\)](#). [Daerr and Douady \(1999\)](#) observed the strong influence of initial packing density and the internal structure on the behaviour of granular flows.

It should be noted that the collapse experiment is highly transient and no clear stationary regime was observed. On the contrary, the acceleration and the deceleration phases cover nearly the whole duration of the spreading. This makes the analysis of the structure of the flow and its relation with other characteristic of the system uneasy. Considering this, we were able

## 3.2 Granular column collapse

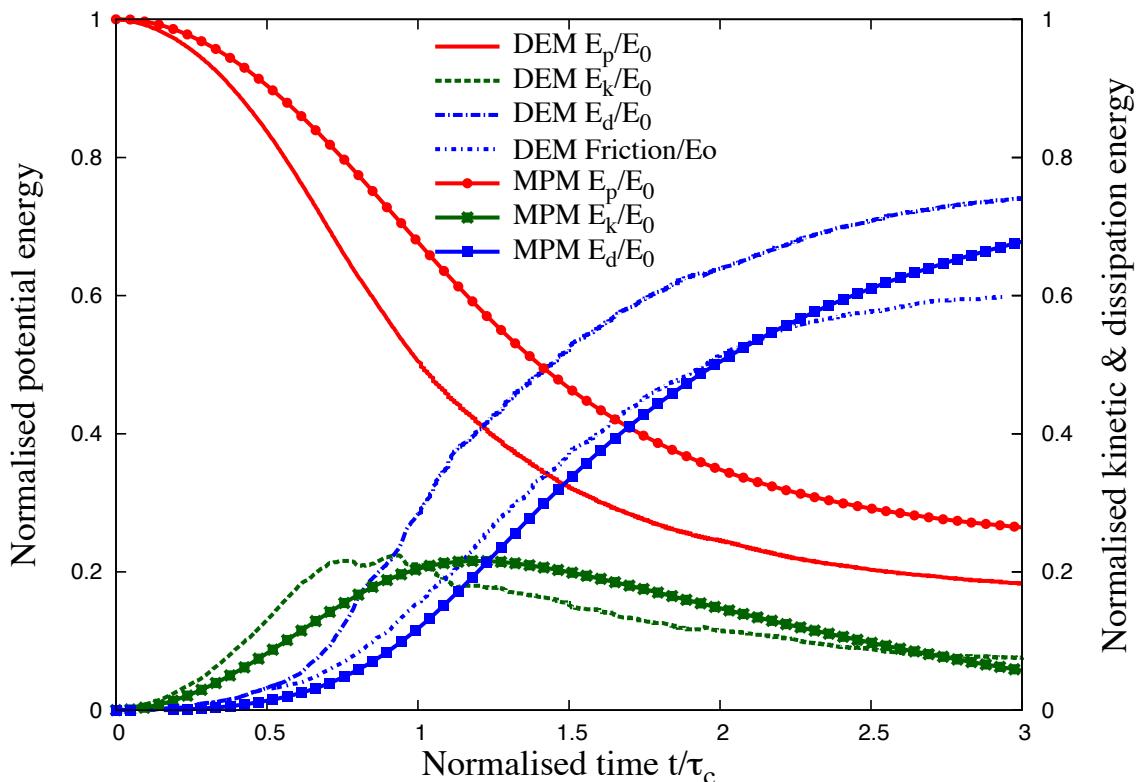
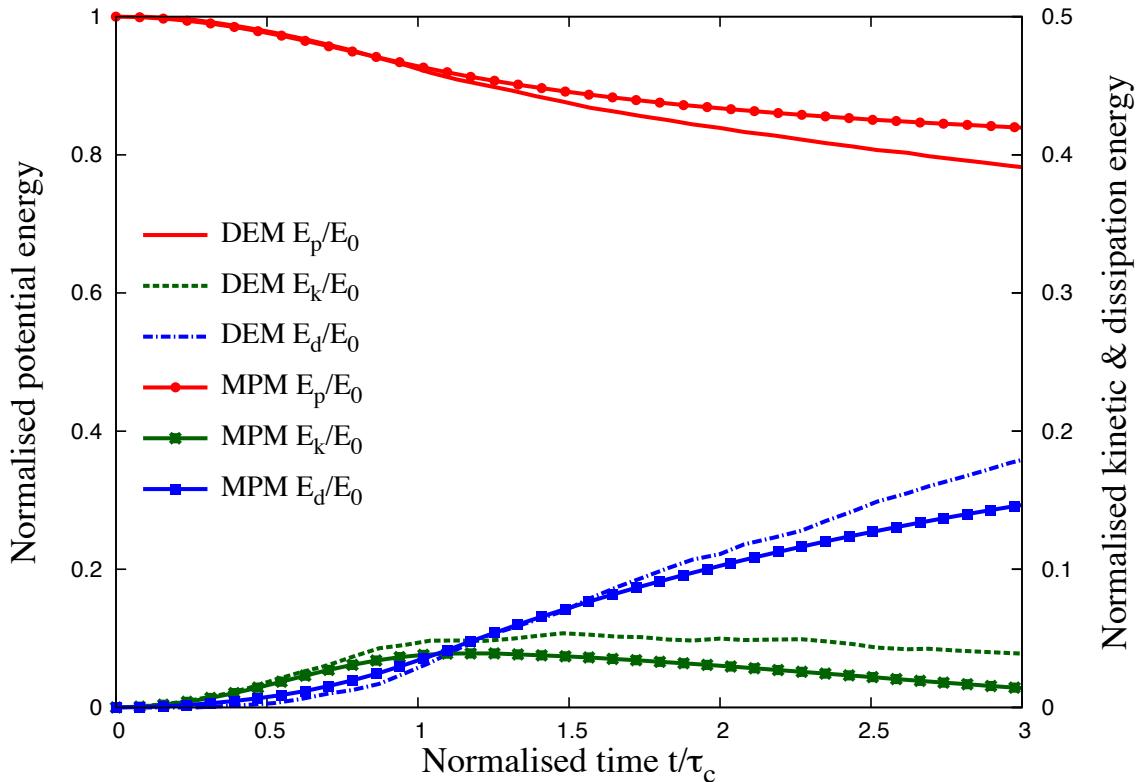


Figure 3.11 Energy evolution of granular column collapse

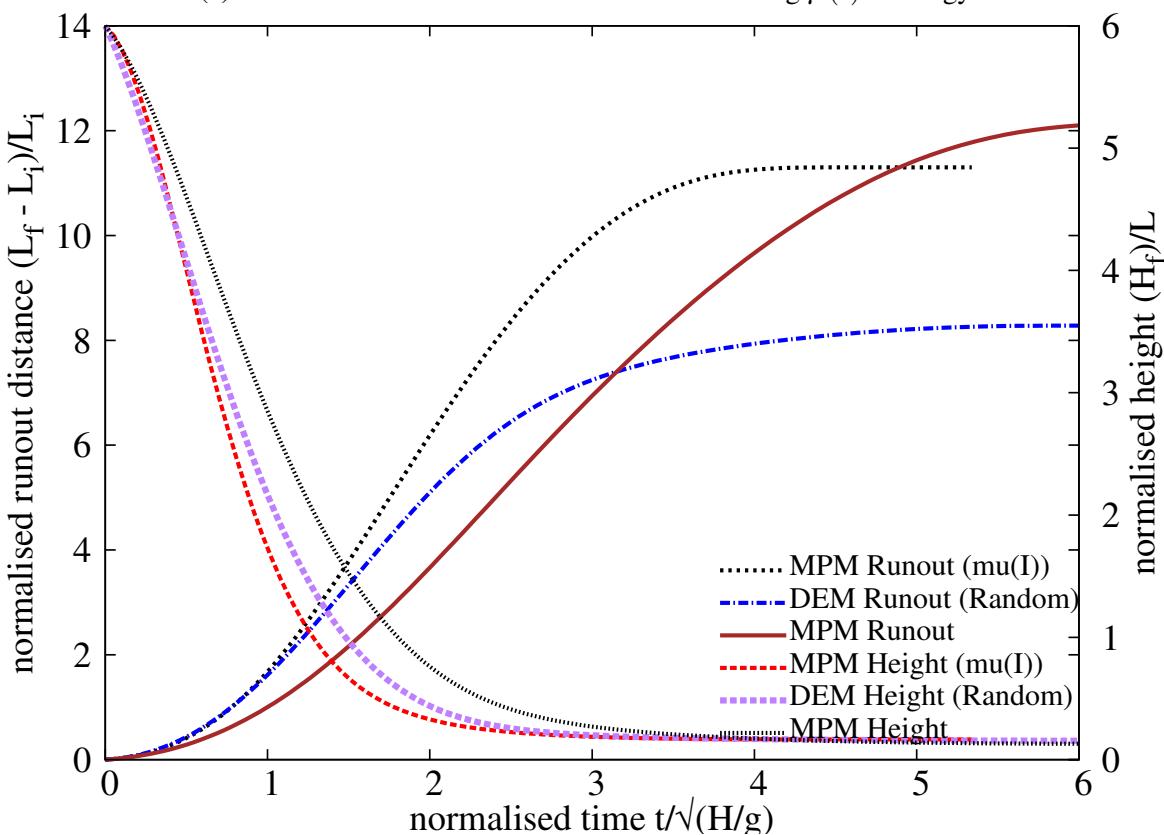
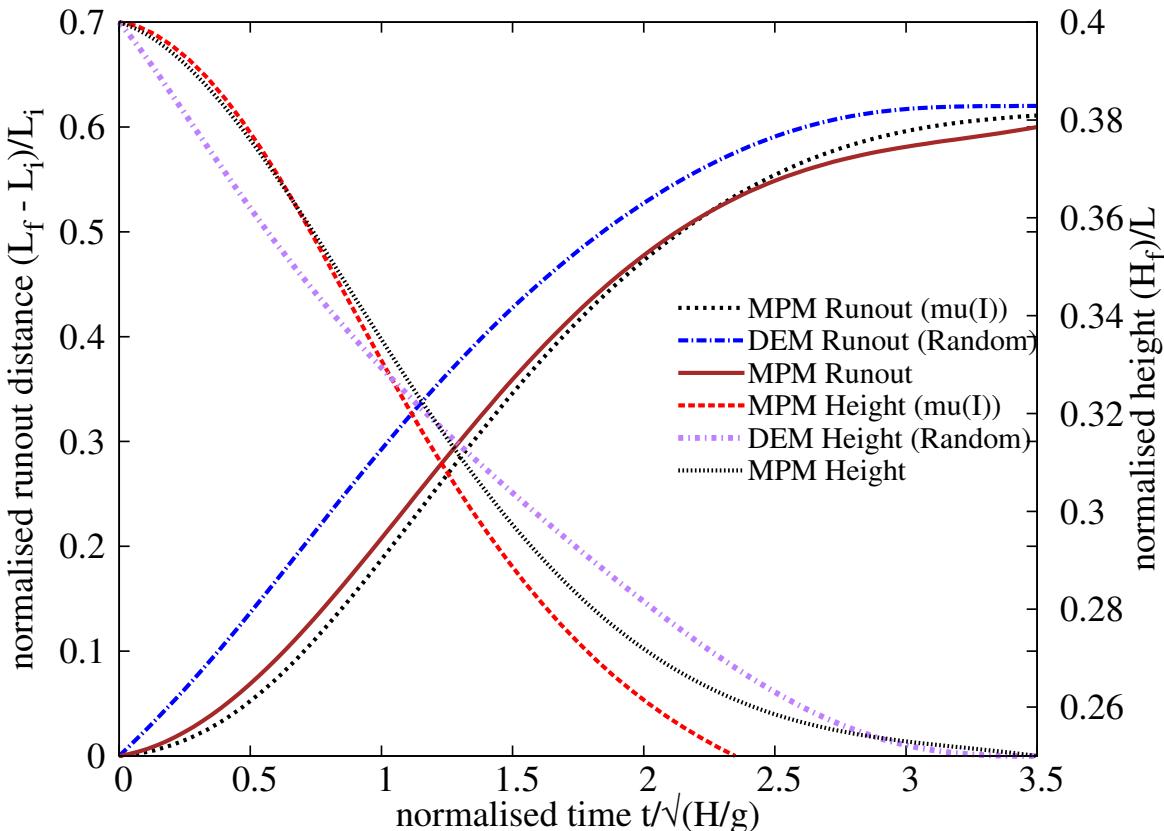


Figure 3.12 Flow evolution of granular column collapse using  $\mu(I)$  rheology

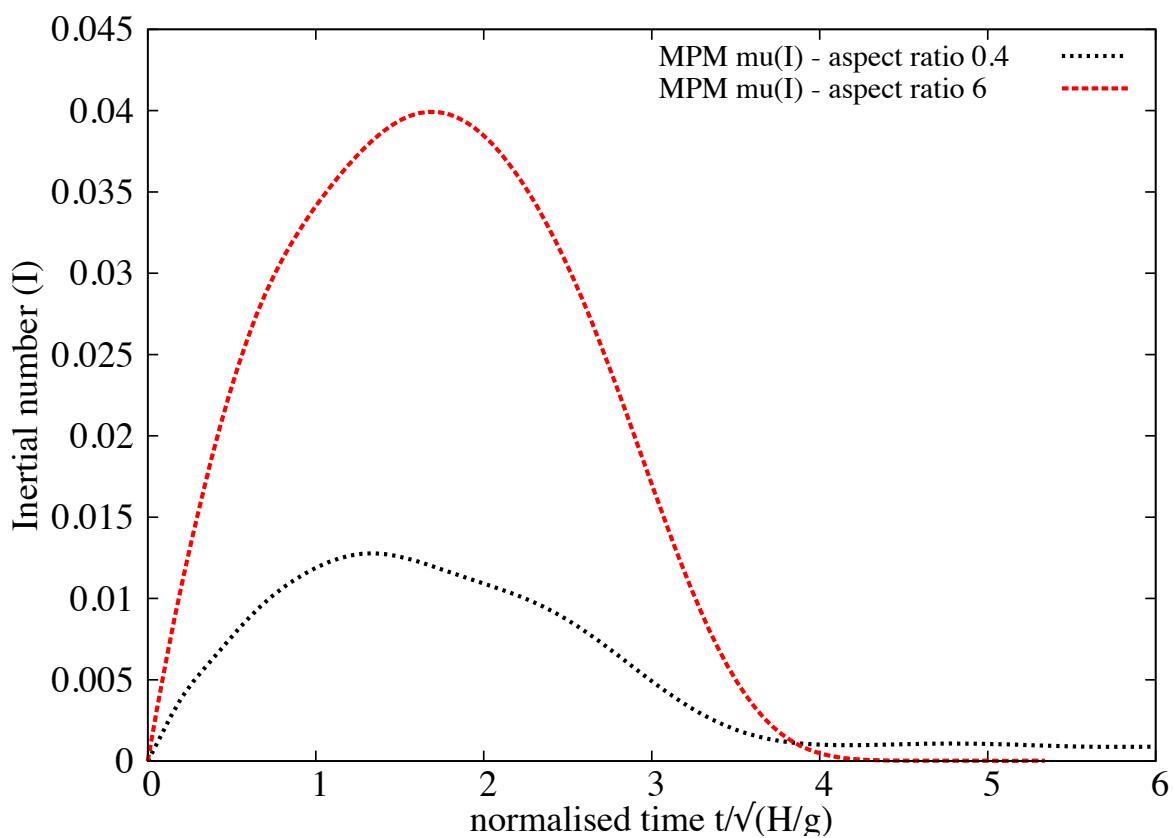


Figure 3.13 Evolution of inertial number with time for columns with ' $a' = 0.4$  and ' $a' = 6$

29 to show nevertheless how the initial condition was dominating the behaviour of the spreading  
 30 through the mass distribution induced in the flow. This means that the knowledge of the final  
 31 run-out is not a sufficient characterization of the deposit: one also needs to know how mass  
 32 is distributed to understand the dynamics and the dissipation process. This is expected to  
 33 be true in natural contexts as well as in experiments. While the inter-grain friction  $\mu$  does  
 34 not affect the early vertical dynamics, nor the power-law dependence, it controls the effective  
 frictional properties of the flow, and its internal structure. It is interesting to note that the details  
 of the structure of the flow do not influence the final run-out dependence, and thus seem to  
 play a marginal role in the overall behaviour of the spreading. This could explain why simple  
 shallow-water model with basic rheology but where the free-fall dynamics was accounted for  
 could reproduce the run-out scalings.

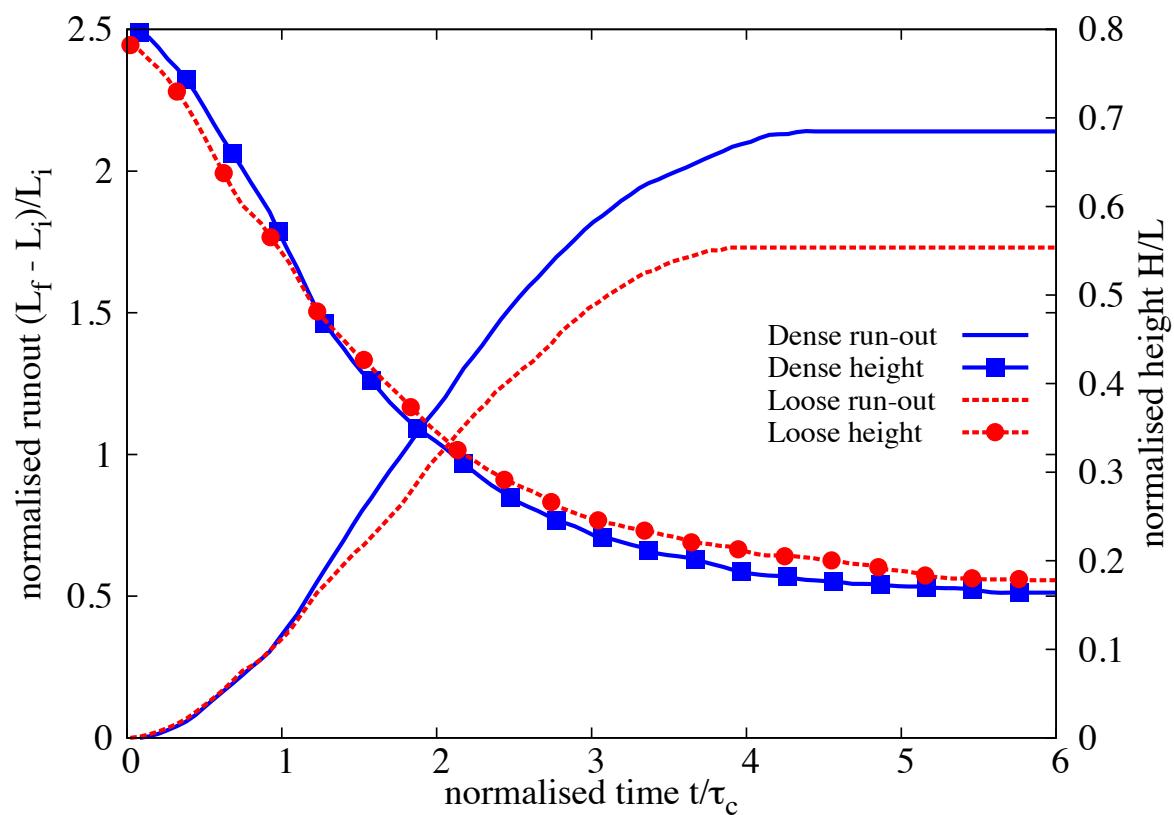
At this stage, it appears that the collapse experiment for large aspect ratios mixes two very  
 different dynamics: while the second stage consists of a “conventional” horizontal granular  
 flows, the first stage implies a large vertical acceleration. It shows how the initial condition can  
 influence the overall behaviour of a granular system, and suggests that triggering mechanisms  
 play a crucial role in the case of natural flows. This stresses the necessity of accounting for  
 vertical acceleration in continuum models in the perspective of producing realistic prediction  
 of the behaviour of granular flows.

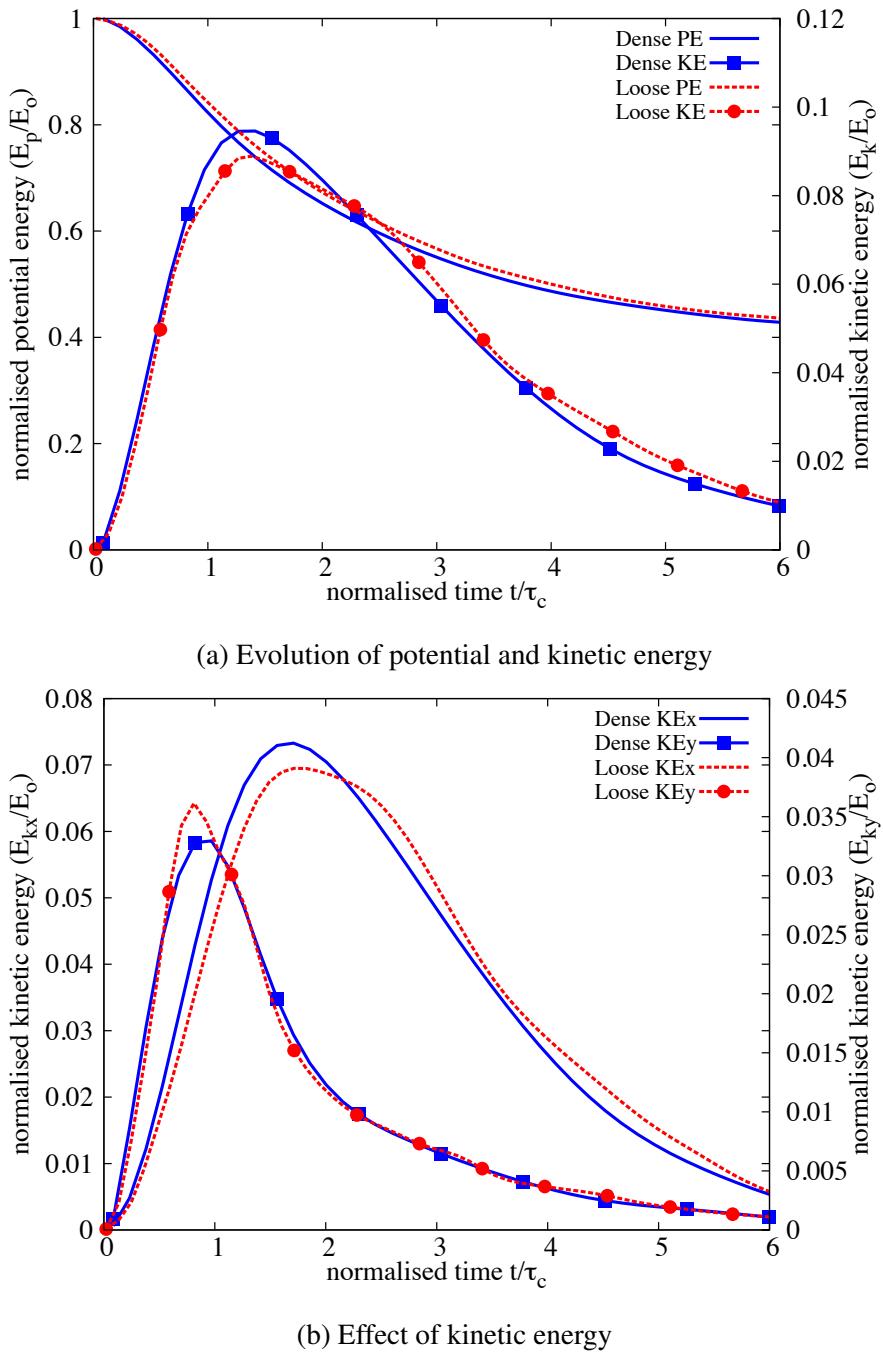
The numerical constants of proportionality, however, show clear material dependence. This  
 corroborates the conclusion of [Balmforth and Kerswell \(2005\)](#); [Lajeunesse et al. \(2004\)](#) and  
 softens that of [Lube et al. \(2005\)](#).

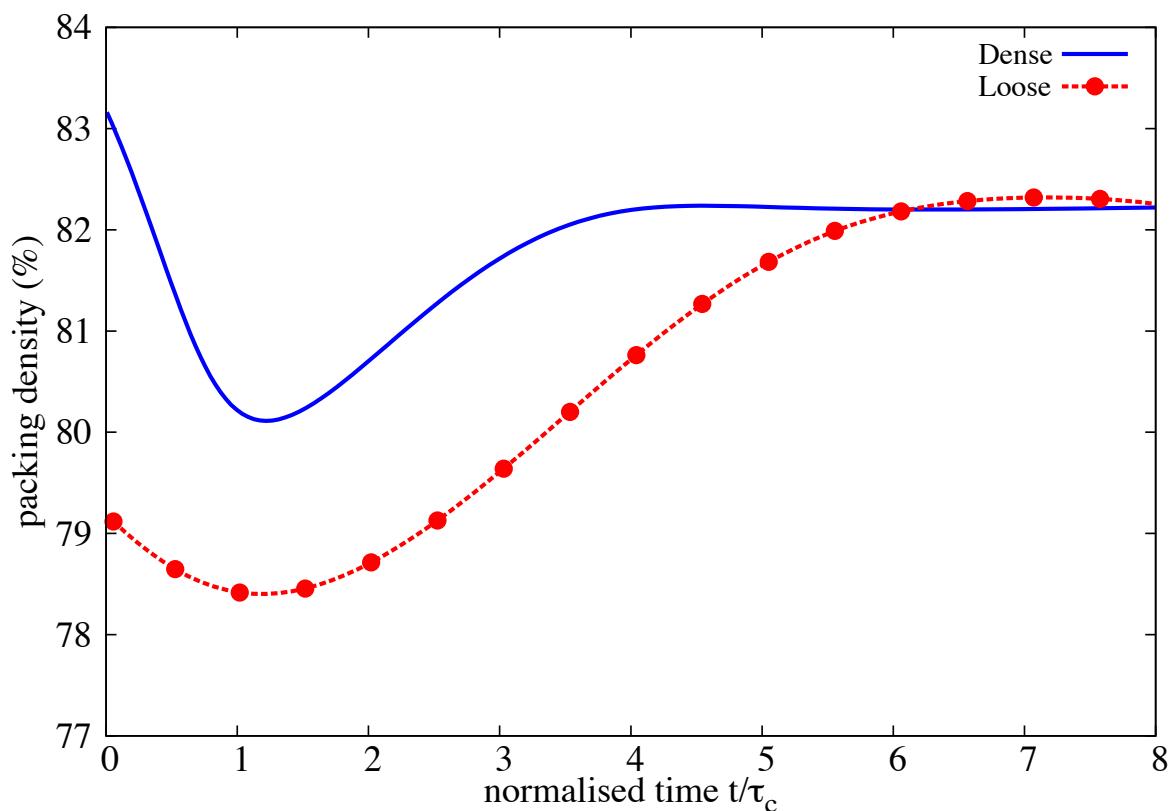
### 3.3 Slopes subjected to impact loading

Transient granular flows occur very often in nature. Well-known examples are rockfalls, debris  
 flows, and aerial and submarine avalanches. They form a major element of landscape reshape  
 and their high destructive potential is a substantial factor of risk. Natural granular flows may be  
 1 triggered as a result of different processes such as gradual degradation, induced by weathering  
 2 or chemical reactions, liquefaction and external forces such as earthquakes.

3 Granular flows have been studied in laboratory experiments in different geometries such as  
 4 tilted piles leading to slope failure and surface avalanches ([Iverson, 1997](#); ?) or by considering  
 5 vertical columns of grains collapsing and spreading under their own weight ([Lajeunesse et al.,](#)  
 6 [2004, 2005](#)). The dynamics observed in the experiments is often nontrivial in the sense that  
 7 the final configurations after the dissipation of the whole kinetic energy can not be readily  
 8 predicted by means of simple laws based on the Mohr-Coulomb nature of the material. For

Figure 3.14 Effect of density on run-out evolution ' $a'$  = 0.8

Figure 3.15 Effect of density on energy evolution  $a = 0.8$

Figure 3.16 Evolution of local packing density ' $a'$  = 0.8

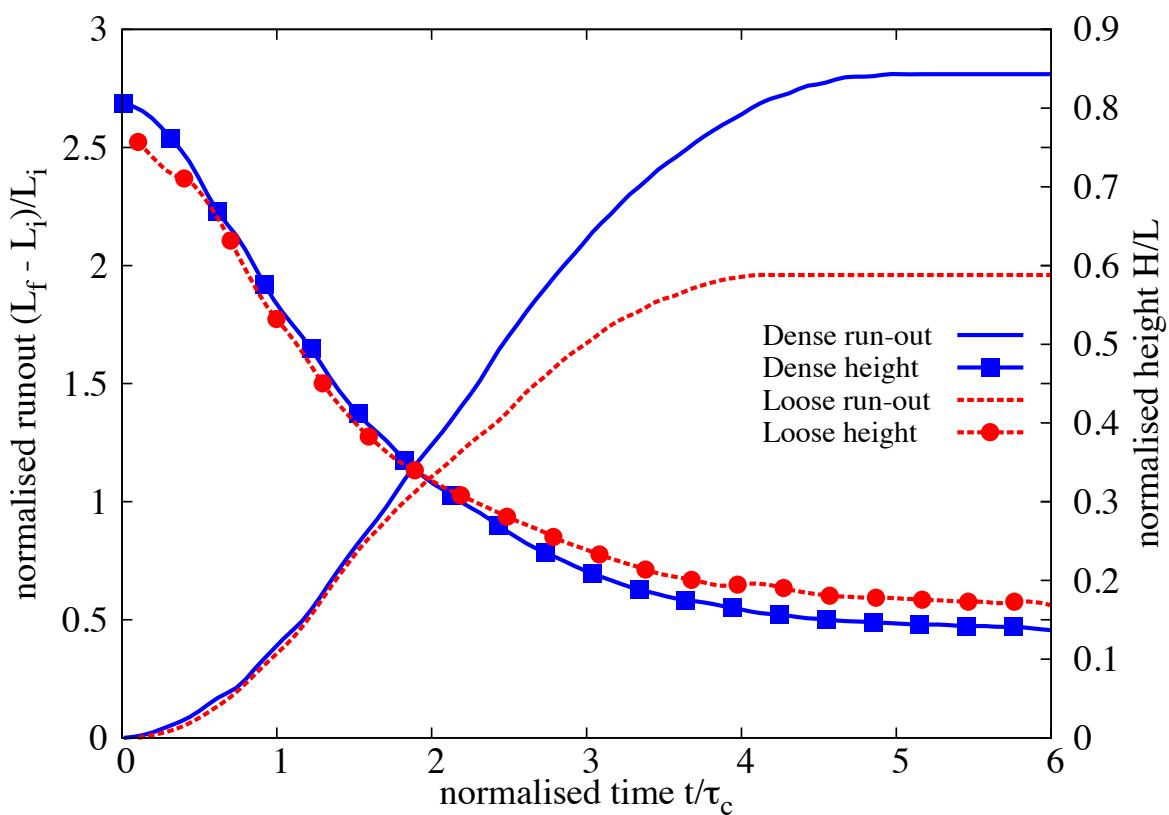


Figure 3.17 Effect of density on run-out evolution ' $a'$  = 0.8 (poly-dispersity ' $r$ ' = 6)

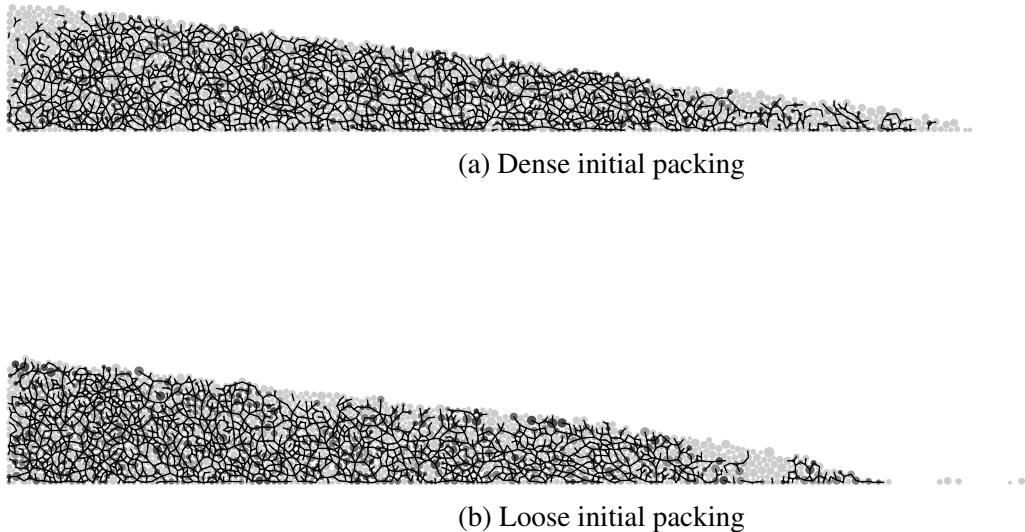
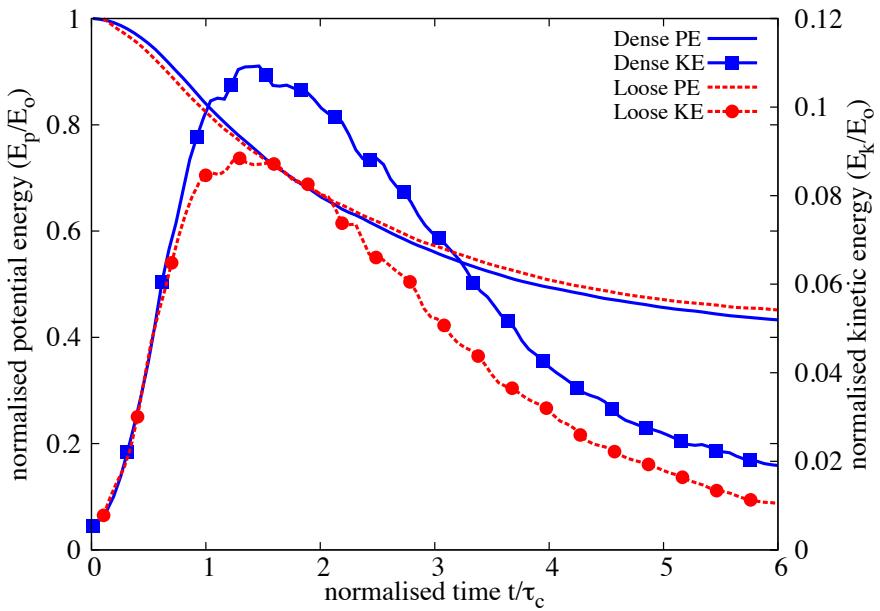


Figure 3.18 Snapshots of granular column collapse  $t = 6\tau_c$

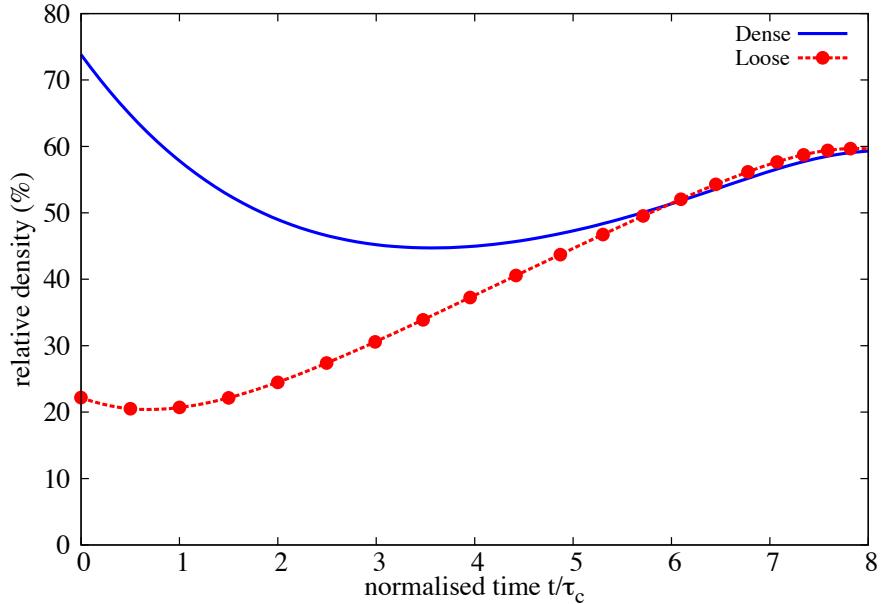
example, in collapsing columns, the run-out distance is found to obey a power-law dependence upon the initial aspect ratio of the column.

The observed nontrivial transient dynamics is often correctly reproduced by the DEM, which provides a powerful tool for the grain-scale analysis of the trigger and its subsequent dynamics (Staron and Lajeunesse, 2009; Staron et al., 2005). However, even in simplified geometries such as those investigated in the experiments, the DEM suffers from a serious short-coming in the number of particles that can be simulated in a reasonable time. This is a critical issue when more complex geometries or long-time granular processes are considered, or when particle size distributions are broad. For this reason, most numerical studies are performed in 2D or simple particle shapes and size distributions are considered.

It is also obvious that classical modelling strategies based on the finite element method (FEM) cannot be used for the simulation of very large deformations. In various applications of FEM, this problem is treated by means of technical tools such as re-meshing. Such methods are, however, not robust and lead to round-off errors and mesh-sensitivity. In contrast, the so-called Material Point Method (MPM) is an alternative approach for continuum problems that allows for indefinitely large deformations without re-meshing <sup>1</sup>. In this method, the material points carry the information on state variables and a background fixed grid is used to solve the governing equations. The information between the material points and the grid is exchanged via suitable shape functions. The MPM has been applied with success to a number of solid



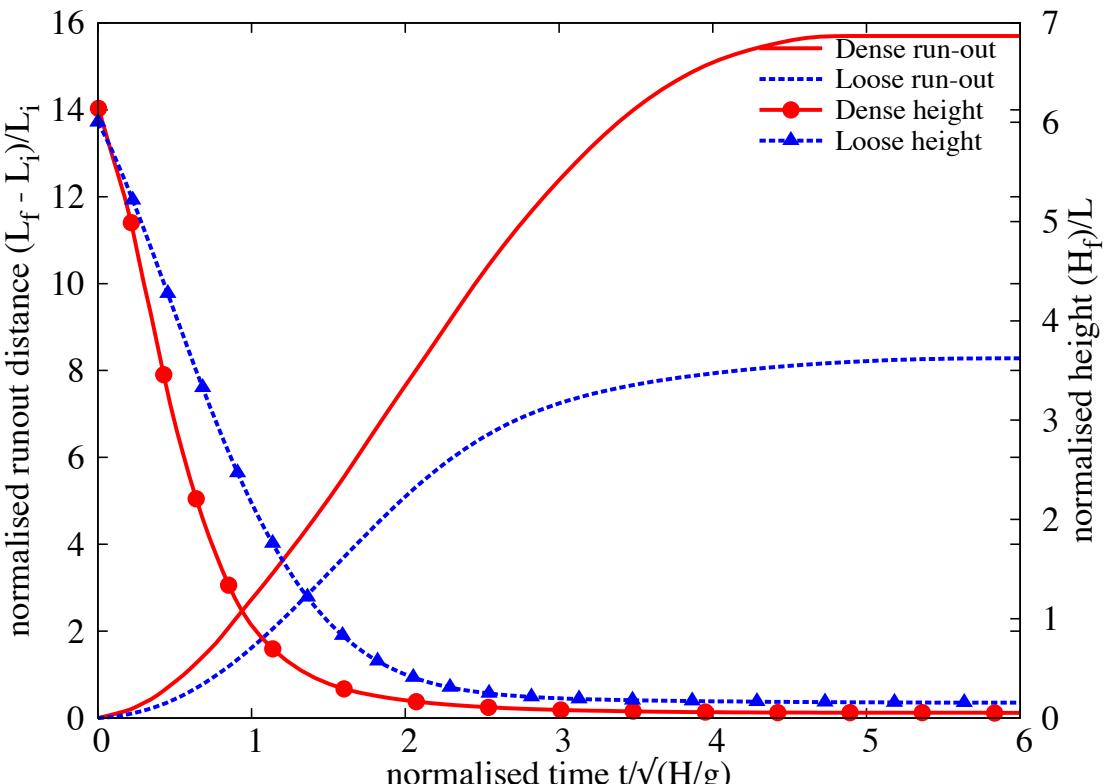
(a) Evolution of potential and kinetic energy



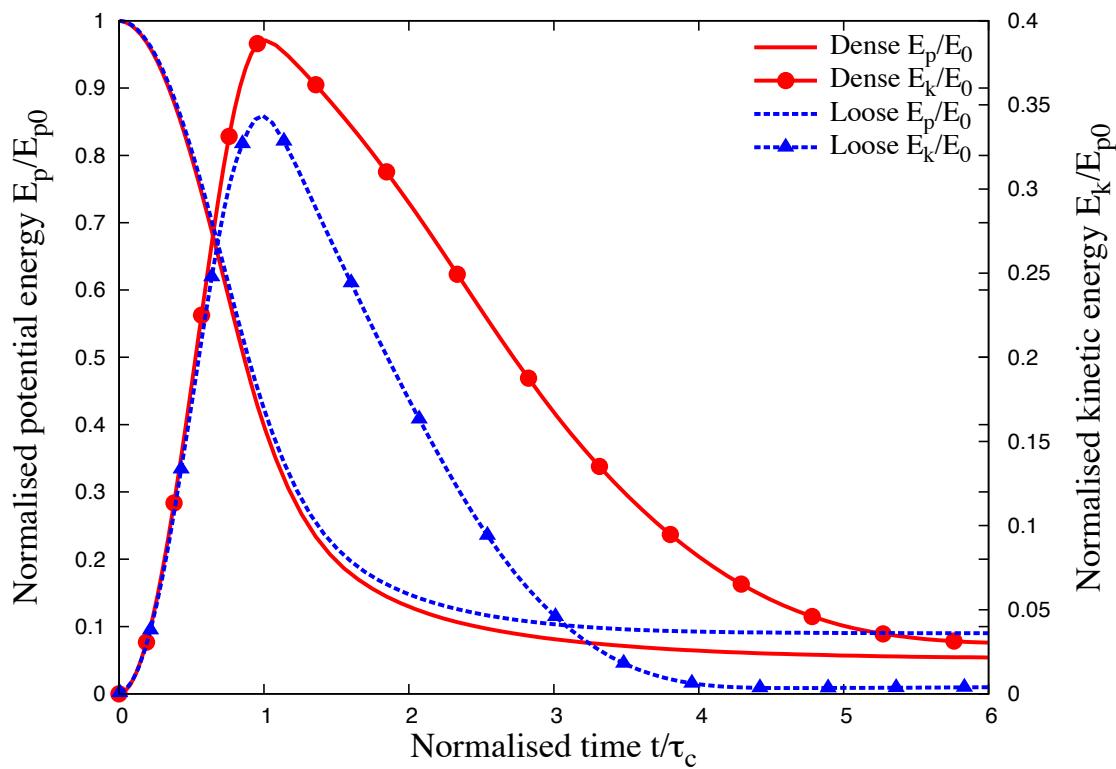
(b) Evolution of packing density

Figure 3.19 Effect of density on energy and packing fraction evolution ' $a'$  = 0.8 (poly-dispersity ' $r$ ' = 6)

## 3.3 Slopes subjected to impact loading



(a) Effect of density on run-out evolution



(b) Effect of density on energy evolution

Figure 3.20 Effect of density on run-out behaviour and energy evolution ' $a'$  = 0.6

mechanics problems and its theoretical foundations have recently been investigated by several authors.

In this paper, we are concerned with the ability of the MPM, as a continuum approach, to reproduce the evolution of a granular pile under its own weight or when destabilized by energy input. In particular, a central issue is whether power-law dependence of the run-out distance and time with respect to the initial geometry or energy can be reproduced by a simple prescription of the Mohr-Coulomb plastic behaviour within a MPM code. We therefore perform extensive simulations by varying continuously different input parameters and compare the data with those obtained from DEM simulations of the same system. We compare in detail the evolution of the profile of the pile and its total kinetic energy between the two methods and for different initial states. As we shall see, the MPM can successfully simulate the transient evolution with a single input parameter, namely the internal angle of friction. This opens the way to the simulation of geological-scale flows on complex topographies.

### 3.3.1 Numerical procedures

The numerical samples are composed of  $\sim 13000$  disks with a uniform distribution of diameters by volume fractions in the range  $[d_{min}, d_{max}]$  with  $d_{max} = 1.5d_{min}$ . The mean particle diameter and mass are  $d \simeq 0.0025$  m and  $m \simeq 0.0123$  kg, respectively. The particles are first poured uniformly into a rectangular box of given width and then the right-hand side wall is shifted further to the right to allow the particles to spread. A half-pile is obtained when all particles come to rest; see ???. This procedure leads to a mean packing fraction  $\simeq 0.83$ .

Table 3.4 DEM simulation of simple shear test (?)

Parameter	Value
Mean grain diameter	$\approx 2.455$ mm
Grain diameter $[d_{min} : d_{max}]$	[2.0, 3.0] mm
Friction coefficient	0.4
Grain density	$2600\text{kg m}^{-3}$
Restitution coefficient	0
Number of grains	1174

The initial static pile is set into motion by applying a constant horizontal gradient  $v_{0x}(y) = k(y_{max} - y)$  with  $k > 0$ . Such a configuration mimics the energy transfer mechanism of a horizontal quake along the bottom of the pile. We are interested in the evolution of the geometry of the pile and its total kinetic energy as a function of the initial input energy  $E_0$ . The run-out distance  $R_f$  is the distance of the rightmost particles from the left wall when the pile

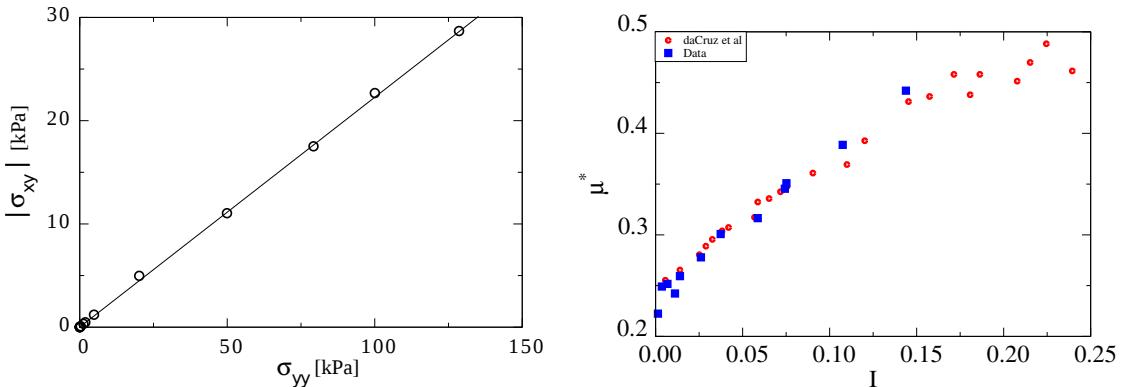
comes to rest. It will be normalized by the initial extension  $R_0$  of the pile, as in the experiments of collapsing columns. The total run-out duration  $t_f$  is the time that the pile takes to reach its final run-out distance  $R_f$ .

The initial static pile is set into motion by applying a horizontal velocity  $v_{0x}(y)$  to all particles during a short interval of time. Different velocity fields were tested: 1) The same velocity  $v_{0x}(y) = v_0$  applied to all particles, 2) The same velocity  $v_{0x}(y) = v_0$  applied to a column of particles next to the left wall, 3) a constant velocity gradient  $v_{0x}(y) = k(y_{max} - y)$  with  $k > 0$ . The first two pushing modes mimic the case of a pile impacted from the left

1 by a moving mass (tsunami, debris...) whereas the last mode represents energy transfer by  
 2 horizontal quake of the bottom line. We will compare briefly below the effect of different  
 3 pushing modes, but later we will mainly explore the third mode. We are interested in the  
 4 evolution of the geometry of the pile and its total kinetic energy as a function of the initial  
 5 energy input  $E_0$ . The run-out distance  $R_f$  is the distance of the rightmost particles from the left  
 6 wall when the pile comes to rest. It will be normalized by the initial extension  $R_0$  of the pile, as  
 7 in the experiments of collapsing columns. The total run-out duration  $t_f$  is the time that the pile  
 8 takes to reach its final run-out distance  $R_f$ .

9 For grain scale simulations, classical DEM and Contact Dynamics approach is used. A  
 10 detailed description of the Contact Dynamics method can be found in [Jean \(1999\)](#); [Radjai and Dubois \(2011\)](#); [Radjai and Richefeu \(2009\)](#); ?. This method is based on implicit time  
 12 integration of the equations of motion and a nonsmooth formulation of mutual exclusion and  
 13 dry friction between particles. The CD method requires no elastic repulsive potential and no  
 14 smoothing of the Coulomb friction law for the determination of forces. For this reason, the  
 15 simulations can be performed with large time steps compared to discrete element simulations.  
 16 The unknown variables are particle velocities and contact forces, which are calculated at each  
 17 time step by taking into account the conservation of momenta and the constraints due to mutual  
 18 exclusion between particles and the Coulomb friction. We use an iterative research algorithm  
 19 based on a nonlinear Gauss-Seidel scheme. The only contact parameters within the CD method  
 20 are the friction coefficient  $\mu_s$ , the normal restitution coefficient  $e_n$  and the tangential restitution  
 21 coefficient  $e_t$  between particles. We will investigate the effect of these parameters on the  
 22 evolution of kinetic energy and the profile of the pile.

23 The natural units of our system are the mean particle diameter  $d$ , mean particle mass  $m$   
 24 and gravity  $g$ . For this reason, in the following we normalize the lengths by  $d$ , the times by  
 $(d/g)^{1/2}$ , the velocities by  $(gd)^{1/2}$  and the energies by  $mgd$ .



(a) Evaluating the critical state friction angle from periodic shear test.  
(b) Evolution of Inertial number with friction  $\mu$

Figure 3.21 Periodic shear test using CD (?).

### 3.3.2 Evolution of pile geometry and run-out

In this section, we consider the spreading process following the initial energy input into the pile. Fig. ?? shows several snapshots of the pile for an initial input energy  $E_0 = 61$  (in dimensionless units). The pile is sheared from the bottom to the top, thus leaving a cavity in the vicinity of the left wall. The cavity is partially filled while the pile continues to spread to the right.

In this section, we consider the spreading process following the initial energy input into the pile. Fig. ?? shows several snapshots of the pile for each pushing mode and for the same initial energy  $E_0 = 61$  (in dimensionless units). In mode 1, where the same velocity is imparted to all particles, the whole pile moves away from the left wall over a short distance and then it spreads out and declines in slope. The spreading continues farther until the slope nearly declines to zero. In mode 2, where the velocity is applied to a column of particles next to the left wall, the particles belonging to the column are literally expelled from the pile. They fall back farther way on the pile after a ballistic travel above the pile. At the same time, the right side of the pile slightly spreads away while the left side is filled by the particles rolling down into the gap left by the column. In mode 3, the pile is sheared from the bottom to the top, leaving thus a cavity in the vicinity of the left wall. The cavity is partially filled while the pile continues to spread.

All pushing modes involve a transient with a sharp change of the geometry of the pile followed by continuous spreading to the right. In mode 2, most of the energy is carried away by the ejected particles. In mode 1, the pile has a rigid-body velocity component and moves away from the left wall, but shows an efficient energy transfer leading to a long run-out distance. The transient is more energy consuming in mode 3 compared to mode 1. For this reason, the run-out distance in mode 3 is long but shorter than in mode 1. In the following, we analyze in more detail the evolution of the pile in mode 3, which mimics a horizontal quake from the bottom

and, despite the creation of a cavity, remains always in contact with the left wall irrespective of the input energy.

Figure 3.4 shows the normalized run-out distance  $(R_f - R_0)/R_0$  and total run-out time  $t_f$  as a function of the input energy  $E_0$ . We observe two regimes both characterized by a power-law run-out distance and time as a function of  $E_0$ . In the first regime, corresponding to the range of low input energies  $E_0 < 40 \text{ mgd}$ , the run-out distance varies as  $R_f \propto (E_0)^\alpha$  with  $\alpha \simeq 0.61 \pm 0.04$  over nearly one decade while the duration keeps a constant value  $t_f \simeq 60 (d/g)^{0.5}$  irrespective of the value of  $E_0$ ! The error on the value of the exponent represents the confidence interval of linear fits on the logarithmic scale. An average run-out speed can be defined from the ratio  $v_s = (R_f - R_0)/t_f$ . According to the data, we have  $v_s \propto (E_0)^{0.61 \pm 0.04}$ . Since the initial average velocity varies as  $v_0 \propto (E_0)^{0.5}$ , this difference between the values of the exponents suggests that the mobilized mass during run-out declines when the input energy is increased. As we shall see below, the constant run-out time reflects also the collapse of the particles into the cavity left behind the pile.

In the second regime, corresponding to the range of high input energies  $E_0 > 40 \text{ mgd}$ , the run-out distance varies as  $R_f \propto (E_0)^{\alpha'}$  over one decade with  $\alpha' \simeq 0.77 \pm 0.03$  while the duration increases as  $t_f \propto (E_0)^{\beta'}$  with  $\beta' \simeq 0.21 \pm 0.04$ . Hence, in this regime the average run-out speed varies as  $v_s \propto (E_0)^{0.56 \pm 0.07}$ . This exponent is close to the value 0.5 in  $v_0 \propto (E_0)^{0.5}$ , and hence, within the confidence interval of the exponents, in the second regime we may assume  $\beta' \simeq \alpha' - 0.5$  and  $v_s \propto v_0$ .

It is worth noting that a similar power-law dependence of the run-out distance and time were found in the case of collapsing columns of grains with respect to the initial aspect ratio Topin et al. (2012). In the column geometry, the particles spread away owing to the kinetic energy acquired during gravitational collapse of the column. Topin et al. found that the run-out distance varies as a power law of the available peak kinetic energy at the end of the free-fall stage with an exponent  $\simeq 0.5$ . This value is below those obtained here for both regimes. This is, however, physically plausible since the distribution of particle kinetic energies at the end of the collapse is more chaotic than in our simulations where the energy is supplied from the very beginning in a well-defined shear mode. As pointed out by Staron et al. (2005), the distribution of kinetic energies is an essential factor for the run-out distance.

### 3.3.3 Decay of kinetic energy

The non-trivial evolution of the pile geometry in two regimes suggests that the energy supplied to the pile is not simply dissipated by shear and friction with the bottom plane. We also need to split the kinetic energy into its different components ( $x$ ,  $y$  and rotation) of the velocity field. The input energy is in the  $x$  component, but due to both the creation of a cavity next to the left

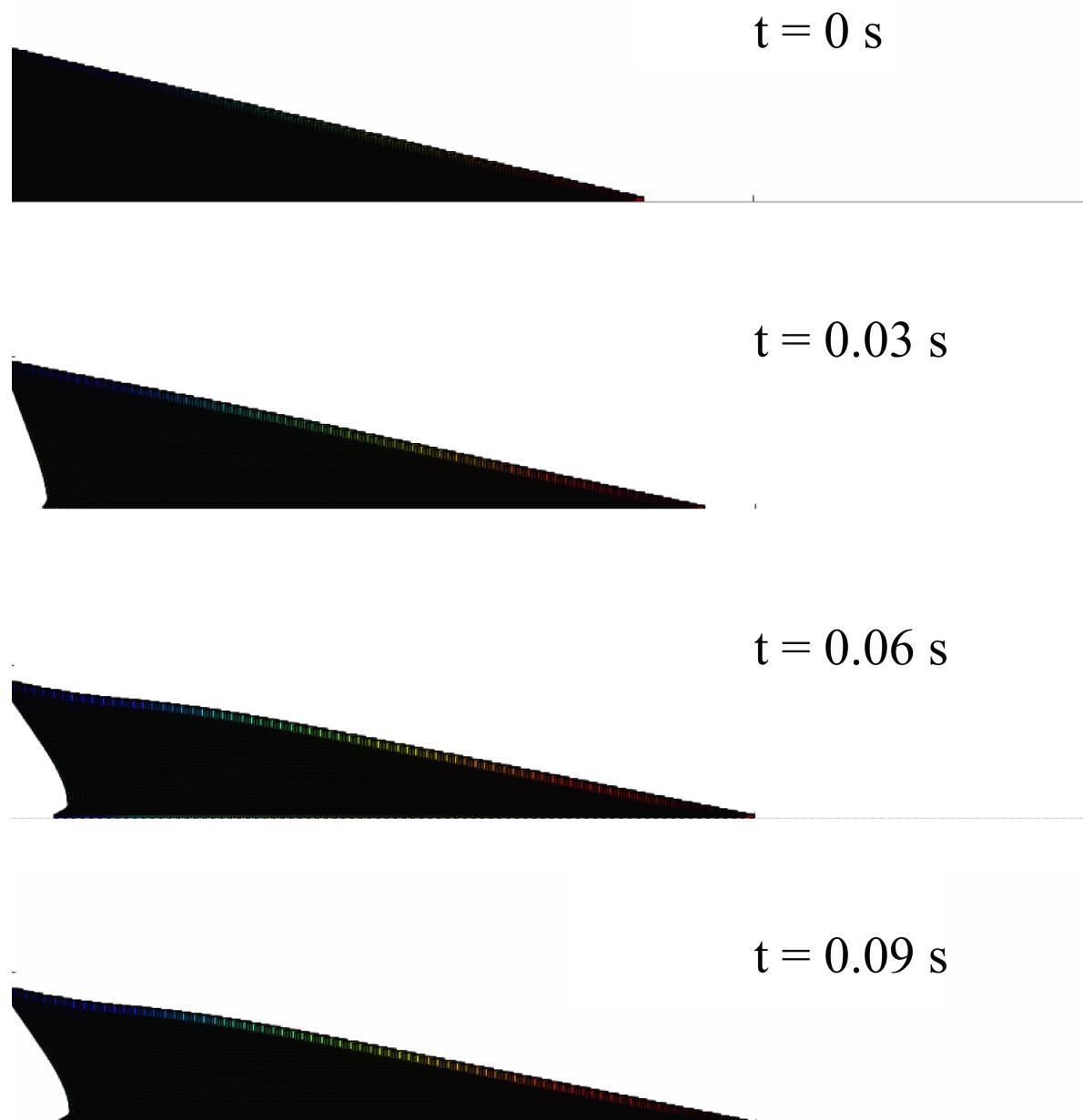


Figure 3.22 Snapshots of MPM simulations of the evolution of granular pile subjected to a gradient impact energy.

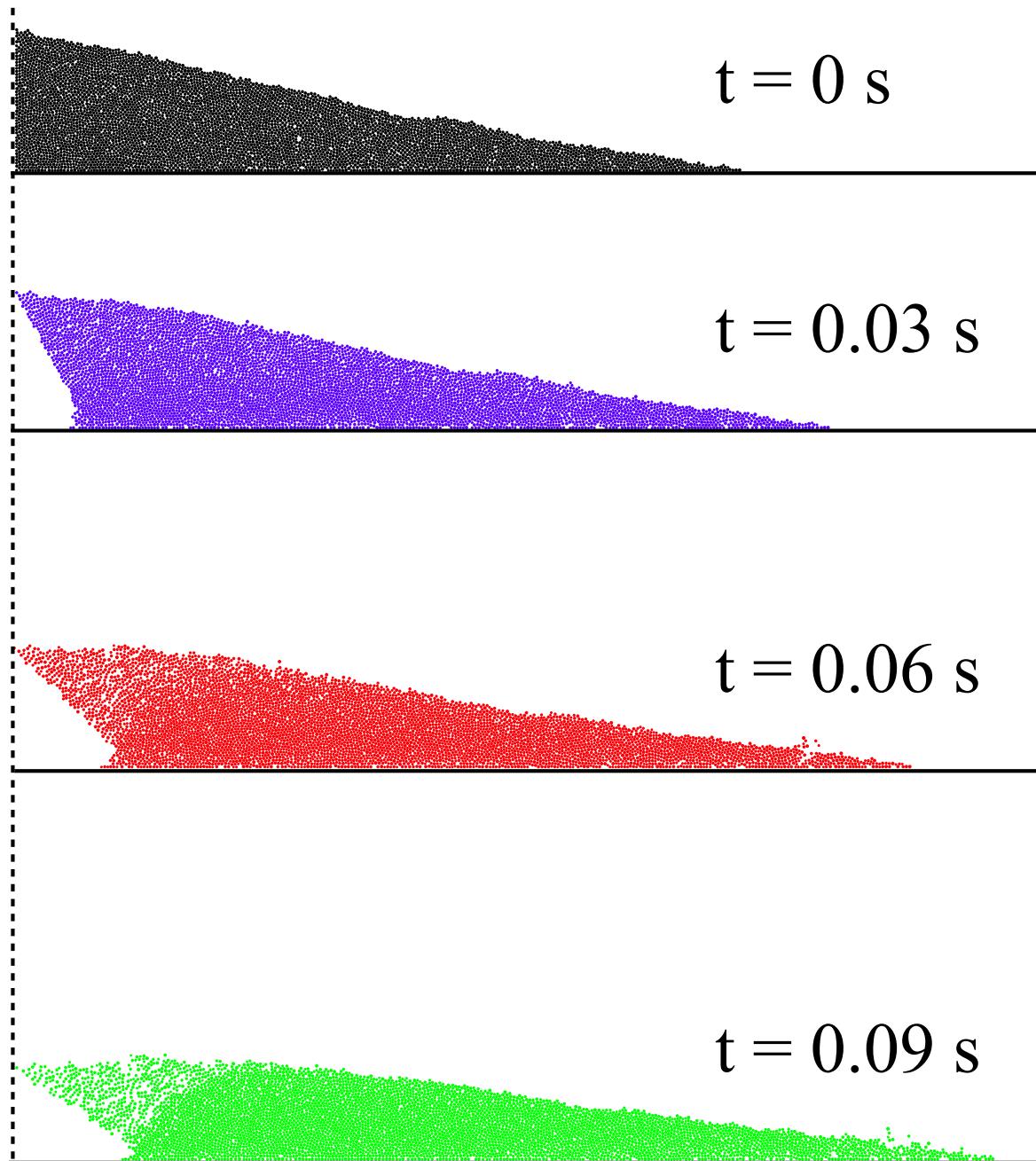


Figure 3.23 Snapshots of CD simulations of the evolution of granular pile subjected to a gradient impact energy (?).

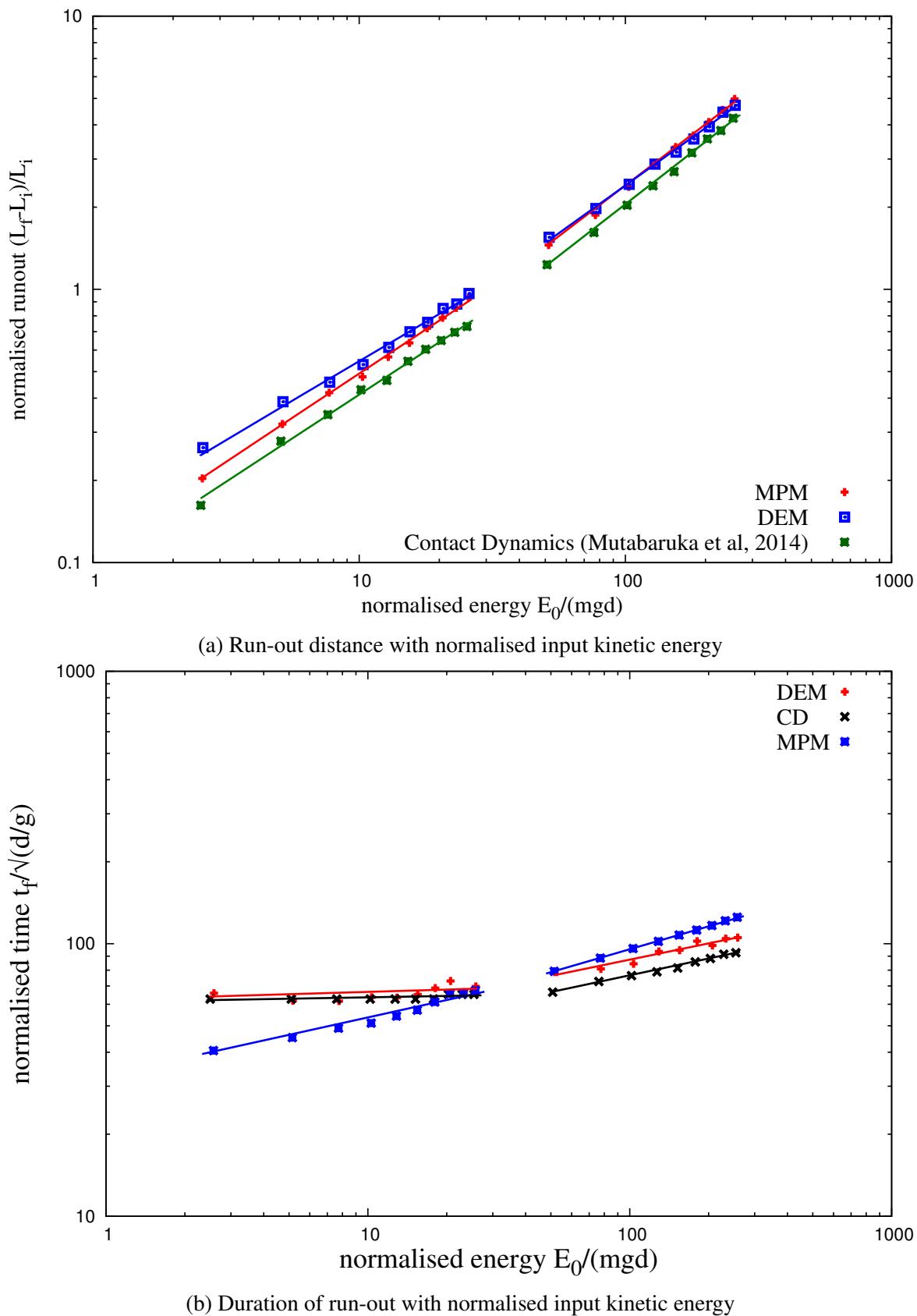


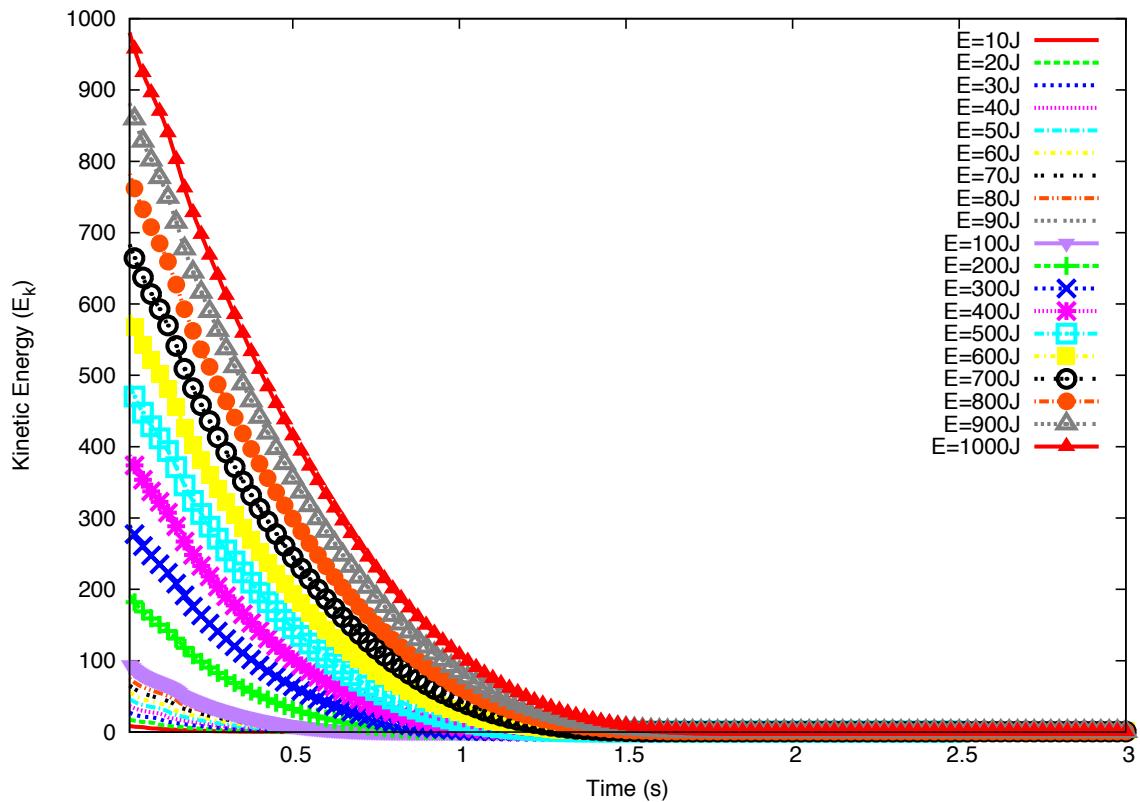
Figure 3.24 Run-out behaviour of a pile subjected a gradient impact energy

wall and the rolling of the particles down the free surface of the pile and between particles, a fraction of the energy is first transferred to the  $y$  component of the velocity field and dissipated during the transient. In this section, we analyse these features in order to arrive in a picture consistent with the evolution of the pile shape.

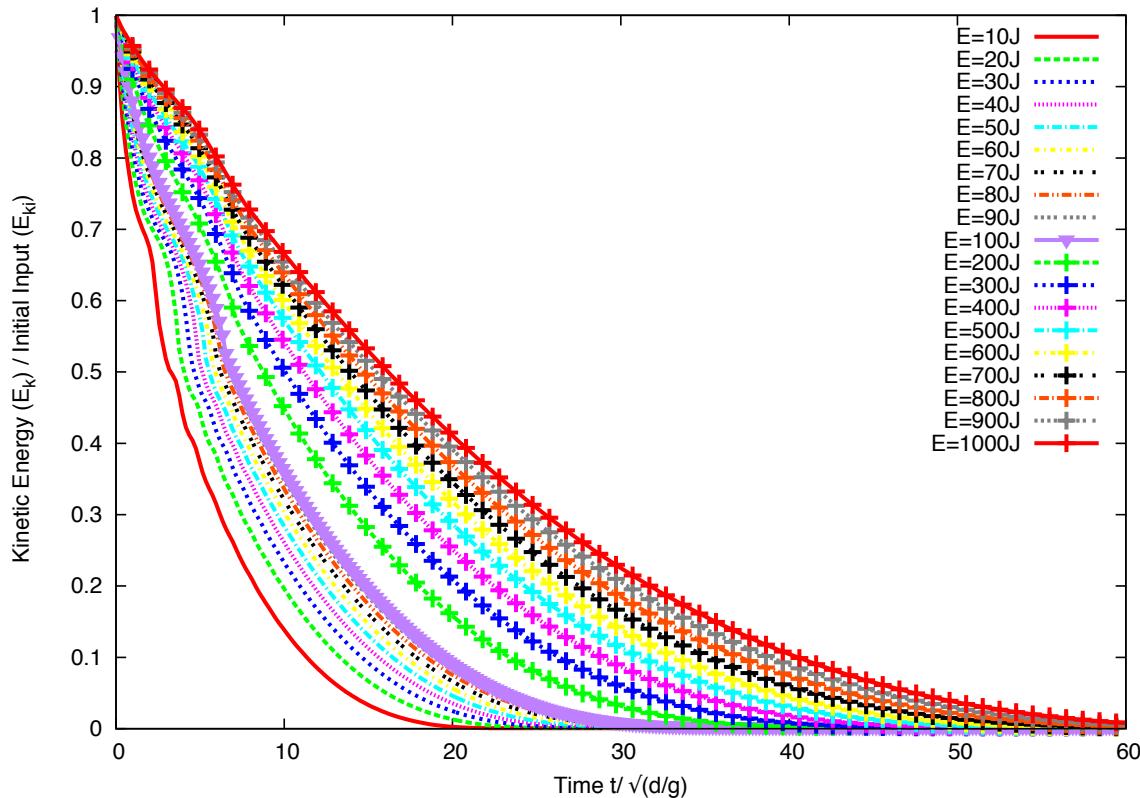
The decay of the total kinetic energy  $E$  is displayed in ??(a) for values of the input energy  $E_0$ . We observe an initial fast drop of  $E$  followed by a regular fall-off until the end of the run-out. This regular fall-off occurs clearly with two different functional forms, thus revealing two stages in the evolution of the pile. ??(b) shows the same plots normalized by  $E_0$ . We see that all plots corresponding to the first regime (low energies) collapse nearly on to a single time evolution. This is consistent with the fact that, as previously shown, in this regime the run-out time  $t_f$  is independent of the input energy. In contrast, the plots corresponding to the second regime (high energies) collapse only at the beginning of run-out, i.e. for  $t < t_1 \simeq 7.5 (d/g)^{0.5}$ .

?? displays the evolution of kinetic energy in the translational ( $E_x$  and  $E_y$ ) and rotational ( $E_\theta$ ) degrees of freedom of the particles.  $E_x$  decays as the total energy, but  $E_y$  and  $E_\theta$  increase and pass through a peak before decaying rapidly to a negligibly small level. The transient is best observed for  $E_y$ , which has significant values only for  $t < t_1$ . This energy represents the proportion of kinetic energy transferred to the  $y$  component of the velocity field due to the destabilization of the pile and collapse of particles in the cavity behind the pile. We note that the lower  $E_0$ , the higher the peak value of  $E_y/E_0$ . This means that, at low values of the input energy a larger fraction of input energy  $E_0$  is consumed in the destabilization process whereas at a high level of input energy, most of it is dissipated in the spreading phase. For this reason, the total duration  $t_1$  of this destabilization transient is nearly the same in both regimes and its value is controlled by the gravity rather than the input energy. The height of the pile being of the order of  $80 d$ , the total free-fall time for a particle located at this height is  $\simeq 12 (d/g)^{0.5}$ , which is of the same order as  $t_1$ . As to the rotational energy, its contribution both to the transient stage and spreading appears to be negligible.

To analyze the second phase in the second regime, we now consider only the kinetic energy  $E'_{x0}$  available at the end of the transient. This energy is responsible for most of the run-out and hence it is expected to control the run-out distance and time. Fig. ??(a) shows the evolution of  $E_x$  normalized by  $E'_{x0}$  as a function of time. The plots have seemingly the same aspect but they show different decay times. A decay time  $\tau$  can be defined as the time required for  $E_x$  to decline by a factor  $1/2$ . Fig. ??(b) shows the same data in which the time  $t'$  elapsed since  $t_1$  is normalized by  $\tau$ . Interestingly, now all the data nicely collapse on the same curve. We checked that this curve can not be fitted by simple functional forms such as variants of exponential decay. This means that the spreading of the pile is not a self-similar process in agreement with the fact that the energy fades away in a finite time  $t'_f$ .



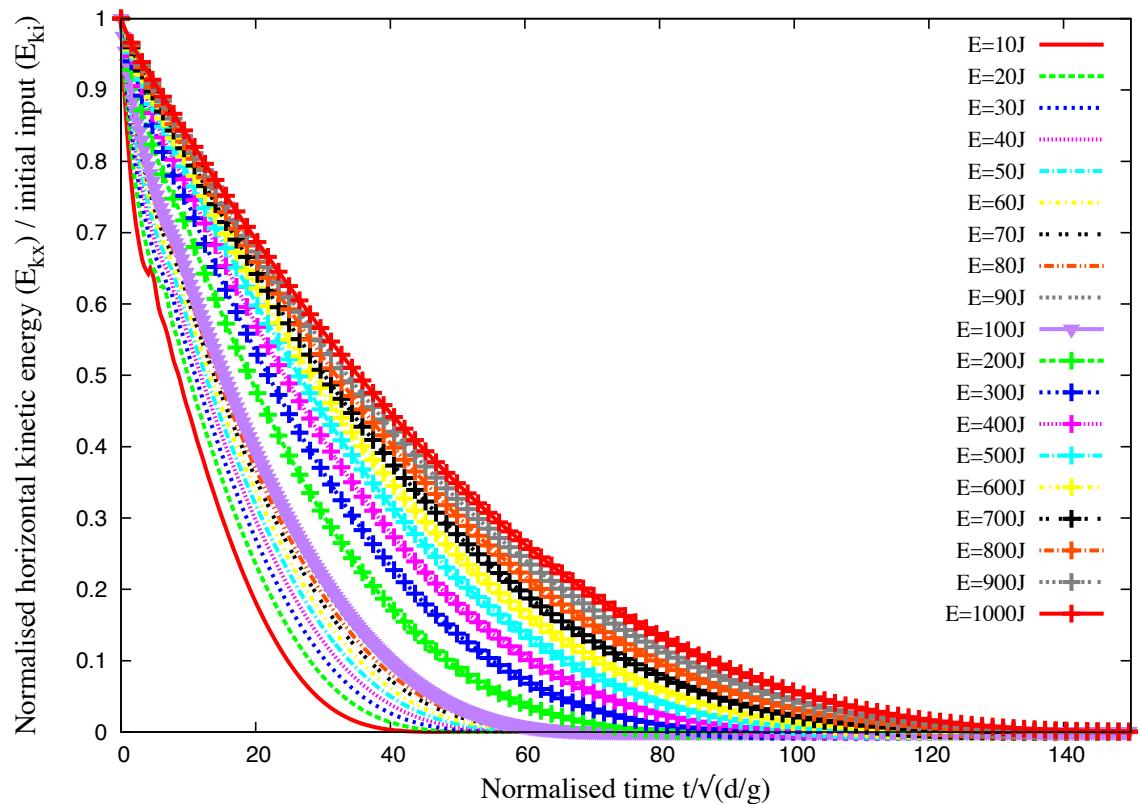
(a) Evolution of total kinetic energy with time



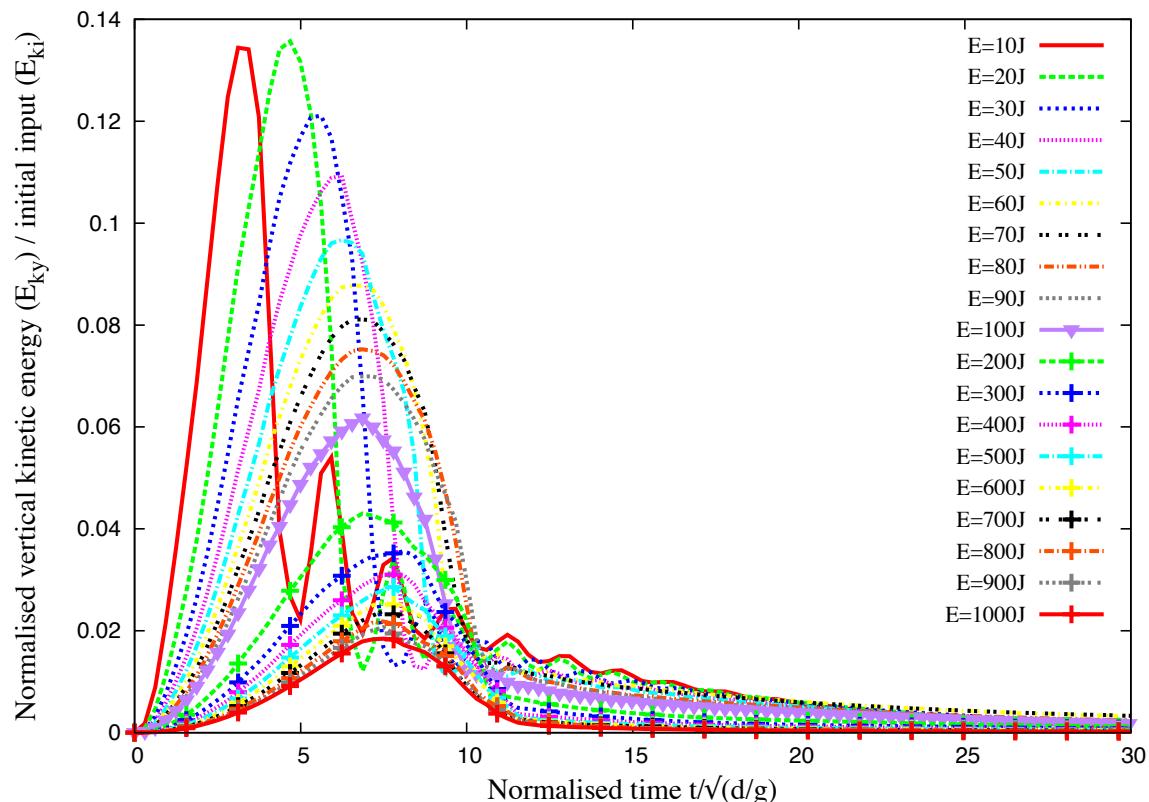
(b) Evolution of normalised kinetic energy with normalised time

Figure 3.25 Evolution of kinetic energy with time

## 3.3 Slopes subjected to impact loading



(a) Evolution of normalised horizontal kinetic energy with time



(b) Evolution of normalised vertical kinetic energy with time

Figure 3.26 Evolution of vertical and horizontal kinetic energy with time

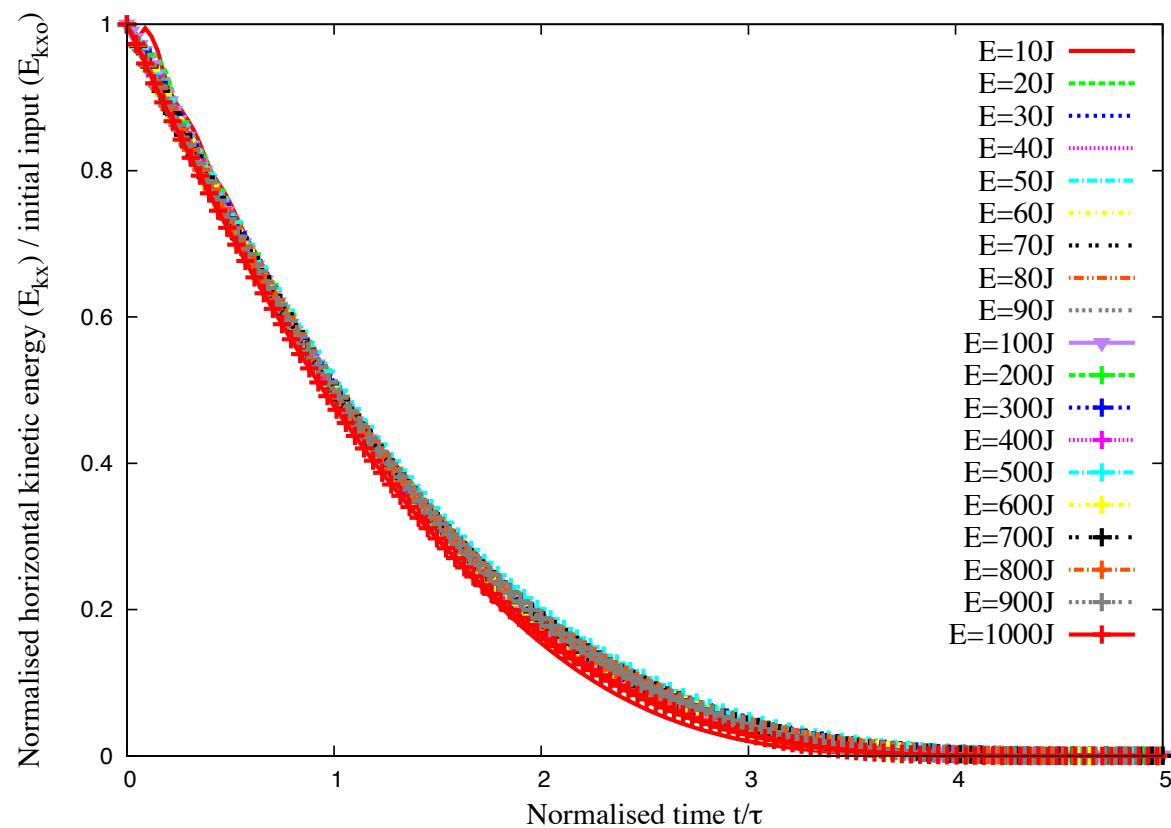


Figure 3.27 Evolution of kinetic energy in the  $x$  component of the velocity field normalized by the available kinetic energy at the end of the transient as a function of normalized time.

The scaling of the data with the decay time  $\tau$  suggests also that the run-out time  $t'_f$  since the beginning of the second phase might be a simple function of  $\tau$ . Figure 3.27 shows both  $t'_f$  and  $\tau$  as a function of  $E'_{x0}$ , where we observe a power law for both times over nearly one decade. The run-out time  $t'_f \propto (E'_{x0})^{\beta'}$  has the same exponent  $\beta' \simeq 0.21 \pm 0.03$  as  $t_f$  as a function of  $E_0$  (see Fig. 3.4). For the decay time we have  $\tau \propto (E'_{x0})^{\beta''}$  with  $\beta'' \simeq 0.28 \pm 0.03$ . The relation between the two times can thus be expressed as

$$t'_f = k \tau (E'_{x0})^{\beta'' - \beta'}, \quad (3.11)$$

where  $k \simeq 5 \pm 0.4$  and  $\beta'' - \beta' \simeq -0.05 \pm 0.06$ . This value is small enough to be neglected within the confidence interval of our data. It is therefore plausible to assume that the run-out

time is a multiple of the decay time and the spreading process is controlled by a single time.

We however note that a weak dependence on the energy  $E'_{x0}$  is consistent with the fact that the whole available energy at the beginning of the second phase is not dissipated in the spreading process (calculated from the position of the tip of the pile) since the pile keeps deforming by the movements of the particles at the free surface even when the tip comes to rest. This can explain the small difference between the two exponents as observed here.

### 3.3.4 Effect of friction

The run-out distance and time and the dissipation of kinetic energy are controlled by the input energy and collective dynamics of the whole pile, as it was analyzed in the previous sections.

But they are expected to depend also on the friction. We performed a series of simulations with different values of base friction. The results are shown in Fig. ?? for the profiles of the pile and evolution of the kinetic energy in time. We see no difference in the results for different values of  $e_n = e_l$ . This is a consequence of the fact that, even at large input energies, the pile remains in a dense state so that multiple collisions inside the pile occur at small time scales compared to the deformation time. When the restitution coefficients are increased, more collisions occur during a longer time interval but the overall energy dissipation rate by collisions remains the same. This effect is a seminal example of collective effects which erase the influence of local parameters at the macroscopic scale. In contrast with the restitution coefficients, however, the effect of the friction coefficient is quite important for the run-out, as observed in Fig. ?? for both the energy decay and geometrical profile of the pile. Both the run-out distance and decay time decrease as the friction coefficient is increased. This effect is much more pronounced at low values of the friction coefficient. The run-out time, for example, is reduced by a factor 4 as  $\mu_s$  is increased from 0.1 to 0.4 while the run-out times and profiles do not change much for  $\mu_s = 0.7$ . This “saturation effect” was evidenced in a systematic way in simple shear tests and

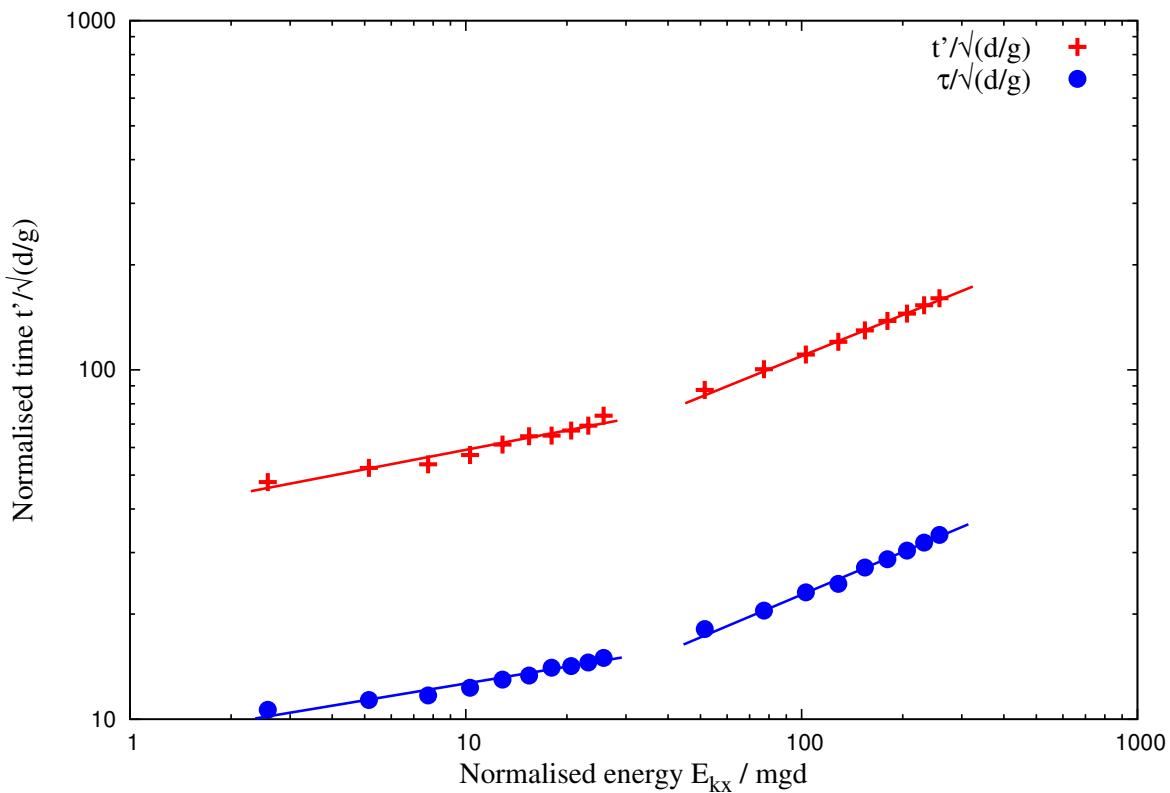
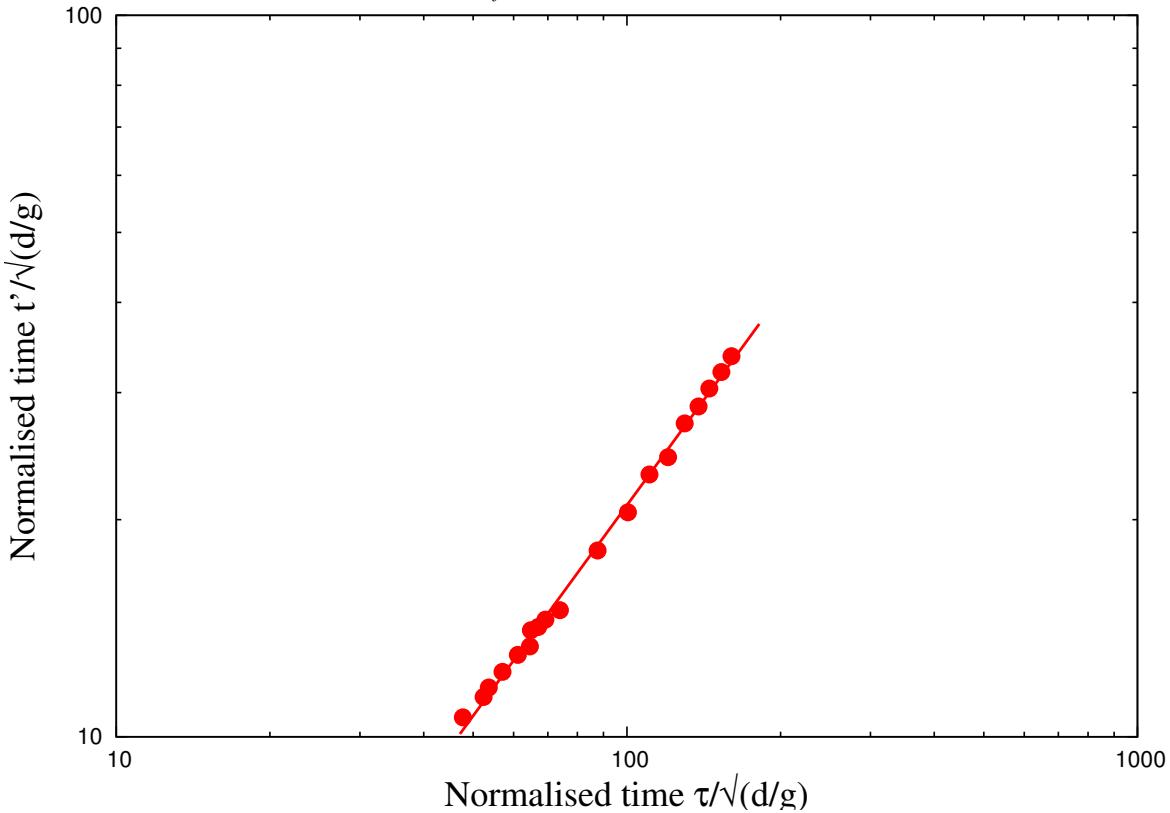
(a) Power law evolution of  $t'_f$  and  $\tau$  as a function of kinetic energy  $E_{kx0}$ .(b) Linear relationship between decay time and run-out time after the transient as a function of the normalized kinetic energy  $E_{kx0}$ .

Figure 3.28 Decay time and run-out time as a function of the normalised kinetic energy  $E_{kx0}$ .

---

3.4 Summary

91

25 explained by the observation that the dissipation rate may reach a saturation point where the  
26 dilation of the granular material and rolling of the particles change in response to the increase  
27 of the friction coefficient [Estrada et al. \(2008\)](#).

28

---

**29 Mode of dissipation**

30 The choice of this geometry was motivated by our main goal to focus on the effect of an input  
31 energy on the consecutive dynamics of a granular material. For the range of input energies  
32 investigated in this pushing test by means of contact dynamics simulations, we observed a  
1 power-law dependence of the run-out distance and time with non-trivial exponents. This is  
2 a central result of this work as it reveals that the power-law behaviour is a generic feature of  
granular dynamics. The values of the exponents are not simple functions of the geometry.

3 We also evidenced two regimes with different values of the exponents: a low-energy  
4 regime and a high-energy regime. The first regime reflects mainly the destabilization of the  
5 pile by the quake with a run-out time independent of the input energy whereas the second  
6 regime is governed by the spreading dynamics induced by the higher value of the input energy.  
7 We showed that the evolution of the pile in this high-energy regime can be described by a  
8 characteristic decay time and the energy available at the end of the first stage where the pile is  
destabilized by the quake.

9 This work may be pursued along two directions: 1) experimental realization of a similar  
10 setup with different modes of energy injection and 2) investigating the effect of various particle  
11 shapes or the presence of an ambient fluid. Although numerical simulations are generally  
12 reliable with realistic results found in the past studies of steady flows, we believe that the  
13 transients are more sensitive situations than steady states and the experiments are necessary  
14 for checking the validation of the results suggested by the simulations. Provided a convenient  
15 method is used for supplying kinetic energy homogeneously into a pile, our configuration is  
also interesting for the investigation of the behavior of a pile immersed in a viscous fluid.

**3.3.5 Effect of material points**

17

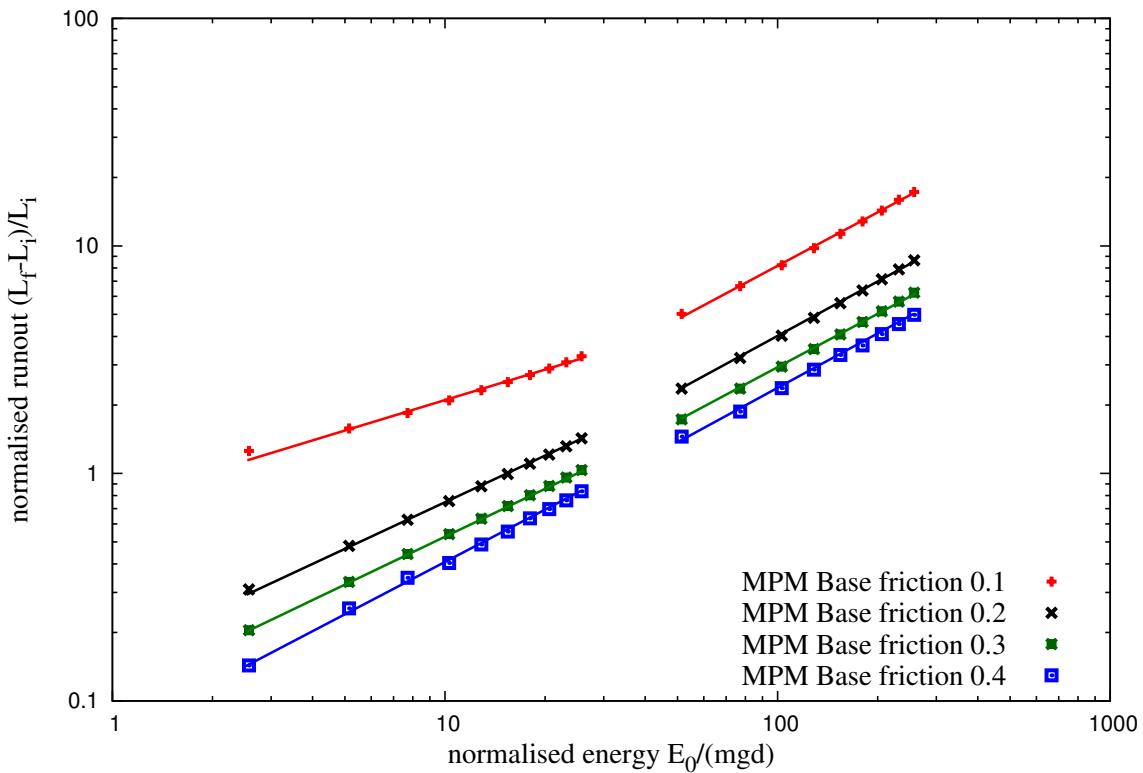
**3.3.6 Comparison with granular column collapse**

18

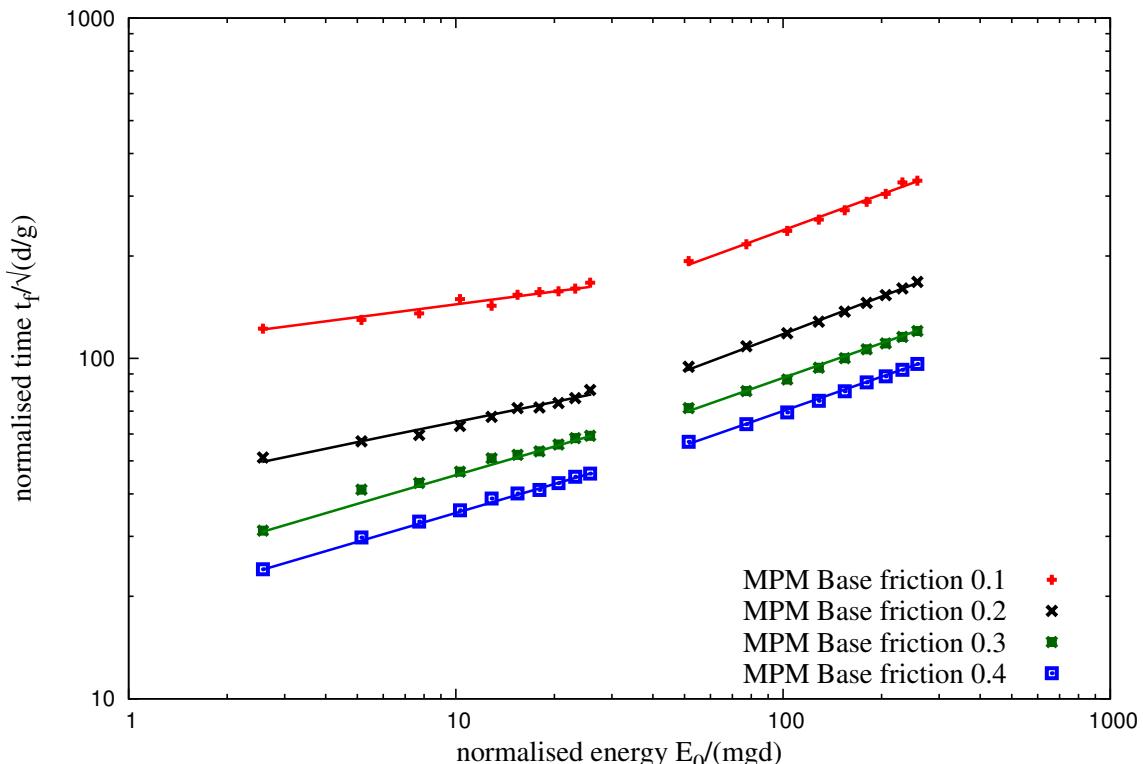
**3.4 Summary**

19

20 Multi-scale simulation of granular column collapse was performed to understand the ability and  
21 limitations of continuum models to capture the micro-mechanics of dense granular flows. The



(a) Effect of friction on the run-out distance



(b) Effect of friction on the duration of run-out.

Figure 3.29 Effect of friction on the run-out behaviour

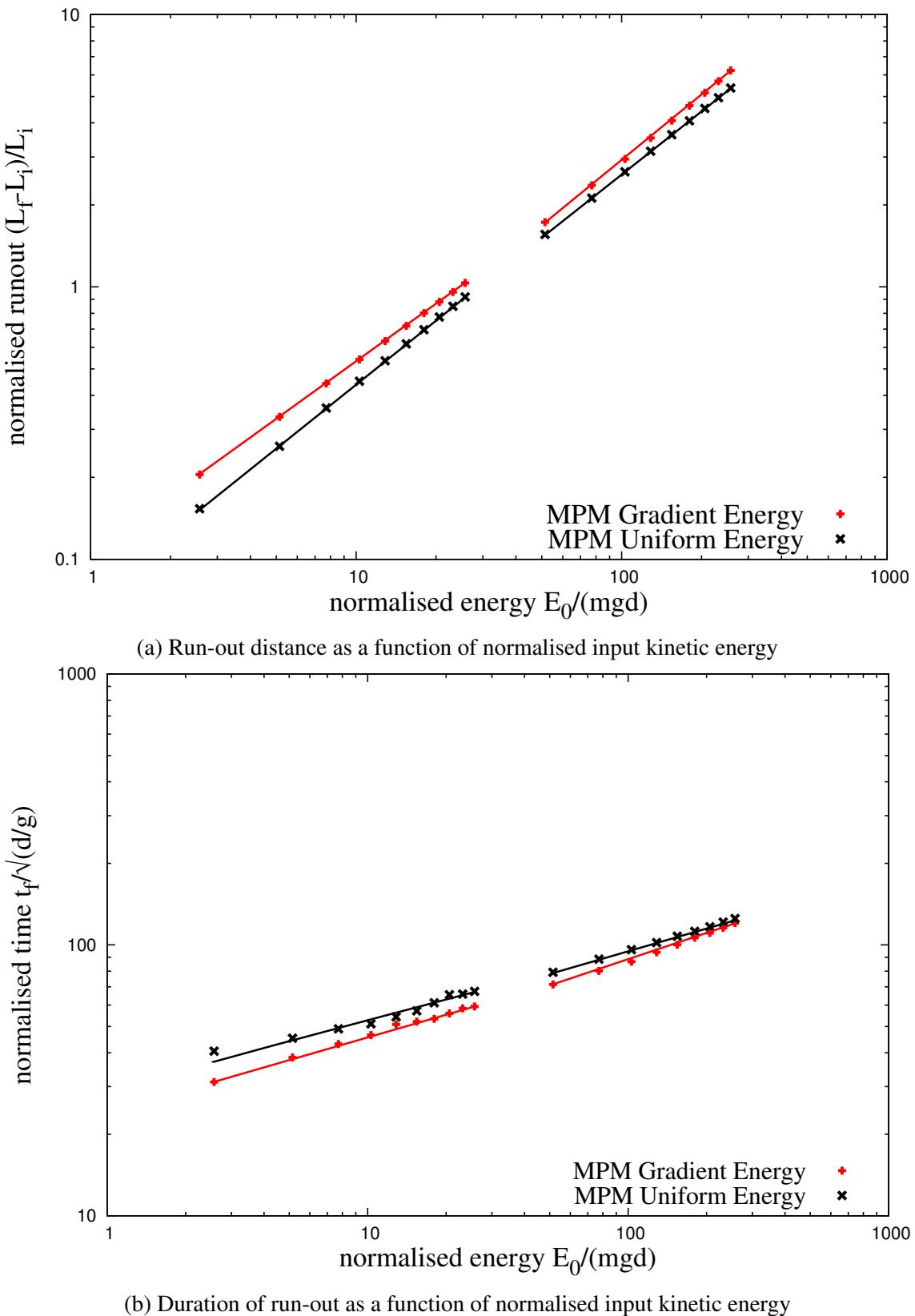


Figure 3.30 Effect of input velocity distribution on the run-out behaviour

$t = 0 \text{ s}$



$t = 0.03 \text{ s}$



$t = 0.06 \text{ s}$



$t = 0.09 \text{ s}$



Figure 3.31 Snapshots of MPM simulations of the evolution of granular pile subjected to a gradient impact energy  $E_0 = 61 \text{ mgd}$ .

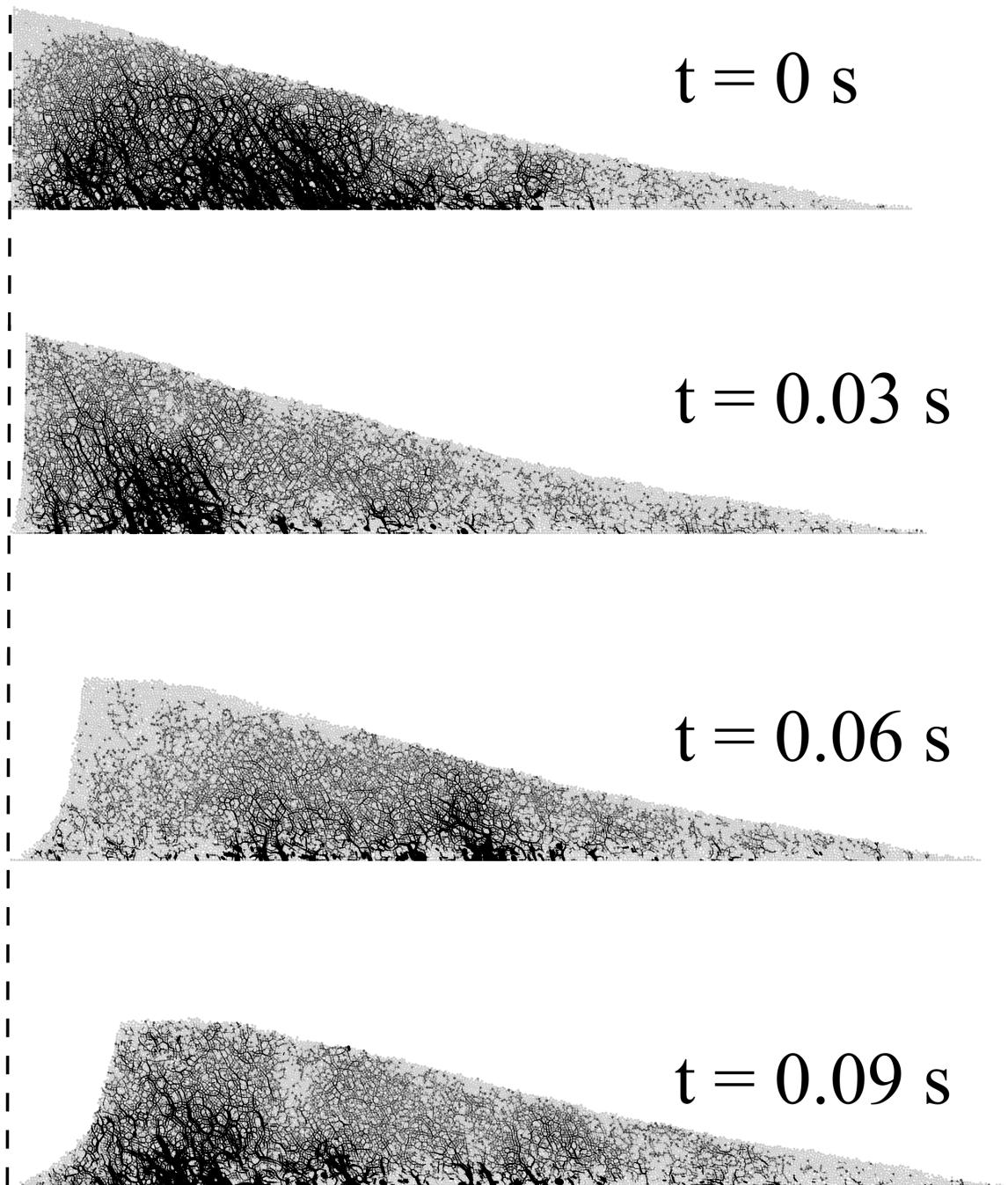


Figure 3.32 Snapshots of DEM simulations of the evolution of granular pile subjected to a gradient impact energy  $E_0 = 61 \text{ mgd}$ .

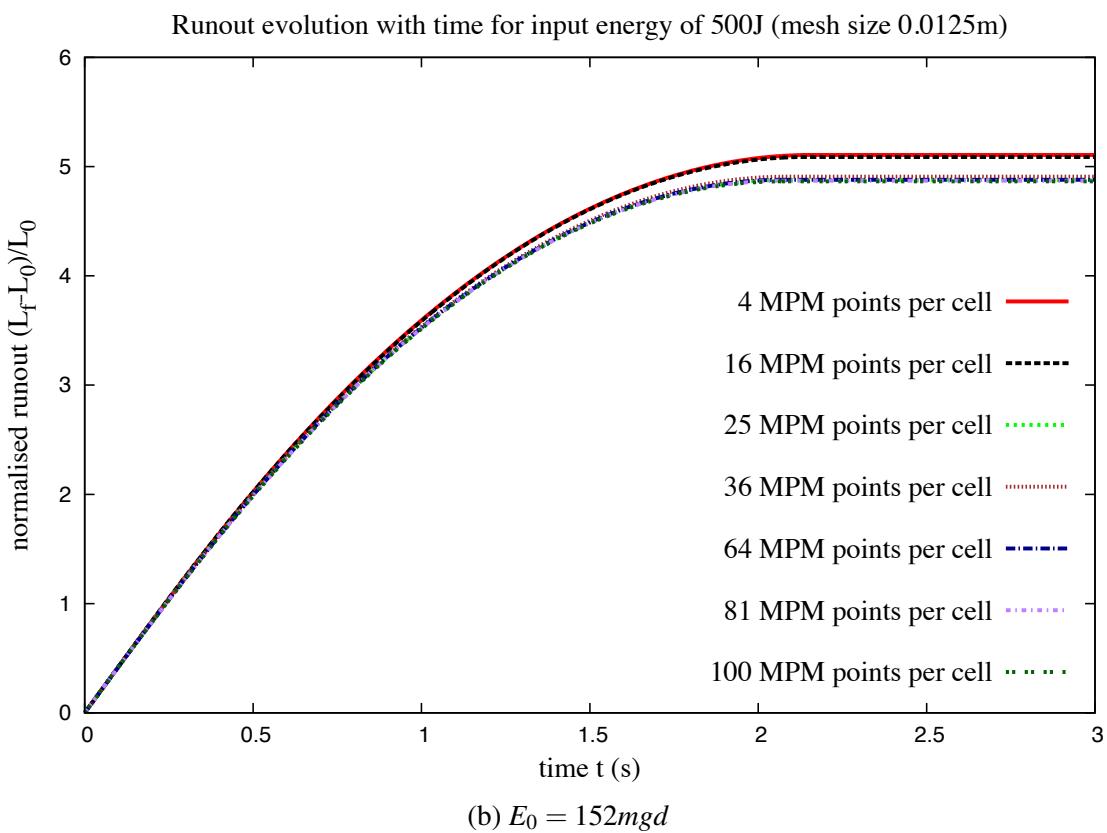
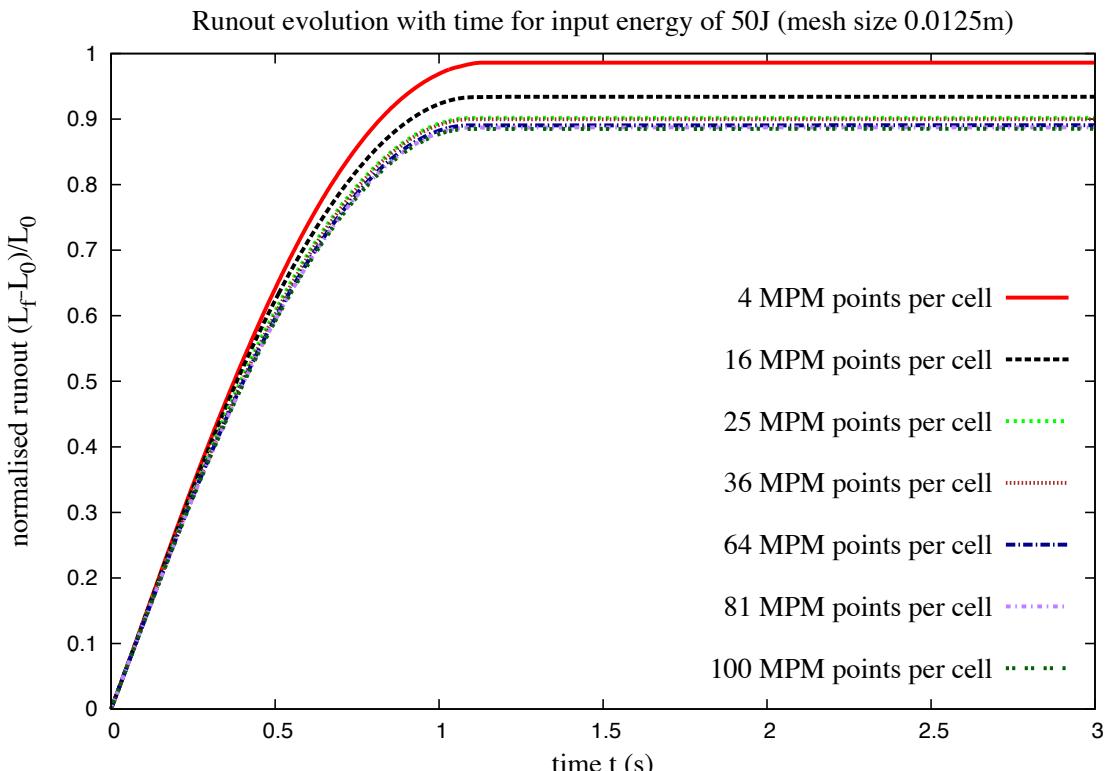


Figure 3.33 Evolution of run-out with time for varying material points per cell.

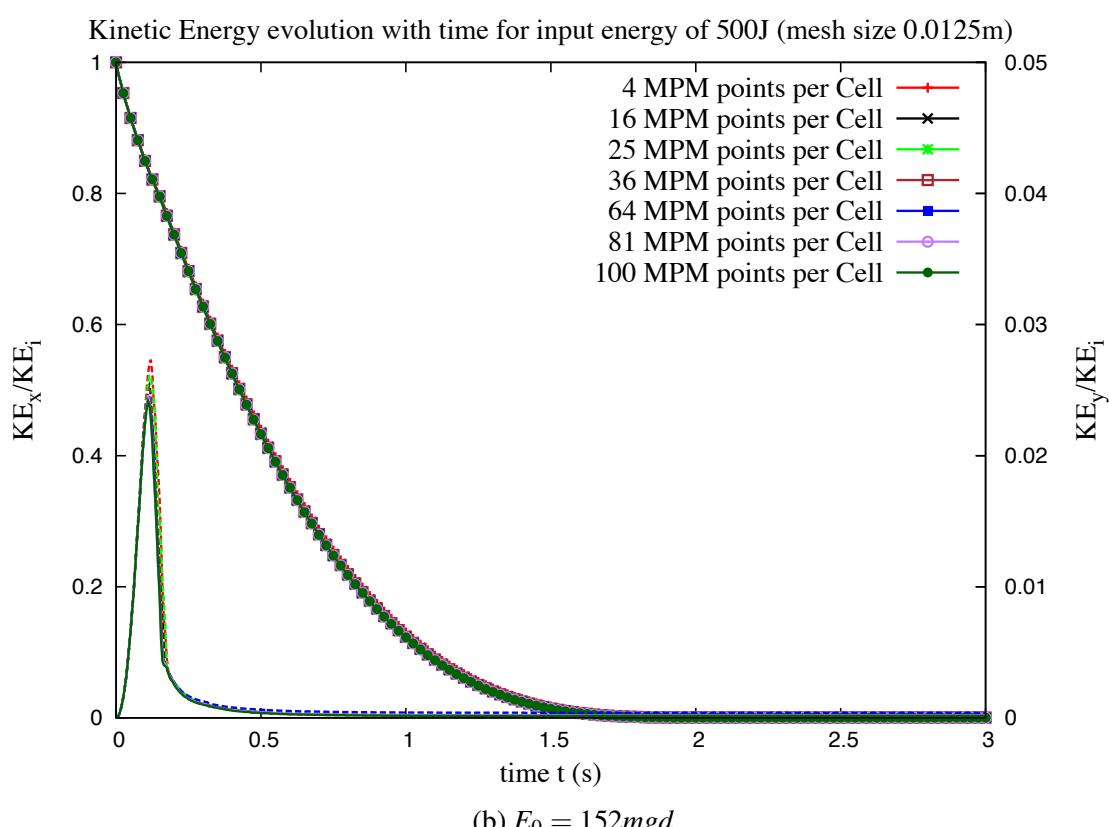
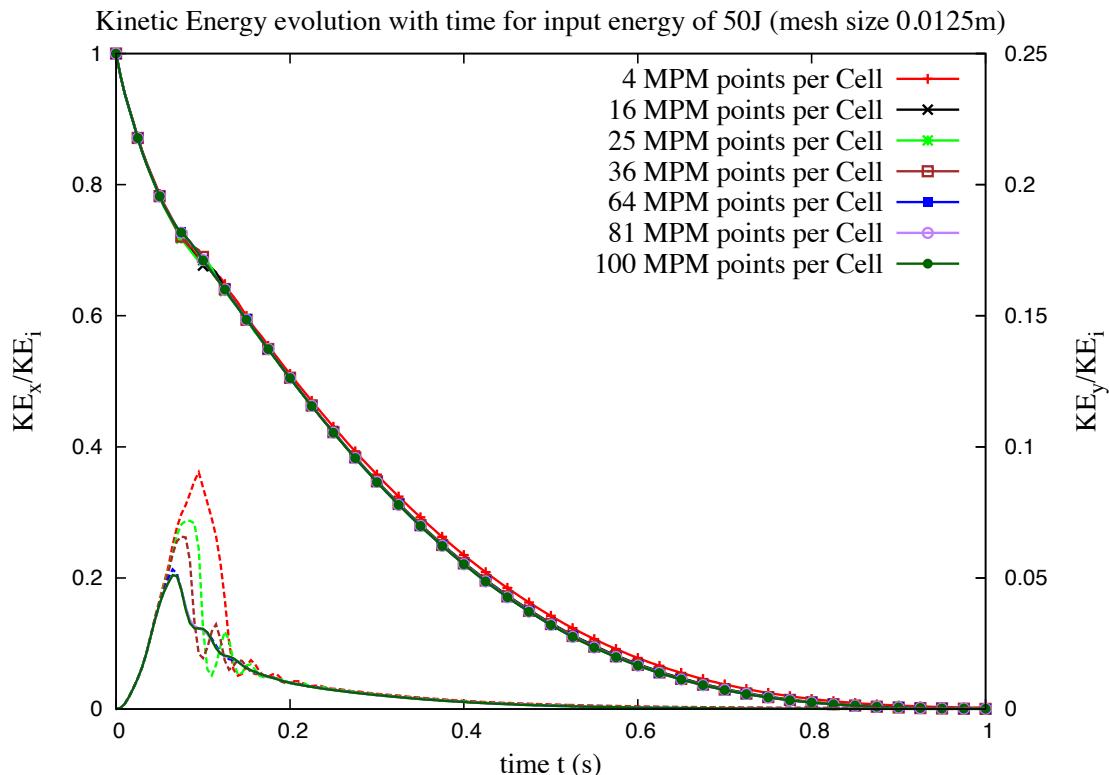


Figure 3.34 Evolution of kinetic with time for varying material points per cell

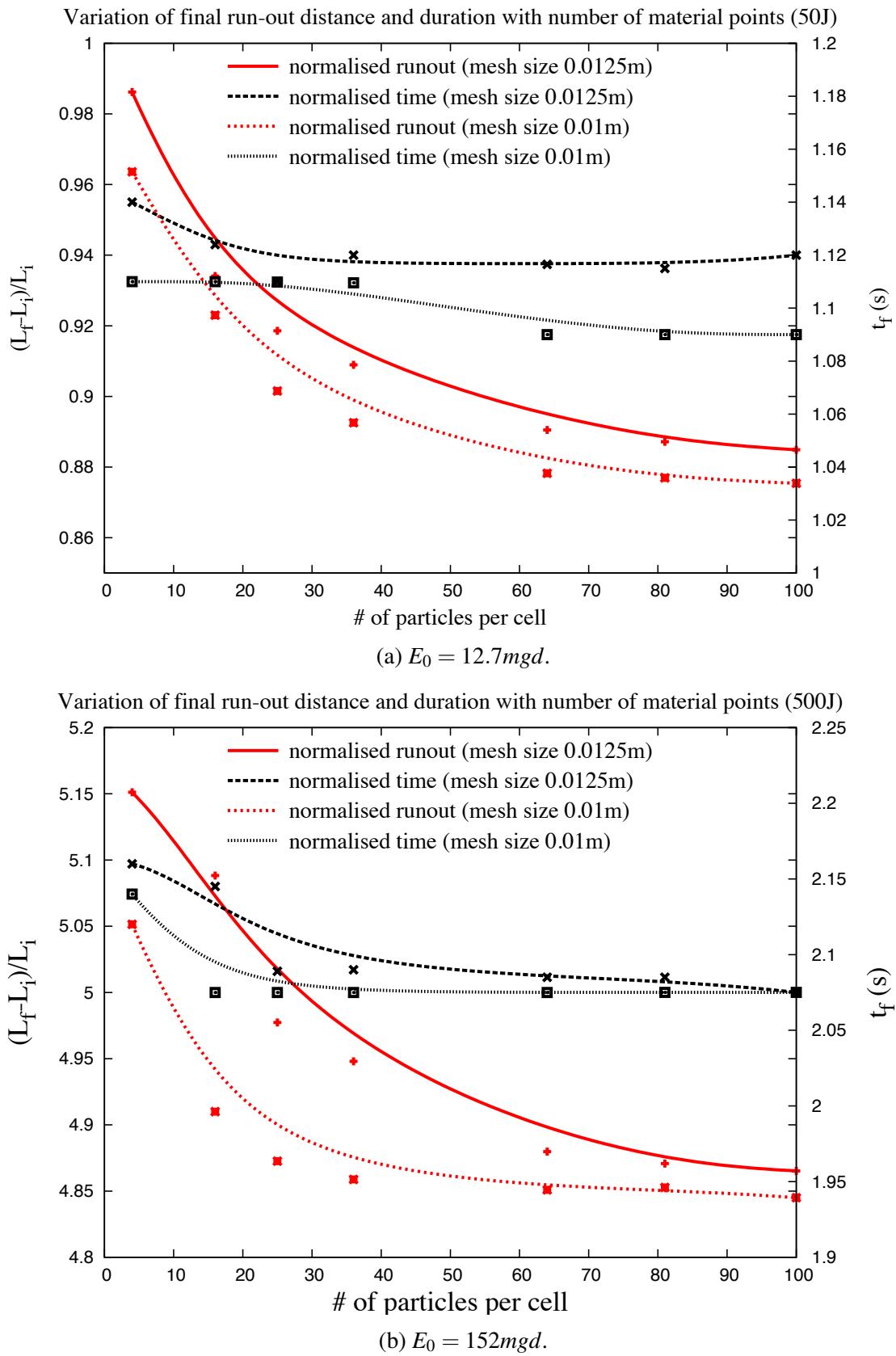


Figure 3.35 Evolution of run-out and duration of flow for varying material points per cell.

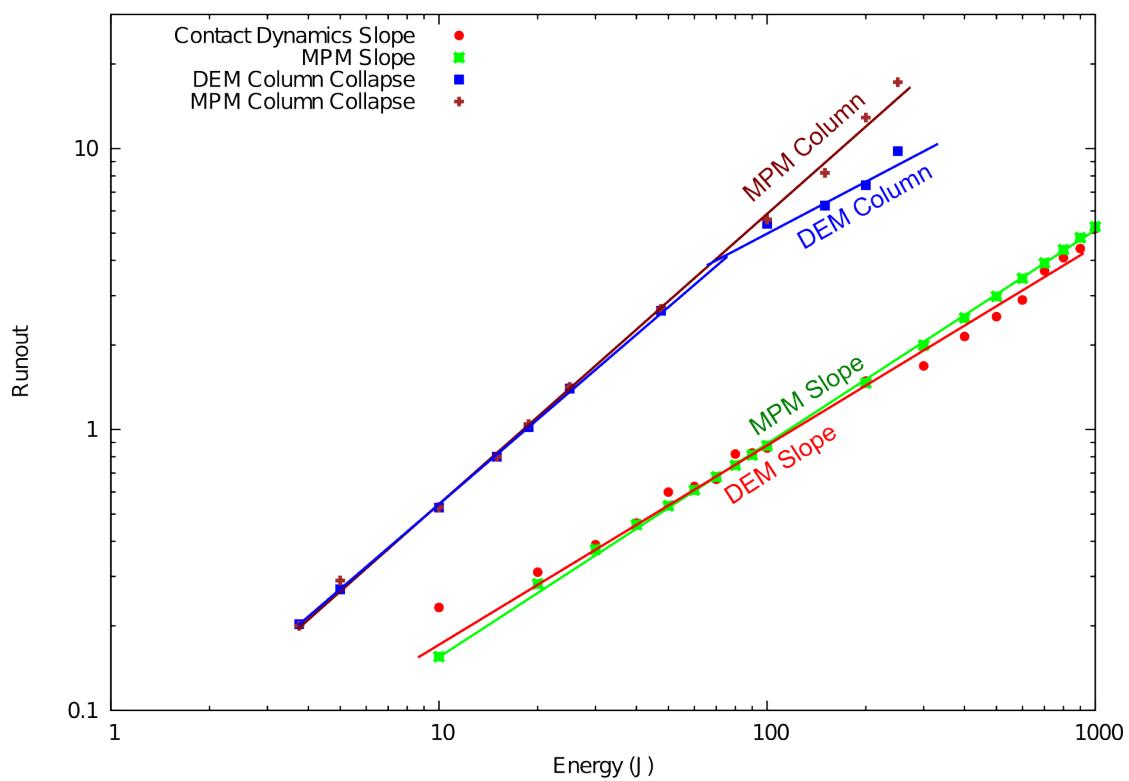


Figure 3.36 Comparison of column collapse with slope subjected to impact loading.

run-out behaviour predicted by both continuum and DEM simulations matches for columns with small aspect ratios, where the dissipation is predominantly frictional. However, MPM predicts larger run-out distances for columns with higher aspect ratios. Energy evolution

22

23

24

1 studies using DEM simulations reveal that the run-out behaviour is independent of frictional  
2 properties of the granular material and collision predominates the initial free-fall regime. The  
3 lack of a collisional energy dissipation mechanism in MPM results in over prediction of run-out  
4 distances.

## 5 Chapter 4

### 6 Numerical modelling of fluid–grain 7 interactions

#### 8 4.1 Fluid simulation using lattice Boltzmann method

9 Grain–fluid systems can be found in many scientific and engineering applications, such as  
10 suspensions, fluidised beds, sediment transport, and geo-mechanical problems. In general, the  
11 fundamental physical phenomena in these systems are not well understood mainly due to the  
12 intricate complexity of grain–fluid interactions and the lack of powerful analysis tools (Han  
13 et al., 2007a). In addition to the interaction among soil grains, the motion of soil grains is  
14 mainly driven by gravity and the hydrodynamic force exerted by the fluid. The fluid flow  
15 pattern can be significantly affected by the presence of soil grains and this often results in a  
16 turbulent flow. Hence, the development of an effective numerical framework for modelling  
17 both the fluid flow patterns and the grain–fluid interactions is very challenging.

18 Development of a numerical framework depends crucially on the size of the soil grains  
19 relative to the domain/mesh size (Feng et al., 2007). Traditionally, the Navier-Stokes equation  
20 is solved by a grid-based Computational Fluid Dynamics (CFD) method, such as the Finite  
21 Volume Method, FVM, (Capelatro and Desjardins, 2013) or a mesh-free technique such as  
22 Smooth Particle Hydrodynamics (SPH) (Sun et al., 2013). The grid size in FVM or the smooth  
23 length in SPH for discretisation of the Navier-Stokes equation is at least an order of magnitude  
24 larger than the grain diameter (Xiong et al., 2014).

25 In situations where the average domain concentration phase is far from dilute, the com-  
26 putational effort is mostly devoted to the grain dynamics. The hydrodynamic forces on the  
27 soil grains are applied based on an empirical relation using the domain-averaged local poros-  
28 ity of the soil grains in the grid. As a result, developing a fast fluid hydrodynamics solver

is unimportant for dense flows. However, most geo-mechanical problems involve complex interactions between the solid and the fluid phase. This requires accurate modelling of the fluid flow pattern. Additionally, geophysical problems, such as submarine landslides and debris flow have a relatively large simulation domain, which requires parallel computation. Implementing traditional grid-based CFD methods face great challenges on multi-processor systems (Xiong et al., 2014). Although mesh-free approaches are free from the problem of parallel scalability, its modelling accuracy and speed are relatively low when compared to grid-based CFD methods. Therefore, an accurate, fast and a highly scalable scheme is required to model fluid - grain systems in geo-mechanics.

The Navier-Stokes equation describes the motion of a non-turbulent Newtonian fluid. The equation is obtained by applying Newton's second law to the fluid motion, together with an assumption that the fluid stress is the sum of the viscous term, proportional to the gradient of the velocity, and the pressure term. Conventional CFD methods compute pertinent flow fields, such as velocity  $u$  and pressure  $p$ , by numerically solving the Navier-Stokes equation in space  $x$  and time  $t$ . Alternatively, the transport equation or the Boltzmann equation, which deals with a single particle distribution function  $f(x, \xi, t)$  in phase space  $(x, \xi)$  and time  $t$ , can be used to solve various problems in fluid dynamics.

The Lattice Boltzmann Method (LBM) (Chen and Doolen, 1998; Han et al., 2007b; He and Luo, 1997a,b; Mei et al., 2000; Zhou et al., 2012) is an alternative approach to the classical Navier-Stokes solvers for fluid flows. LBM works on an equidistant grid of cells, called lattice cells, which interact only with their direct neighbours. In LBM, the discretisation of continuum equations is based on microscopic models and mesoscopic continuum theories. LBM is a special discretising scheme of the Boltzmann equation where the particle distribution functions (mass fractions) collide and propagate on a regular grid. The important aspect, however, is the *discretisation of the velocity*, which means that the particle velocities are restricted to a predefined set of orientations.

The theoretical premises of the LB equation are that (1) hydrodynamics is insensitive to the details of microscopic physics, and (2) hydrodynamics can be preserved so long as the conservation laws and associated symmetries are respected in the microscopic and mesoscopic level. Therefore, the computational advantages of LBM are achieved by drastically reducing the particle velocity space  $\xi$  to only a very few discrete points without seriously degrading the hydrodynamics (Mei et al., 2000). This is possible because LBM rigorously preserves the hydrodynamic moments of the distribution function, such as mass density and momentum fluxes, and the necessary symmetries (He and Luo, 1997a,b). LBM has evolved as a comprehensive fluid solver and its theoretical aspects link well with the conventional central finite difference scheme (Cook et al., 2004).

<sup>18</sup> **4.1.1 Formulation**

<sup>19</sup> LBM is a ‘micro-particle’ based numerical time-stepping procedure for the solution of incom-  
<sup>20</sup> pressible fluid flows. Consider a 2D incompressible fluid flow with density  $\rho$  and kinematic  
<sup>21</sup> viscosity  $\nu$ , in a rectangular domain  $D$ . The fluid domain is divided into rectangular grids or  
<sup>22</sup> lattices, with the same grid length ‘ $h$ ’ in both  $x$ - and  $y$ -directions (see figure 4.1).

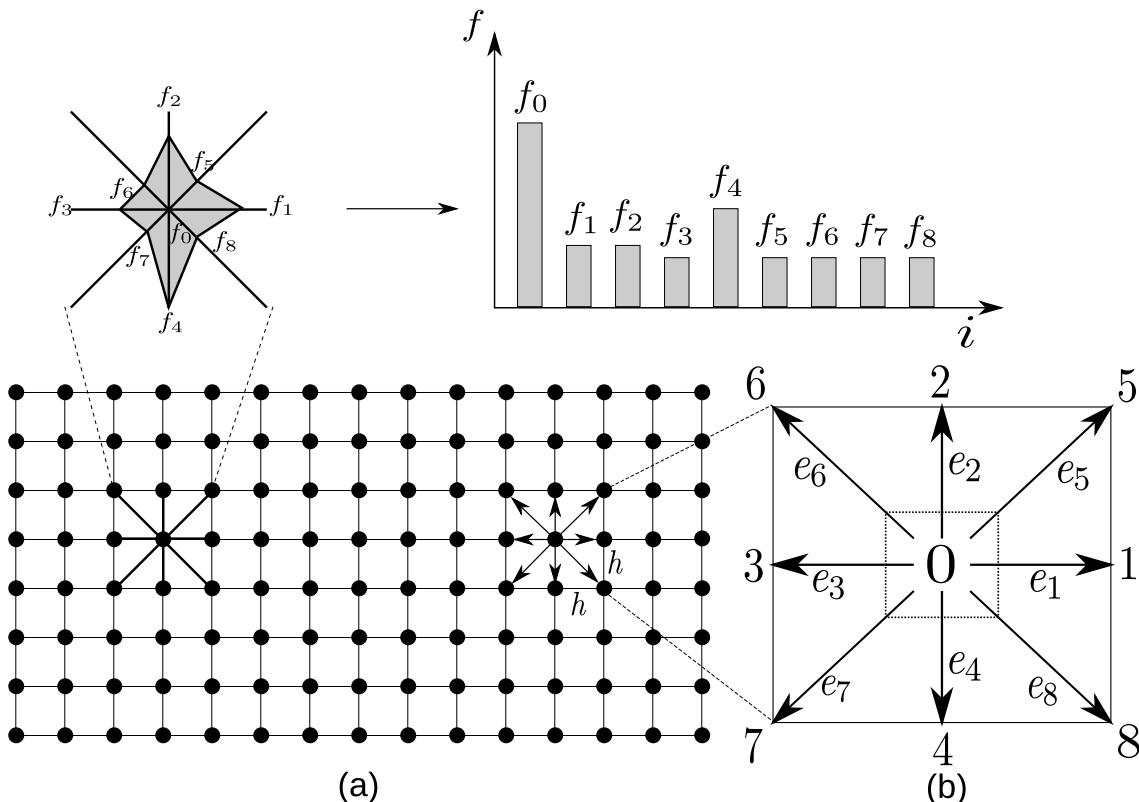


Figure 4.1 The Lattice Boltzmann discretisation and D2Q9 scheme: (a) a standard LB lattice and histogram views of the discrete single particle distribution function/direction-specific densities  $f_i$ ; (b) D2Q9 model.

These lattices are usually classified in the literature using the  $D\alpha Q\beta$ -notation, where  $\alpha$  denotes the space dimensionality and  $\beta$  is the number of discrete velocities (but also including the possibility of having particles at rest) within the momentum discretisation. The most common lattices are the  $D2Q9$  and the  $D3Q19$ -models, see He et al. (1997). The present study focuses on two-dimensional problems, hence the  $D2Q9$  momentum discretisation is adopted.

LBM discretises the Boltzmann equation in space to a finite number of possible particle spatial positions, microscopic momenta, and time. Particle positions are confined to the lattice nodes. The fluid particles at each node are allowed to move to their eight intermediate neighbours with eight different velocities  $e_i (i = 1, \dots, 8)$ . A particle can remain at its own node,

which is equivalent to moving with zero velocity  $e_o$ . The particle mass is uniform, hence these microscopic velocities and momentum are always effectively equivalent (Han et al., 2007b). Referring to the numbering system shown in figure 4.1, the nine discrete velocity vectors are defined as

$$\begin{cases} e_0 = (0, 0); \\ e_1 = C(1, 0); e_2 = C(0, 1); e_3 = C(-1, 0); e_4 = C(0, -1); \\ e_5 = C(1, 1); e_6 = C(-1, 1); e_7 = C(-1, -1); e_8 = C(1, -1), \end{cases} \quad (4.1)$$

where  $C$  is the lattice speed that is defined as  $C = h/\Delta t$ , and  $\Delta t$  is the discrete time step. The primary variables in LB formulation are called the *fluid density distribution functions*,  $f_i$ , each representing the probable amount of fluid particles moving with the velocity  $e_i$  along the direction  $i$  at each node. The macroscopic variables are defined as functions of the particle distribution function (see figure 4.1)

$$\begin{cases} \rho = \sum_{i=0}^{\beta-1} f_i & \text{(macroscopic fluid density)} \\ \text{and} \\ \overrightarrow{u} = \frac{1}{\rho} \sum_{i=0}^{\beta-1} f_i \overrightarrow{e}_i & \text{(macroscopic velocity)}, \end{cases} \quad (4.2)$$

where  $i \in [0, \beta - 1]$  is an index spanning the discretised momentum space. There are nine fluid density distribution functions,  $f_i (i = 0, \dots, 8)$ , associated with each node in the *D2Q9* model. The evolution of the density distribution function at each time step for every lattice point is governed by

$$f_i(\mathbf{x} + \mathbf{e}_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) - \frac{1}{\tau} [f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)] \quad (i = 0, \dots, 8), \quad (4.3)$$

where for any grid node  $\mathbf{x}$ ,  $\mathbf{x} + \mathbf{e}_i \Delta t$  is its nearest node along the direction  $i$ .  $\tau$  is a non-dimensional relaxation time parameter, which is related to the fluid viscosity; and  $f_i^{eq}$  is termed as the equilibrium distribution function that is defined as

$$\begin{cases} f_0^{eq} = w_0 \rho (1 - \frac{3}{2C^2} \mathbf{v} \cdot \mathbf{v}) \\ \text{and} \\ f_i^{eq} = w_i \rho (1 + \frac{3}{C^2} \mathbf{e}_i \cdot \mathbf{v} \frac{9}{2C^2} (\mathbf{e}_i \cdot \mathbf{v})^2 - \frac{3}{2C^2} \mathbf{v} \cdot \mathbf{v}) \quad (i = 0, \dots, 8), \end{cases} \quad (4.4)$$

## 4.1 Fluid simulation using lattice Boltzmann method

105

<sup>15</sup> in which,  $w_i$  represents the fixed weighting values:

<sup>16</sup>

$$w_0 = \frac{4}{9}, \quad w_{1,2,3,4} = \frac{1}{9}, \quad \text{and} \quad w_{5,6,7,8} = \frac{1}{36}. \quad (4.5)$$

The right-hand side of [eq. 4.3](#) is often denoted as  $f_i(\mathbf{x}, t_+)$  and termed the post collision distribution. LBM ensures conservation of total mass and total momentum of the fluid particles at each lattice node (see [eq. 4.3](#)). The lattice Boltzmann modelling consists of two phases: *collision* and *streaming*. The collision phase computed in the right-hand side of [eq. 4.3](#) involves only those variables that are associated with each node  $\mathbf{x}$ , and therefore is a local operation. The streaming phase then explicitly propagates the updated distribution functions at each node to its neighbours  $\mathbf{x} + e_i \Delta t$ , where no computations are required and only data exchange between neighbouring nodes are necessary. These features, together with the explicit time-stepping nature and the use of a regular grid, make LB computationally efficient, simple to implement and easy to parallelise ([Han et al., 2007b](#)).

The streaming step involves the translation of the distribution functions to their neighbouring sites according to the respective discrete velocity directions, as illustrated in figure [4.2](#) in the *D2Q9* model. The collision step, (see [figure 4.3](#)) consists of re-distribution the local discretised Maxwellian equilibrium functions in such a way that local mass and momentum are invariants. In incompressible flows, the energy conservation is equivalent to the momentum conservation ([He et al., 1997](#)).

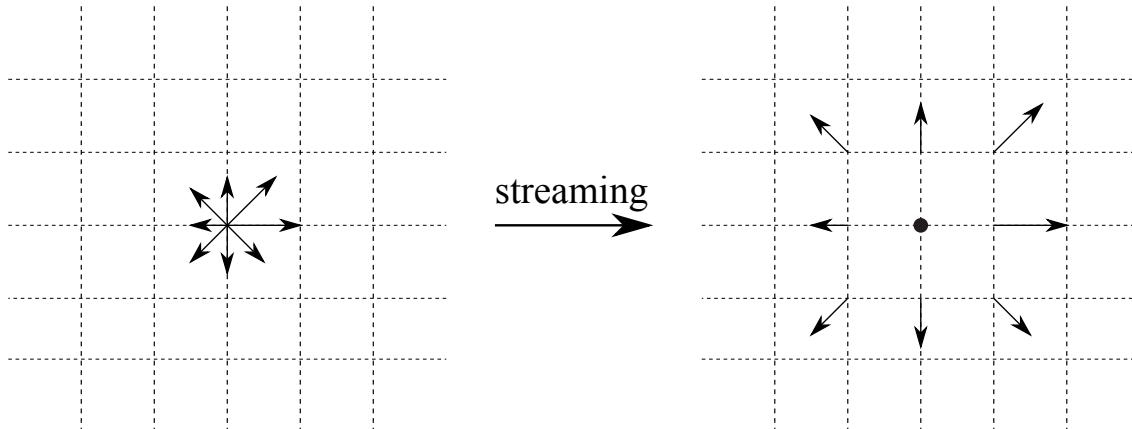


Figure 4.2 Illustration of the streaming process on a *D2Q9* lattice. The magnitude of the distribution functions remains unchanged, but they move to a neighbouring node according to their direction.

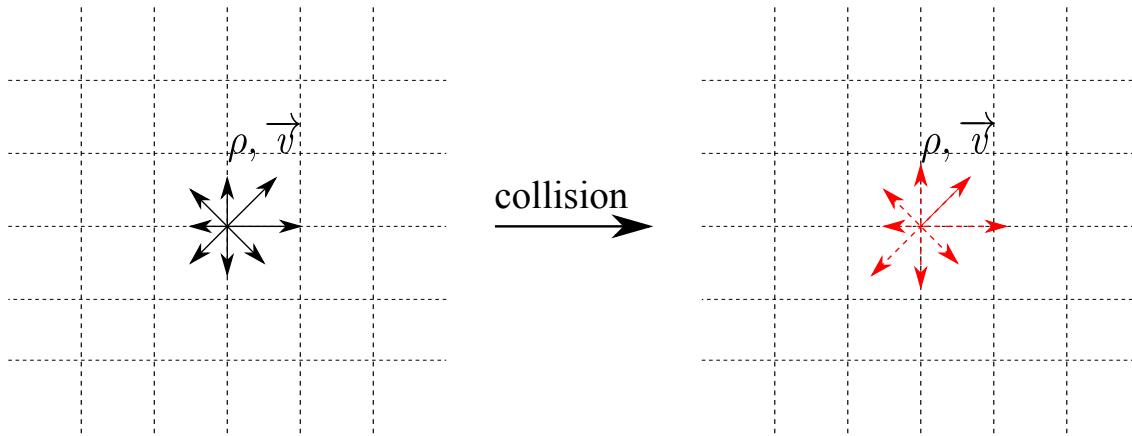


Figure 4.3 Illustration of the collision process on a  $D2Q9$  lattice. The local density  $\rho$  and velocity  $v$  are conserved, but the distribution functions change according to the relaxation to local Maxwellian rule.

The standard macroscopic fluid variables, such as density  $\rho$  and velocity  $v$ , can be recovered from the distribution functions as

$$\rho = \sum_{i=0}^8 f_i, \quad \text{and} \quad \rho \mathbf{v} = \sum_{i=0}^8 f_i e_i. \quad (4.6)$$

The fluid pressure field ‘ $p$ ’ is determined by the equation of state

$$p = C_s^2 \rho, \quad (4.7)$$

where  $C_s$  is termed the fluid speed of sound and is related to the lattice speed  $C$  as

$$C_s = C/\sqrt{3}. \quad (4.8)$$

The kinematic viscosity of the fluid  $\nu$  is implicitly determined by the model parameters  $h$ ,  $\Delta t$  and  $\tau$  as

$$\nu = \frac{1}{3}(\tau - \frac{1}{2}) \frac{h^2}{\Delta t} = \frac{1}{3}(\tau - \frac{1}{2}) Ch, \quad (4.9)$$

which indicates that these three parameters are related to each other and have to be appropriately selected to represent the correct fluid viscosity. An additional constraint to the parameter selection is the lattice speed  $C$ , which must be sufficiently large in comparison to the maximum fluid velocity  $v_{max}$ , to ensure accuracy of the solution. The ‘computational’ Mach number,  $M_a$ , defined as

$$M_a = \frac{v_{max}}{C}. \quad (4.10)$$

Theoretically, for an accurate solution, the Mach number is required to be  $<< 1$ . In practice,  $M_a$  should be at least smaller than 0.1 (He et al., 1997). From a computational point of view, it is more convenient to choose  $h$  and  $\tau$  as two independent parameters and  $\Delta t$  as the derived parameter

$$\Delta t = (\tau - \frac{1}{2}) \frac{h^2}{3v}. \quad (4.11)$$

It can be observed that  $\tau$  has to be greater than 0.5 (He et al., 1997). Since there is no a priori estimation available to determine appropriate values of  $h$  and  $\tau$ , for a given fluid flow problem and a known fluid viscosity  $v$ , a *trial and error* approach is employed to ensure a smaller *Mach Number*. This is similar to choosing an appropriate Finite Element mesh size, without using automatic adaptive mesh techniques.

### 4.1.2 Lattice Boltzmann - Multi-Relaxation Time (LBM-MRT)

The Lattice Boltzmann Bhatnagar-Gross-Krook (LGBK) method is capable of simulating various hydrodynamics, such as multiphase flows and suspensions in fluid (Succi, 2001; Succi et al., 1989). However, LBM suffers from numerical instability when the dimensionless relaxation time  $\tau$  is close to 0.5. The Lattice Boltzmann Method – Multi-Relaxation Time (LBM-MRT) overcomes the deficiencies of linearised single relaxation LBM-BGK approach, such as the fixed Prandtl number ( $Pr=v/\kappa$ ), where the thermal conductivity ‘ $\kappa$ ’ is unity (Liu et al., 2003). LBM-MRT offers better numerical stability and has more degrees of freedom. In LBM-MRT the advection is mapped onto the momentum space by a linear transformation and the flux is finished within the velocity space (Du et al., 2006).

The lattice Boltzmann equation with multiple relaxation time approximation is written as

$$f_\alpha(\mathbf{x} + \mathbf{e}_i \Delta_t, t + \Delta_t) - f_\alpha(\mathbf{x}, t) = -\mathbf{S}_{\alpha i}(f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)), \quad (4.12)$$

where  $\mathbf{S}$  is the collision matrix. The nine eigen values of  $\mathbf{S}$  are all between 0 and 2 so as to maintain linear stability and separation of scales. This ensures that the relaxation times of non-conserved quantities are much faster than the hydrodynamic time scales. The LGBK model is a special case in which the nine relaxation times are all equal and the collision matrix  $\mathbf{S} = \frac{1}{\tau} \mathbf{I}$ , where  $\mathbf{I}$  is the identity matrix. The evolutionary progress involves two steps, advection and flux:

$$f_\alpha^+(\mathbf{x}, t) - f_\alpha(\mathbf{x}, t) = -\mathbf{S}_{\alpha i}(f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)) \quad (4.13)$$

$$f_\alpha(\mathbf{x} + e_\alpha \Delta_t, t + \Delta_t) = f_\alpha^+(\mathbf{x}, t). \quad (4.14)$$

The advection (eq. 4.13) can be mapped to the momentum space by multiplying with a transformation matrix  $\mathbf{M}$ . The evolutionary equation of LBM–MRT is written as

$$\mathbf{f}(\mathbf{x} + \mathbf{e}_i \Delta_t, t + \Delta_t) - \mathbf{f}(\mathbf{x}, t) = -M^{-1} \hat{\mathbf{S}}(\hat{\mathbf{f}}(\mathbf{x}, t) - \hat{\mathbf{f}}^{eq}(\mathbf{x}, t)), \quad (4.15) \quad 10$$

where  $\mathbf{M}$  is the transformation matrix mapping a vector  $\mathbf{f}$  in the discrete velocity space  $\mathbb{V} = \mathbb{R}^b$  to a vector  $\hat{\mathbf{f}}$  in the moment space  $\mathbb{V} = \mathbb{R}^b$ .

$$\hat{\mathbf{f}} = \mathbf{M}\mathbf{f}, \quad (4.16) \quad 13$$

$$\mathbf{f}(\mathbf{x}, t) = [f_0(\mathbf{x}, t), f_1(\mathbf{x}, t), \dots, f_8(\mathbf{x}, t)]^T. \quad (4.17) \quad 14$$

The collision matrix  $\hat{\mathbf{S}} = M S M^{-1}$  in moment space is a diagonal matrix:

$$\hat{\mathbf{S}} = \text{diag}[s_1, s_2, s_3, \dots, s_9]. \quad (4.17) \quad 17$$

The transformation matrix  $\mathbf{M}$  can be constructed via Gram-Schmidt orthogonalisation procedure. The general form of the transformation matrix  $\mathbf{M}$  can be written as

$$\mathbf{M} = [|p\rangle, |e\rangle, |e^2\rangle, |u_x\rangle, |q_x\rangle, |u_y\rangle, |q_y\rangle, |p_{xx}\rangle, |p_{xy}\rangle]^T, \quad (4.18)$$

whose elements are,

$$|p\rangle = |e_\alpha|^0 \quad (4.19a)$$

$$|e\rangle_\alpha = Q e_\alpha^2 - b_2 \quad (4.19b)$$

$$|e^2\rangle_\alpha = a_1(Q e_\alpha^4 - b_6) + a_2(Q e_\alpha^4 - b_6) \quad (4.19c)$$

$$|u_x\rangle_\alpha = e_{\alpha,x} \quad (4.19d)$$

$$|q_x\rangle_\alpha = (b_1 e_\alpha^2 - b_3) e_{\alpha,x} \quad (4.19e)$$

$$|u_y\rangle_\alpha = e_{\alpha,y} \quad (4.19f)$$

$$|q_y\rangle_\alpha = (b_1 e_\alpha^2 - b_3) e_{\alpha,y} \quad (4.19g)$$

$$|p_{xx}\rangle_\alpha = d e_{\alpha,x}^2 - e_\alpha^2 \quad (4.19h)$$

$$|p_{xy}\rangle_\alpha = e_{\alpha,x} e_{\alpha,y}, \quad (4.19i)$$

where  $d = 2$  and  $Q = 9$ ,  $b_1 = \sum_{\alpha=1}^Q e_{\alpha,x}^2$ ,  $b_2 = \sum_{\alpha=1}^Q e_\alpha^2$ ,  $b_3 = \sum_{\alpha=1}^Q e_\alpha^2 e_{\alpha,x}^4$ ,  $a_1 = ||e^2||^2$ , and  $a_2 = \sum_{\alpha=0}^{Q-1} (Q c_\alpha^2 - b_2) \times (Q c_\alpha^4 - b_6)$ .

## 4.1 Fluid simulation using lattice Boltzmann method

109

<sup>15</sup> Explicitly, the transformation matrix can be written as

$$\begin{aligned} \text{16} \quad \mathbf{M} = & \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \quad (4.20) \\ \text{17} \end{aligned}$$

<sup>18</sup> The corresponding equilibrium distribution functions in moment space  $\widehat{\mathbf{f}}^{eq}$  is given as

$$\text{19} \quad \widehat{\mathbf{f}}^{eq} = [\rho_0, e^{eq}, e^{2eq}, u_x, q_x^{eq}, q_y^{eq}, p_{xx}^{eq}, p_{xy}^{eq}]^T, \quad (4.21)$$

<sup>20</sup> where

$$\text{21} \quad e^{eq} = \frac{1}{4}\alpha_2 p + \frac{1}{6}\gamma_2(u_x^2 + y_y^2) \quad (4.22a)$$

$$\text{22} \quad e^{2eq} = \frac{1}{4}\alpha_3 p + \frac{1}{6}\gamma_4(u_x^2 + y_y^2) \quad (4.22b)$$

$$\text{23} \quad q_x^{eq} = \frac{1}{2}c_1 u_x \quad (4.22c)$$

$$\text{24} \quad q_y^{eq} = \frac{1}{2}c_2 u_y \quad (4.22d)$$

$$\text{25} \quad p_{xx}^{eq} = \frac{3}{2}\gamma_1(u_x^2 - u_y^2) \quad (4.22e)$$

$$\text{26} \quad p_{xy}^{eq} = \frac{3}{2}\gamma_3(u_x u_y). \quad (4.22f)$$

<sup>28</sup> To get the correct hydrodynamic equation, the values of the co-efficients are chosen as  $\alpha_2 = 24$ ,  
<sup>29</sup>  $\alpha_3 = -36$ ,  $c_1 = c_2 = -2$ ,  $\gamma_1 = \gamma_3 = 2/3$ ,  $\gamma_2 = 18$  and  $\gamma_4 = -18$ . The values of the elements in  
<sup>30</sup> the collision matrix are:  $s_8 = s_9 = \tau$  and  $s_1 = s_4 = s_6 = 1.0$  and the others vary between 1.0  
and 2.0 for linear stability. Through the Chapman-Enskog expansion (Du et al., 2006), the  
incompressible Navier-Stokes equation can be recovered and the viscosity is given as

$$\nu = c_s^2 \Delta t (\tau - 0.5). \quad (4.23) \quad \text{3}$$

### 4.1.3 Boundary conditions

Boundary conditions (BC) form an important part of any numerical technique. In many cases, the boundary conditions can strongly influence the accuracy of the algorithm. Velocity and pressure are not the primary variables in LBM, hence the standard pressure, velocity, and mixed boundary conditions cannot be imposed directly. Alternative conditions in terms of the distribution functions are adopted to describe the boundary conditions.

#### Periodic boundary condition

The simplest type of boundary condition is the periodic boundary. In this case, the domain is folded along the direction of the periodic boundary pair. For boundary nodes, the neighbouring nodes are on the opposite boundary, using the normal referencing of neighbours (see figure 4.1a).

- <sup>1</sup> From the perspective of submarine landslide modelling, the periodic boundary conditions are
- <sup>2</sup> useful for preliminary analysis, as they imply a higher degree of symmetry of the fluid domain.
- <sup>3</sup> Further information on the periodic boundary condition can be found in Aidun et al. (1998).

#### No-slip boundary condition

The most commonly adopted BC in the lattice Boltzmann approach is the no-slip BC, especially the simple bounce-back rule, which is quite elegant and surprisingly accurate. The basic idea is that the incoming distribution functions at a wall node are reflected back to the original fluid nodes, but with the direction rotated by  $\pi$  radians. The bounce-back boundary condition is one of the benefits of LBM, as it is trivial to implement and it allows one to effortlessly introduce obstacles into the fluid domain. However, the boundary conditions have been proven to be only first-order accurate in time and space (Pan et al., 2006). A straightforward improvement is to consider the wall-fluid interface to be situated halfway between the wall and the fluid lattice nodes (Ziegler, 1993). It involves defining the *solid* nodes as those lying within the stationary wall regions, and the *fluid* nodes otherwise. Then, if  $i$  is the direction between a fluid node  $n_1$  and a solid node  $n_2$ , the bounce-back rule requires that the incoming fluid particle from  $n_1$  to  $n_2$  be reflected back along the direction it came from, i.e.,

$$f_{-i}(\mathbf{x}, t + \Delta t) = f_i(\mathbf{x}, t_+), \quad (4.24)$$

where  $-i$  denotes the opposite direction of  $i$ . The bounce back rule is illustrated in figure 4.4. This simple rule ensures that no tangential velocity exists along the fluid-wall interface, thereby a non-slip condition is imposed, and can be extended to any shapes or objects in a fluid flow (Han et al., 2007a; Zou and He, 1997). The slip boundary conditions have similar treatment to the

## 4.1 Fluid simulation using lattice Boltzmann method

111

- <sup>22</sup> non-slip condition, except that the distribution functions are reflected in the boundary instead  
<sup>23</sup> of bounce-back ([Succi, 2001](#)).

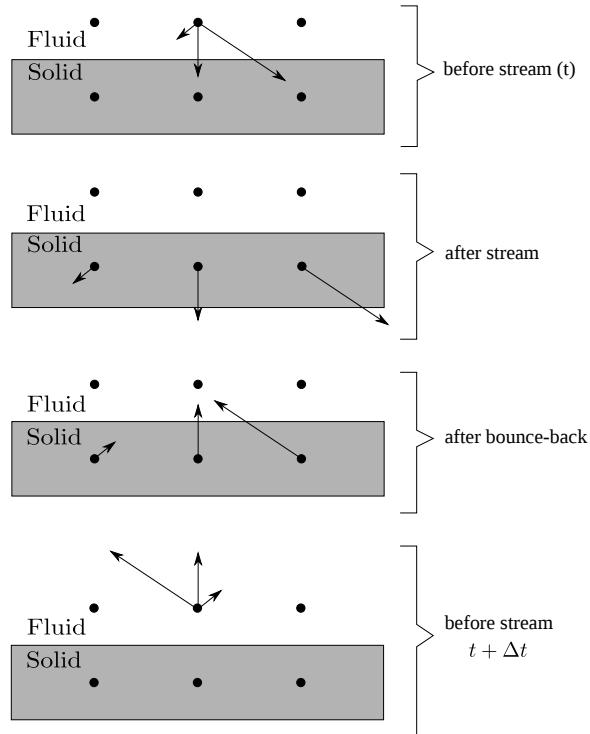


Figure 4.4 Half-way bounce back algorithm for the *D2Q9* model adopted after [Sukop and Thorne \(2006\)](#).

<sup>24</sup> **Pressure and velocity boundary condition**

- <sup>25</sup> The pressure (Dirichlet) boundary condition can be imposed in lattice Boltzmann by specifying  
<sup>26</sup> a fluid density at the pressure boundary ([Zou and He, 1997](#)). To impose a pressure boundary  
<sup>27</sup> along the y-direction (for example, consider the left hand side inlet boundary in figure 4.5), a  
<sup>28</sup> density  $\rho = \rho_{in}$  is specified from which the velocity is computed. The vertical component of  
<sup>1</sup> the velocity on the boundary is set as zero,  $u_y = 0$ . After streaming,  $f_2, f_3, f_4, f_6$ , and  $f_7$  are  
<sup>2</sup> known,  $u_x$  and  $f_1, f_5, f_8$  are to be determined from [eq. 4.2](#) as

$$f_1 + f_5 + f_8 = \rho_{in} - (f_0 + f_2 + f_3 + f_4 + f_6 + f_7) \quad (4.25) \quad 3$$

$$f_1 + f_5 + f_8 = \rho_{in} u_x + (f_3 + f_6 + f_7) \quad (4.26) \quad 4$$

$$f_5 - f_8 = f_2 - f_4 + f_6 - f_7, \quad (4.27) \quad 5$$

6

Consistency of equations (4.25) and (4.26) gives

$$u_x = 1 - \frac{[f_0 + f_2 + f_4 + 2 * (f_3 + f_6 + f_7)]}{\rho_{in}}. \quad (4.28)$$

The bounce-back rule for the non-equilibrium part of the particle distribution normal to the inlet is used to find  $f_1 - f_1^{eq} = f_3 - f_3^{eq}$ . The values of  $f_5$  and  $f_8$  can be obtained from  $f_1$ :

$$f_1 = f_3 + \frac{2}{3}\rho_{in}u_x \quad (4.29)$$

$$f_5 = f_7 - \frac{1}{2}(f_2 - f_4) + \frac{1}{6}\rho_{in}u_x \quad (4.29)$$

$$f_8 = f_6 + \frac{1}{2}(f_2 - f_4) + \frac{1}{6}\rho_{in}u_x. \quad (4.29)$$

The corner node at inlet needs some special treatment. Considering the bottom node at inlet as an example, after streaming,  $f_3, f_4, f_7$  are known;  $\rho$  is defined, and  $u_x = u_y = 0$ . The particle distribution functions  $f_1, f_2, f_5, f_6$ , and  $f_8$  are to be determined. The bounce-back rule for the non-equilibrium part of the particle distribution normal to the inlet and the boundary is used to find

$$f_1 = f_3 + (f_1^{eq} - f_3^{eq}) = f_3 \quad (4.30)$$

$$f_2 = f_4 + (f_1^{eq} - f_3^{eq}) = f_4. \quad (4.31)$$

<sup>1</sup> Using these we can compute

$$f_5 = f_7 \quad (4.32)$$

$$f_6 = f_8 = \frac{1}{2}[\rho_{in} - (f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 + f_8)]. \quad (4.33)$$

<sup>5</sup> Similar procedure can be applied to the top inlet node and the outlet nodes. Von Neumann  
<sup>6</sup> boundary conditions constrain the flux at the boundaries. A velocity vector  $u = [u_0 \ v_0]^T$  is  
<sup>7</sup> specified, from which the density and pressure are computed based on the domain. The velocity  
<sup>8</sup> boundary condition can be specified in a similar way (Zou and He, 1997). The pressure  
<sup>9</sup> and velocity boundary conditions contribute additional equation(s) to determine the unknown  
<sup>10</sup> distribution functions. In the case of velocity boundary, the boundary condition equation is  
<sup>11</sup> sufficient to determine the unknown distribution functions in the D2Q9 model, however the  
<sup>12</sup> pressure boundary conditions require additional constitutive laws to determine the unknown  
<sup>13</sup> distribution functions.

Table 4.1 LBM parameters used in simulating laminar flow through a circular pipe.

Parameter	Value
Density $\rho$	1000 kg m <sup>-3</sup>
Relaxation parameter $\tau$	0.51
Kinematic viscosity	$1 \times 10^{-6}$ m <sup>2</sup> s <sup>-1</sup>
Grid resolution ‘h’	1 <sup>-2</sup> m
Number of steps	50,000
Error in predicting horizontal velocity	0.009 %

## <sup>14</sup> 4.2 Validation of the lattice Boltzmann method

<sup>15</sup> To verify the incompressible LBM model implemented in the above section, numerical simulation  
<sup>16</sup> of a transient development of steady state Poiseuille flow in a straight channel is performed.  
<sup>17</sup> At  $t = 0$ , the LBM water particles ( $\rho = 1000\text{kg/m}^3$ ) are simulated to flow through a channel  
<sup>18</sup> of width ‘H’ (= 0.4 m) and simulation length ‘L’ (2.5H) under constant body force. Periodic  
 boundary conditions are applied at either end of the channel and the pressure gradient is set  
 to zero, which simulates the condition of a continuous flow of fluid in a closed circular pipe.  
 The length ‘L’ has no effect on the simulation as no streamwise variation is detected in the  
 solution. The parameters adopted in LBM simulation are presented in table 4.1. Sufficient time  
 is allowed for the flow to travel beyond the required development length so that the flow is  
 laminar (Durst et al., 2005). The development length  $X_D$  required for a flow to be fully laminar  
 is

$$X_D/H = [(0.619)^{1.6} + (0.0567R_e)^{1.6}]^{1/1.6}, \quad (4.34)$$

where  $R_e$  is the Reynolds number. The velocity profile at steady state is presented in figure 4.5.  
 A maximum horizontal velocity of 0.037863 m s<sup>-1</sup> is observed along the center-line of the  
 channel. The maximum horizontal velocity is compared with the closed-form based on the  
 Haygen-Poiseuille flow equation for no-slip boundary condition (Willis et al., 2008)

$$U_x = \frac{\Delta P}{2\mu L} \left[ \frac{H^2}{4} - y^2 \right], \quad (4.35)$$

where  $v_x$  is the horizontal velocity (m/s);  $\Delta P$  is the pressure gradient,  $\mu$  dynamic viscosity of  
 the fluid. LBM predicts the maximum horizontal velocity within an error of 0.009 %.

In order to further validate the accuracy of the lattice Boltzmann code, the transient  
 development of the Poiseuille’s flow is compared with the CFD simulation performed using  
 ANSYS Fluent. Finite Volume Method is a common CFD technique, which involves solving

the governing partial differential equation (Navier-Stokes) over the discretised control volume. This guarantees the conservation of fluxes over a particular control volume. The finite volume equations yield governing equations of the form

$$\frac{\partial}{\partial t} \iiint Q dV + \iint F dA = 0, \quad (4.36)$$

where  $Q$  is the vector of conserved variables,  $F$  is the vector of fluxes in the Navier-Stokes equation,  $V$  is the volume of control volume element, and  $A$  is the surface area of the control volume element.

A 2D rectangular plane of length 1 m and height 0.04 m is discretised into 400 cells of size  $1^{-2}$  m (see figure 4.6). A constant velocity is applied at the inlet. Water ( $\rho = 998.2 \text{ kg/m}^3$ , viscosity ' $\eta'$  =  $1 \times 10^{-3} \text{ Ns/m}^2$ ) is allowed to flow through the channel and it develops into a fully laminar flow. The least squares approach was adopted to solve the gradient, and a maximum of 100 iteration steps were carried out until the solution converged.

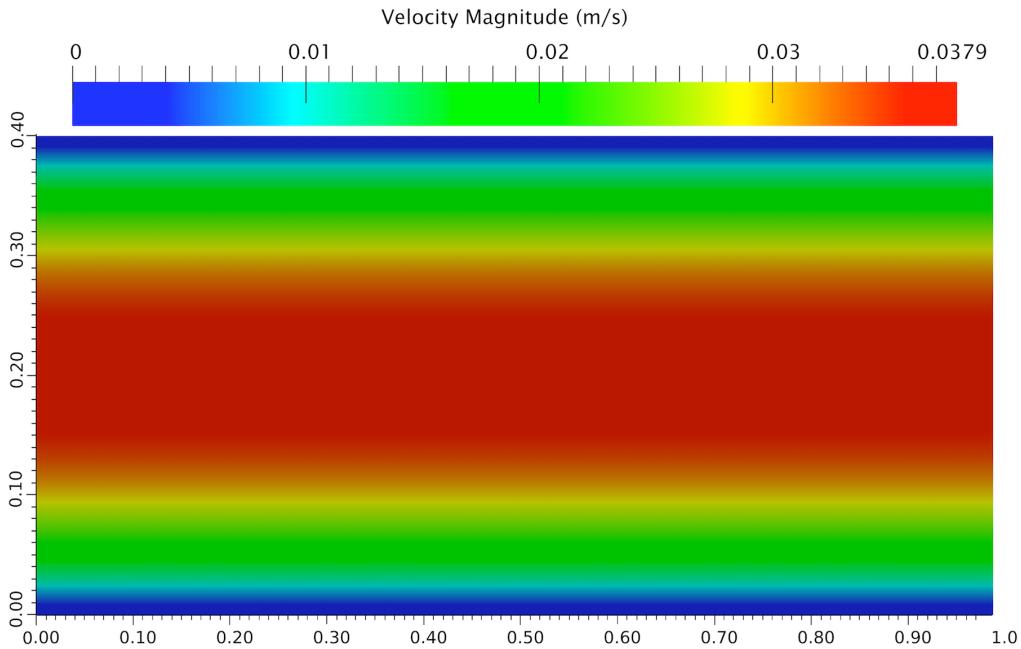


Figure 4.5 Velocity profile: LBM Simulation of a laminar flow through a channel.

The velocity profile obtained from the CFD simulation at cross-section ‘L/4’ is shown in figure 4.7. Figure 4.8 compares the development of computed velocity profiles with the analytical solution. At normalised time  $t = 1$ , the flow approaches steady state. It can be observed that LBM has excellent agreement with CFD and the analytical solution at various stages of flow evolution.

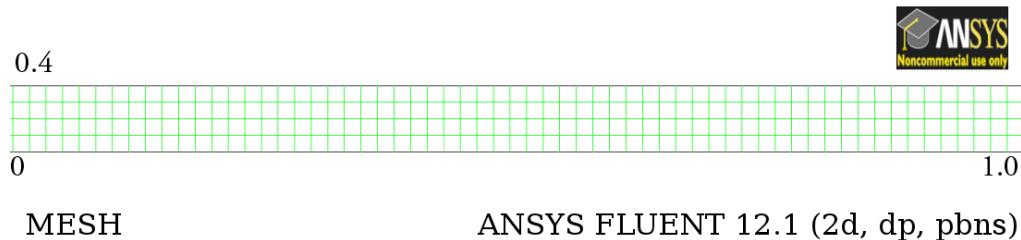


Figure 4.6 Finite Volume mesh used in the CFD analysis of laminar flow through a channel.

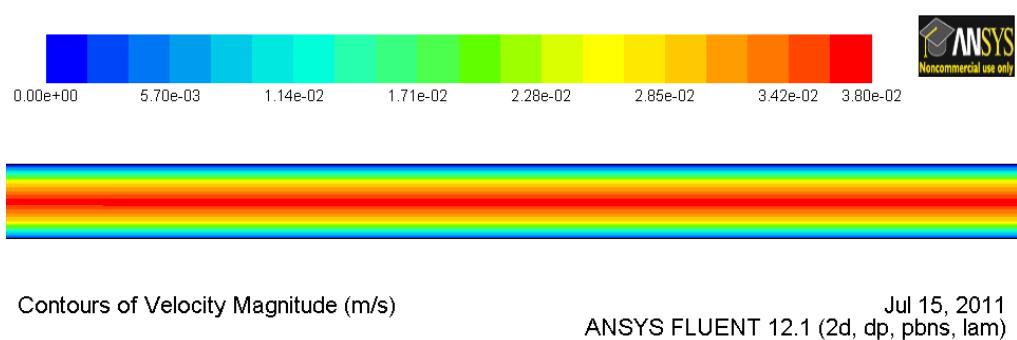


Figure 4.7 Velocity profile: CFD analysis of laminar flow through a channel.

In order to study the capability of the lattice Boltzmann technique to simulate fluid–solid interaction, LB simulation of a fluid flow around a rectangular obstacle is compared with the CFD technique. A solid wall of height ‘H/2’ is placed at length ‘L/4’ in the channel. Bounce-back algorithm is employed to model the fluid-wall interaction in LBM. In the CFD model, the control volume is discretised into 10,000 cells. A constant velocity is applied in the inlet and the horizontal velocity profile is recorded. Both, CFD and LBM simulations were performed to study the influence of a solid wall on the fluid flow behaviour.

The horizontal velocity profile obtained after 50,000 LBM iterations is presented in figure 4.9. LBM is able to capture the velocity shedding around the edges of the wall. The velocity profile obtained from the CFD analysis is presented in figure 4.10. The horizontal velocity profile at ‘L/4’ at  $t = 1$  is shown in figure 4.11. The maximum horizontal velocity from the CFD analysis is 0.3% higher in comparison with the LBM simulation. The discrepancy in the horizontal velocity profile (figure 4.11) can be attributed to the relaxation parameter used in the LBM, which is obtained by a trial and error procedure. The velocity profile obtained from the LBM simulation compares qualitatively with the FE analysis performed by Zhong and Olson (1991). Thus, it can be concluded that the lattice Boltzmann method is a suitable form of numerical representation of the Navier-Stokes equation to model fluid – solid interactions.

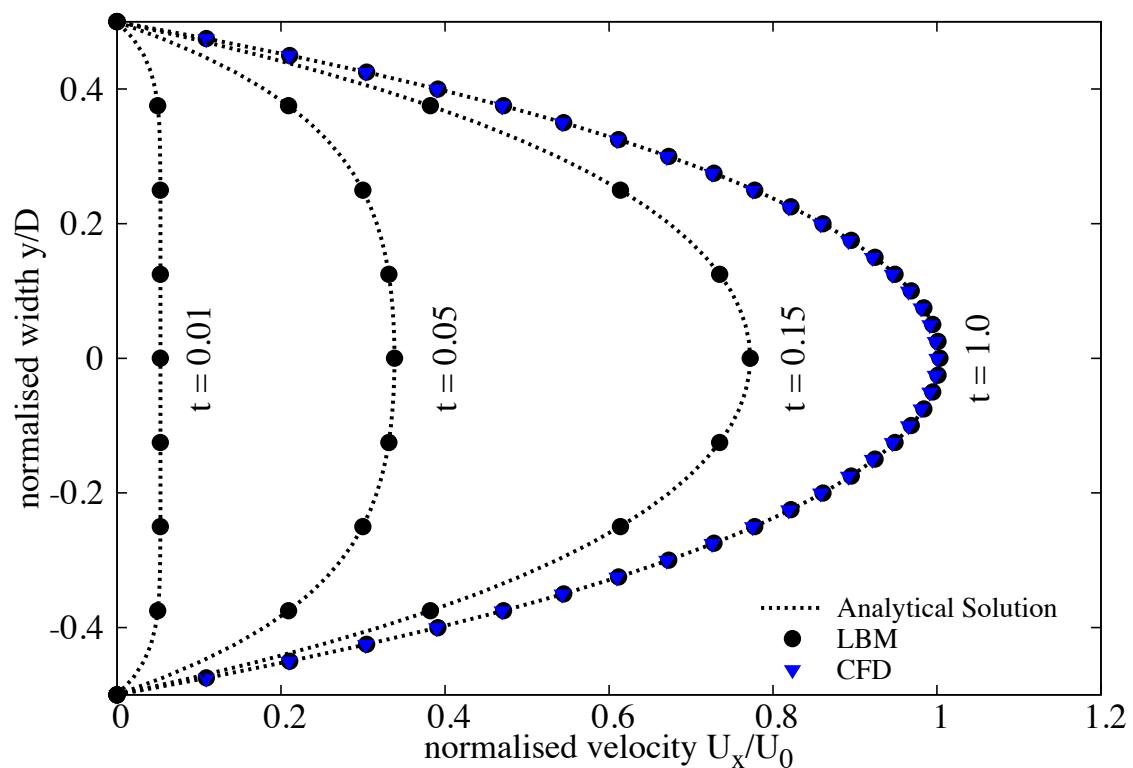


Figure 4.8 Development of the Poiseuille velocity profile in time: comparison between LBM simulation, CFD simulation and the analytical solution. Time is made dimensionless by  $H/U_0$ .

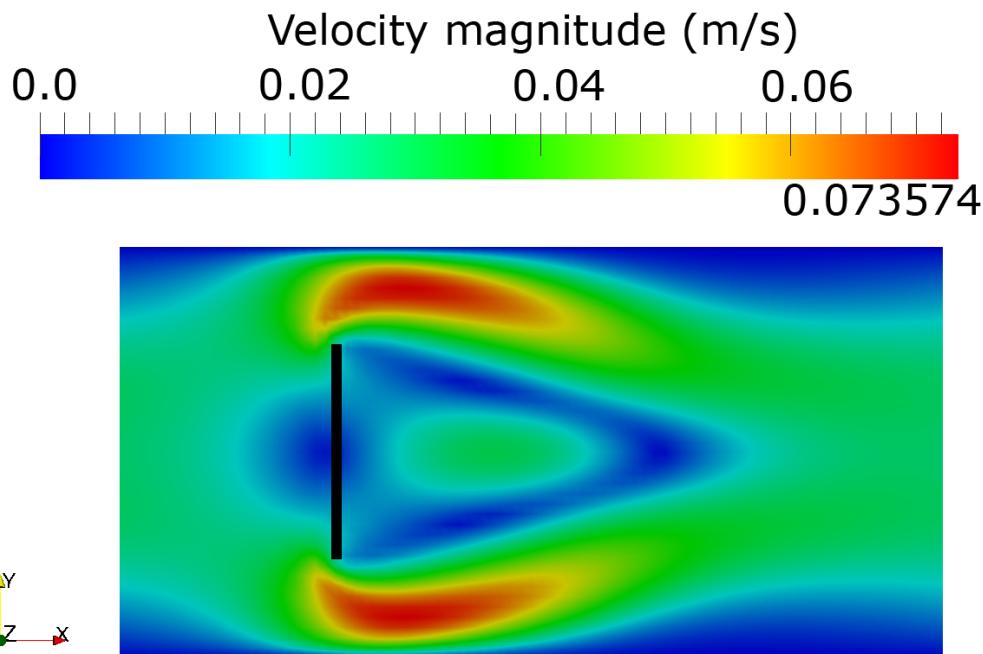


Figure 4.9 LBM simulation of velocity profile for a laminar flow through a pipe with an obstacle at L/4.

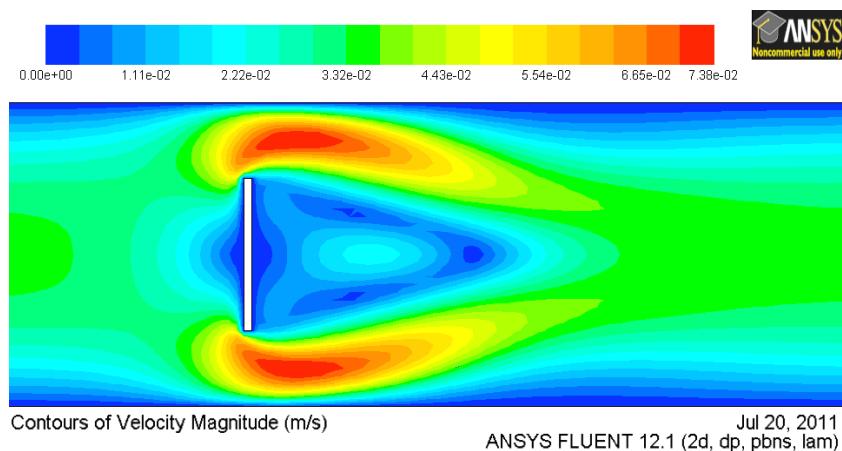


Figure 4.10 CFD simulation of velocity contour for a laminar flow through a pipe with an obstacle at L/4.

## **4.3 Turbulence in lattice Boltzmann method**

- <sup>2</sup> The above formulation of lattice Boltzmann has been successfully applied to many fluid flow problems, however it is restricted to flows with low Reynolds number. Modelling fluids with low viscosity like water and air remains a challenge, necessitating very small values of  $h$ , and/or  $\tau$  very close to 0.5 (He et al., 1997). The standard lattice Boltzmann can deal with laminar

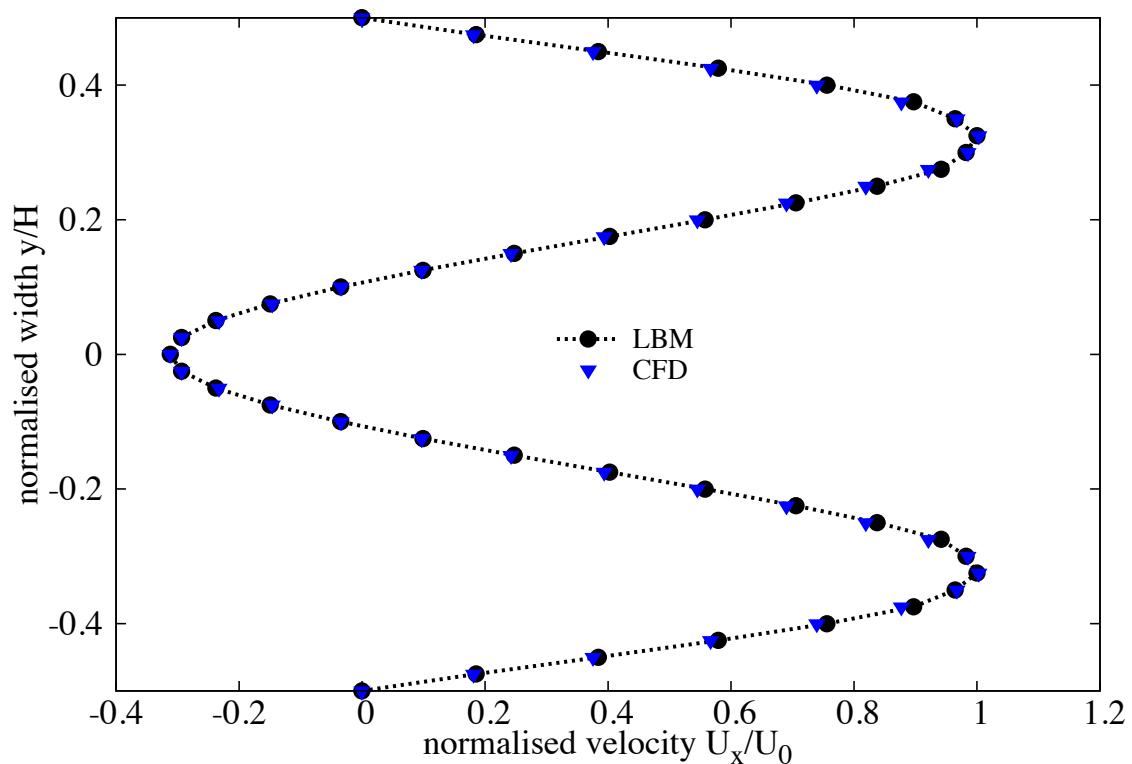


Figure 4.11 LBM and CFD simulation of a flow around an obstacle.

## 4.3 Turbulence in lattice Boltzmann method

119

flows, while practical problems with small kinematic viscosity are often associated with flows having large Reynolds numbers, i.e. flows which are unsteady or turbulent in nature. The turbulent flows are characterised by the occurrence of eddies with multiple scales in space, time and energy.

The large eddy simulation (LES) is the most widely adopted approach to solve turbulent flow problems. It directly solves the large scale eddies, which carry the predominant portion of the energy, and the smaller eddies are modelled using a sub-grid approach. The separation of scales is achieved by filtering of the Navier-Stokes equations, from which the resolved scales are directly obtained. The unresolved scales are modelled by a one-parameter Smagorinski sub-grid methodology, which assumes that the Reynolds stress tensor is dependent only on the local strain rate (Smagorinsky, 1963). It involves parametrising the turbulent energy dissipation in the flows, where the larger eddies extract energy from the mean flow and ultimately transfer some of it to the smaller eddies which, in turn, pass the energy to even smaller eddies, and so on up to the smallest scales. At the smallest scale, the eddies convert the kinetic energy into the internal energy of the fluid. At this scale, the viscous friction dominates the flow (Frisch and Kolmogorov, 1995).

In the Smargonisky model, the turbulent viscosity  $\nu$  is related to the strain rate  $S_{ij}$  and a filtered length scale ‘h’ as follows

$$S_{ij} = \frac{1}{2}(\partial_i u_j + \partial_j u_i) \quad (4.37)$$

$$\nu_t = (S_c h)^2 \bar{S} \quad (4.38)$$

$$\bar{S} = \sqrt{\sum_{i,j} \tilde{S}_{i,j} \tilde{S}_{i,j}}, \quad (4.39)$$

where  $S_c$  is the Smagorinski constant, which is close to 0.03 (Yu et al., 2005). The effect of the unresolved scale motion is taken into account by introducing an effective collision relaxation time scale  $\tau_t$ , so that the total relaxation time  $\tau_*$  is written as

$$\tau_* = \tau + \tau_t, \quad (4.40)$$

where  $\tau$  and  $\tau_t$  are respectively the standard relaxation times corresponding to the true fluid viscosity  $\nu$  and the turbulence viscosity  $\nu_t$ , defined by a sub-grid turbulence model. The new

2 viscosity  $\nu_*$  corresponding to  $\tau_*$  is defined as

$$3 \quad \nu_* = \nu + \nu_t \\ 4 \quad = \frac{1}{3}(\tau_* - \frac{1}{2})C^2\Delta t = \frac{1}{3}(\tau + \tau_t - \frac{1}{2})C^2\Delta t \quad (4.41)$$

$$5 \quad \nu_t = \frac{1}{3}\tau_t C^2\Delta t. \quad (4.42) \\ 6$$

7 The Smagorinski model is easy to implement and the lattice Boltzmann formulation remains  
 8 unchanged, except for the use of a new turbulence-related viscosity  $\tau_*$ . The component  $s_1$  of  
 9 the collision matrix becomes  $s_1 = \frac{1}{\tau + \tau_t}$ .

10 The effectiveness of LBM-LES model in simulating unsteady flows is verified by modelling  
 11 the Kármán vortex street. In fluid dynamics, a Kármán vortex street is a repeating pattern of  
 12 vortices caused by unsteady separation of fluid flow around circular obstacles. A vortex street  
 13 will only be observed above a limiting value of Reynolds number of 90. The Reynolds number  
 14 is computed based on the cylinder diameter ‘D’ and the mean flow velocity  $U$  of the parabolic  
 15 inflow profile:

$$16 \quad Re = \frac{UD}{\nu}. \quad (4.43)$$

17 LBM particles are simulated to flow through a 2D rectangular channel with an aspect ratio  
 18 ‘L/H’ of 2.5. A cylinder of diameter ‘d’ = 0.27H is placed at H/2. The pressure gradient at  
 19 the inlet and the outlet is varied to create flows with different mean velocities. Numerical  
 20 simulations of vortex shedding behind a circular obstacle are carried out for three different  
 21 fluid flow regimes (Reynolds number of 55, 75, and 112). The fully developed fluid flows for  
 22 different Reynolds numbers are shown in figure 4.12. It can be observed from figure 4.12 that  
 23 the von Kármán vortex street can only be observed at high a Reynolds number of 112 ( $Re >$   
 24 90), which shows the ability of the LBM turbulence model to capture instabilities in fluid flow.

One important quantity taken into account in the present analysis is the Strouhal number  $St$ ,  
 1 a dimensionless number describing oscillating unsteady flow dynamics. The Strouhal number  
 2 is computed from the cylinder diameter D, the measured frequency of the vortex shedding f,  
 3 and the maximum velocity  $U_{max}$  at the inflow plane:  
 4

$$5 \quad St = \frac{fD}{U_{max}}. \quad (4.44) \\ 6$$

The characteristic frequency  $f$  is determined by a spectral analysis (Fast Fourier Transform -  
 6 FFT) of time series of the fluid pressure. Table 4.2 shows that the Strouhal numbers computed  
 7 from LBM simulations have a very good agreement with FVM results obtained by Breuer et al.  
 8 (2000). This shows the ability of LBM-LES in capturing unsteady flow dynamics.  
 9

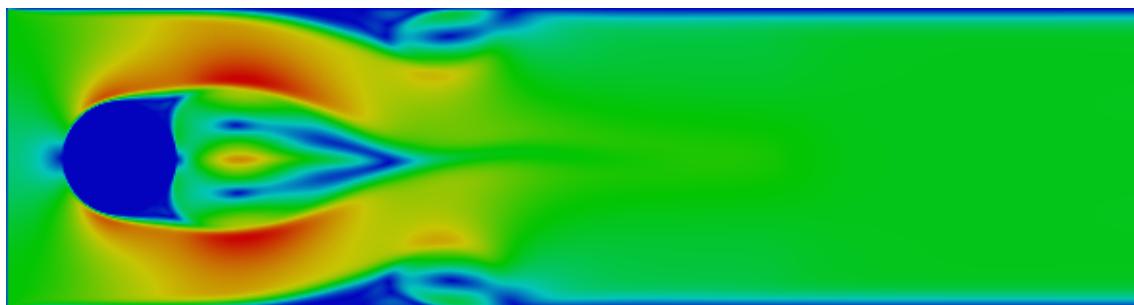
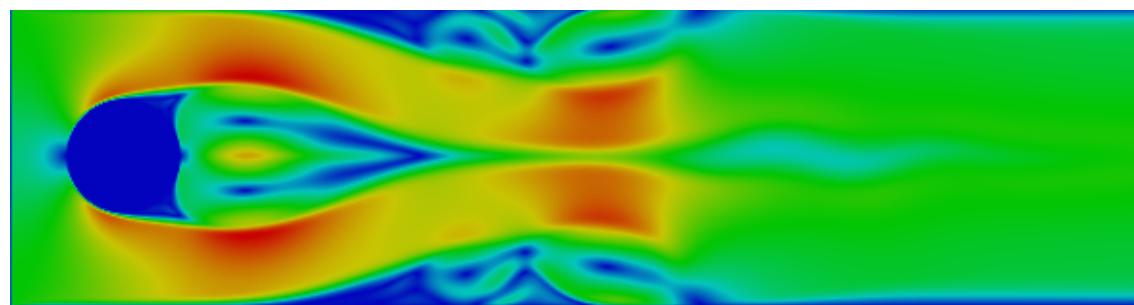
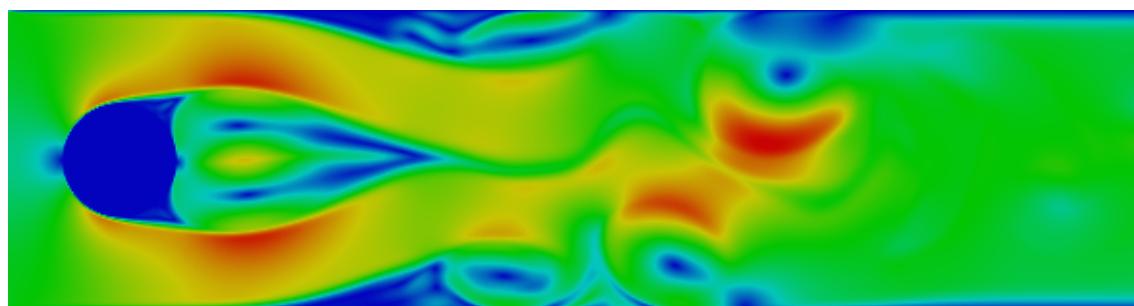
(a)  $\text{Re} = 55$ (b)  $\text{Re} = 75$ (c)  $\text{Re} = 112$ 

Figure 4.12 Kármán vortex street

Table 4.2 Computed Strouhal number for fluid flows with different Reynolds number

Reynolds number	Strouhal number	
	LBM	FVM
55	0.117	0.117
75	0.128	0.129
112	0.141	0.141

\* FVM results are from [Breuer et al. \(2000\)](#)

## 4.4 Coupled LBM and DEM for fluid-grain interactions

Modelling fluid–grain interactions in submarine landslides requires the ability to simulate the interactions at the dynamic fluid – solid boundaries. In principle, the conventional FE and FVM based approaches for solving the Navier-Stokes equations with moving boundaries and/or structural interaction ([Bathe and Zhang, 2004](#)) can be applied to particle fluid interaction problems. The common feature of these approaches is to model the interaction between the fluid and the solid to a high degree of accuracy. However, the main computational challenge is the need to continuously generate new geometrically adapted meshes to circumvent severe mesh distortion, which is computationally very intensive ([Han et al., 2007b](#)).

The lattice Boltzmann approach has the advantage of accommodating large particle sizes and the interaction between the fluid and the moving grains can be modelled through relatively simple fluid - grain interface treatments. Further, employing DEM to account for the grain/grain interaction naturally leads to a combined LB – DEM solution procedure. The Eulerian nature of the lattice Boltzmann formulation, together with the common explicit time step scheme of both LBM and DEM makes this coupling strategy an efficient numerical procedure for the simulation of fluid – grain systems.

LBM – DEM technique is a powerful predictive tool for gaining insights into many fundamental physical phenomena in fluid-solid systems. Such a coupled methodology was first proposed by ([Cook et al., 2004](#)) for simulating fluid-grain systems dominated by fluid-grain and grain-grain interactions. To capture the actual physical behaviour of the fluid-grain system, it is essential to model the boundary condition between the fluid and the grain as a non-slip boundary condition, i.e. the fluid velocity near the grain should be similar to the velocity of the grain boundary. The soil grains in the fluid domain are represented by lattice nodes. The discrete nature of the lattice will result in stepwise representations of the surfaces, which are otherwise circular, this is neither accurate nor smooth, unless sufficiently small lattice spacing is adopted.

## 9 Modified bounce back rule

10 To accommodate the movement of solid particles in the commonly adopted bounce-back rule  
 11 (see section 4.1.3), Ladd (1994) modified the ‘no-slip’ rule for a given boundary link  $i$  to be

$$12 \quad f_i(\mathbf{x}, t + \Delta t) = f_i(\mathbf{x}, t_+) - \alpha_i e_i \cdot v_b \quad (\alpha_i = 6w_i \rho / C_s^2), \quad (4.45)$$

13 where  $f_i(\mathbf{x}, t_+)$  is the post collision distribution at the fluid or solid boundary node  $\mathbf{x}$ , and  $v_b$  is  
 14 the velocity at the nominal boundary point at the middle of the boundary link  $i$

$$15 \quad \mathbf{v}_b = \mathbf{v}_c + \boldsymbol{\omega} \times (\mathbf{x} + e_i \Delta t / 2 - \mathbf{x}_c), \quad (4.46)$$

16 in which  $v_c$  and  $\boldsymbol{\omega}$  are the translational and angular velocities at the mass centre of the solid  
 17 particle, respectively.  $\mathbf{x}_c$  and  $\mathbf{x} + e_i \Delta t / 2$  are the coordinates of the centre and the nominal  
 18 boundary point, respectively. The impact force on the soil grain from the link is defined as

$$19 \quad \mathbf{F}_i = 2[f_i(\mathbf{x}, t_+) - \alpha_i e_i \cdot v_b] / \Delta t. \quad (4.47)$$

20 The corresponding torque  $\mathbf{T}_i$ , produced by the force with respect to the centre of the particle is  
 21 computed as

$$22 \quad \mathbf{T}_i = \mathbf{r}_c \times \mathbf{F}_i (\mathbf{r}_c = \mathbf{x} + \mathbf{e}_i \Delta t / 2 - \mathbf{x}_c). \quad (4.48)$$

23 Then the total hydrodynamic force and torque exerted on the particle can be calculated by  
 24 summing up the forces and torques from all the related boundary links:

$$25 \quad \begin{aligned} \mathbf{F} &= \sum_i \mathbf{F}_i \\ \mathbf{T} &= \sum_i \mathbf{T}_i. \end{aligned} \quad (4.49)$$

26 Ladd and Verberg (2001) described a methodology that minimises the oscillations resulting  
 27 from soil grains crossing lattices at a very high speed. The methodology involves combining  
 28 several extensions for the fluid simulation like the treatment of moving curved boundaries with  
 29 the scheme of Yu et al. (2003) and a fluid/grain force interaction method with the momentum  
 1 exchange method of Ladd and Verberg (2001). The simulation of the moving curved grain  
 2 surfaces results in the intersection of links between two nodes at arbitrary distances (Iglberger  
 3 et al., 2008). These distance values are referred to as delta values:

$$4 \quad \delta = \frac{\text{Distance between fluid node and soil surface}}{\text{Distance between fluid node and soil node}} \in [0, 1]. \quad (4.50)$$

For each pair of a fluid and grain node, a delta value has to be calculated. Delta values of zero are not possible as the nodes on the surface are considered as solid nodes. The algorithm for computation of the  $\delta$  value is presented in [Iglberger et al. \(2008\)](#). Figure 4.13 shows the three possible situations for delta values between 0 and 1. The fluid particles in LBM are always considered to be moving at the rate of one lattice per time step ( $\delta\mathbf{x}/\delta t$ ), for delta values smaller than 0.5. For  $\delta$  values larger than 0.5, the fluid particles would come to rest at an intermediate node  $\mathbf{x}_i$ . In order to calculate the reflected distribution function in node  $\mathbf{x}_f$ , an interpolation scheme has to be applied. The linear interpolation scheme of [Yu et al. \(2003\)](#) is used in the present study, which uses a single equation, irrespective of the value of  $\delta$  being smaller or larger than 0.5, to the reflected distribution function that is computed as

$$\begin{aligned} f_{\bar{\alpha}}(\mathbf{x}_f, t + \delta t) = & \frac{1}{1 + \delta} \cdot [(1 - \delta) \cdot f_{\alpha}(\mathbf{x}_f, t + \delta t) + \delta \cdot f_{\alpha}(\mathbf{x}_b, t + \delta t) \\ & + \delta \cdot f_{\bar{\alpha}}(\mathbf{x}_{f2}, t + \delta t) - 2w_a\rho_w \frac{3}{c^2} e_a \cdot \mathbf{u}_w], \end{aligned} \quad (4.51)$$

where  $w_{\alpha}$  is the weighting factor,  $\rho_w$  is the fluid density in node  $\mathbf{x}_f$ , and  $\mathbf{u}_w$  is the velocity at the bounce-back wall. In order to couple the fluid-grain interaction, the LBM approach is extended by adopting a force integration scheme, to calculate the fluid force acting on the grain surface, and the momentum exchanged method described earlier. The physical force acting on grain agglomerates is calculated as the sum over all fluid/grain node pairs, resulting in

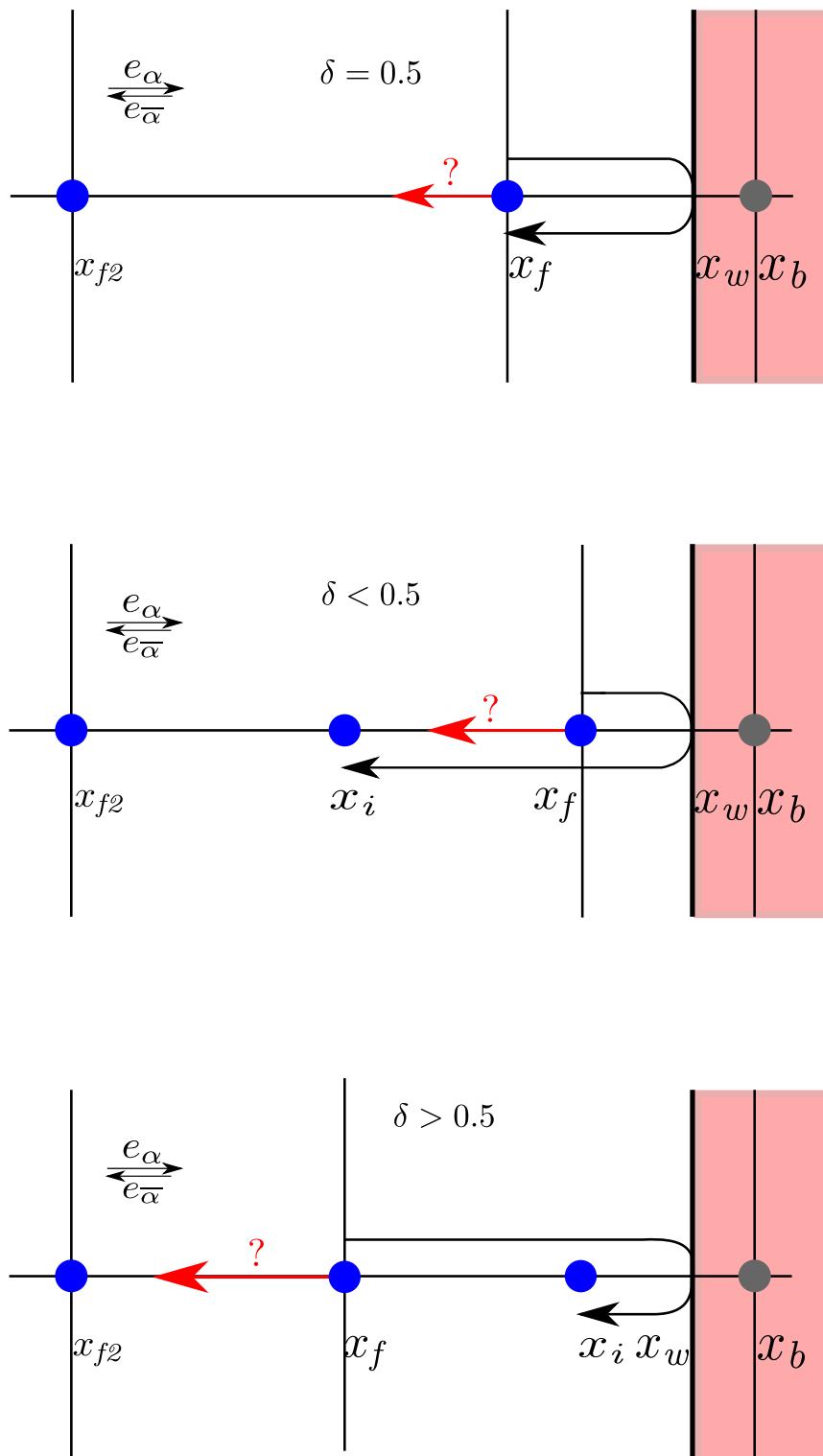
$$F = \sum_{\mathbf{x}_b} \sum_{\alpha=1}^{19} \mathbf{e}_{\alpha} [f_{\alpha}(\mathbf{x}_b, t) + f_{\bar{\alpha}}(\mathbf{x}_f, t)] \delta\mathbf{x}/\delta t. \quad (4.52)$$

After the force calculations, the coupled rigid body physics can be simulated in order to move the grains / grain-agglomerates according to the applied forces. The total hydrodynamic forces and torque exerted on a grain can be computed as ([Cook et al., 2004](#); [Noble and Torczynski, 1998](#))

$$\mathbf{F}_f = Ch \left[ \sum_n \left( \beta_n \sum_i f_i^m e_i \right) \right] \quad (4.53)$$

$$\mathbf{T}_f = Ch \left[ \sum_n (\mathbf{x}_n - \mathbf{x}_c) \times \left( \beta_n \sum_i f_i^m e_i \right) \right]. \quad (4.54)$$

The summation is over all lattice nodes covered by the soil grain, and  $\mathbf{x}_n$  represents the coordinate of the lattice node  $n$ .

Figure 4.13 Bounce back boundaries for different values of  $\delta$

When grains are not in direct contact among themselves, but are driven by the fluid flow and body force, i.e. gravity, their motion can be determined by Newton's equation of motion

$$m\mathbf{a} = \mathbf{F}_f + m\mathbf{g} \quad (4.55) \quad 5$$

$$J\ddot{\theta} = \mathbf{T}_f, \quad (4.56) \quad 6$$

5

6

7

8

9

where  $m$  and  $J$  are respectively the mass and the moment of inertia of a grain,  $\ddot{\theta}$  is the angular acceleration,  $\mathbf{g}$  is the gravitational acceleration,  $\mathbf{F}_f$  and  $\mathbf{T}_f$  are respectively the hydrodynamic forces and torque. The equation can be solved numerically by an explicit numerical integration, such as the central difference scheme.

10

11

12

13

The interaction between the soil grains, and the soil grains with the walls are modelled using the DEM technique. To solve the coupled DEM–LBM formulation, the hydrodynamic force exerted on soil grains and the static buoyancy force are considered by reducing the gravitational acceleration to  $(1 - \rho/\rho_s)\mathbf{g}$ , where  $\rho_s$  is the density of the grains. When taking into account all forces acting on an element, the dynamic equations of DEM can be expressed as

$$m\mathbf{a} + c\mathbf{v} = \mathbf{F}_c + \mathbf{F}_f + m\mathbf{g}, \quad (4.57) \quad 14$$

14

15

16

17

18

where  $\mathbf{F}_c$  denotes the total contact forces from other elements and/or the walls, and  $c$  is a damping coefficient. The term  $c\mathbf{v}$  represents a viscous force that accounts for the effect of all possible dissipation forces in the system including energy lost during the collision between grains. Considering a linear contact model

$$\mathbf{F}_c = k_n\delta, \quad (4.58) \quad 20$$

20

21

22

23

where  $k_n$  is the normal stiffness and  $\delta$  is the overlap, the critical time step associated with the explicit integration is determined as (He et al., 1997)

$$\Delta t_{cr} = 2(\sqrt{1 + \xi^2} - \xi)/\omega, \quad (4.59) \quad 25$$

25

26

where  $\omega = \sqrt{k_n/m}$  is the local contact natural frequency and  $\xi = c/2m\omega$  is the critical damping ratio. The actual time step used for the integration of the Discrete Element equations is

$$\Delta t_D = \lambda \Delta t_{cr}. \quad (4.60) \quad 28$$

28

29

The time step factor  $\lambda$  is chosen to be around 0.1 to ensure both stability and accuracy (He et al., 1997).

30

31

When combining the Discrete Element modelling of the grain interactions with the LB formulation, an issue arises. There are now two time steps:  $\Delta t$  for the fluid flow and  $\Delta t_D$  for the particles. Since  $\Delta t_D$  is normally smaller than  $\Delta t$ ,  $\Delta t_D$  is slightly reduced to a new value  $\Delta t_s$  so that  $\Delta t$  and  $\Delta t_s$  have an integer ratio  $n_s$

$$\Delta t_s = \frac{\Delta t}{n_s} \quad (n_s = [\Delta t / \Delta t_D] + 1). \quad (4.61)$$

This results in a sub-cycling time integration for the Discrete Element part. At every step of the fluid computation,  $n_s$  sub-steps of integration are performed for the Discrete Element Method (4.57) using the time step  $\Delta t_s$ . The hydrodynamic force  $\mathbf{F}_f$  is unchanged during the sub-cycling.

#### 4.4.1 Draft, kiss and tumbling: Sedimentation of two grains

In multiphase flows, fundamental mechanisms of fluid – grain and grain – grain interactions are very important for accurately predicting the flow behaviours. The sedimentation of two circular grains in a viscous fluid serves as the simplest problem to study these two types of interactions, and many experimental and numerical studies have been carried out to investigate this behaviour (Komiwas et al., 2005; Wang et al., 2014). Fortes (1987) observed experimentally that in the sedimentation of two grains under gravity in a Newtonian fluid, the two grains would undergo the draft, kiss and tumbling (DKT) phenomenon.

The *draft*: grain 2 is first placed within the hydrodynamic drag above grain 1. As the hydrodynamic drag of grain 1 is a depression zone, grain 2 is attracted inside. The *kiss*: grain 2 increases its vertical velocity until it touches grain 1. The horizontal velocity of grain 1 increases and its vertical velocity decreases below that of grain 2. *Tumbling*: grain 2 having the same horizontal velocity and higher vertical velocity than grain 1, overtakes grain 1.

LBM-DEM simulation of two grains under gravity in a viscous Newtonian fluid reproduces the draft, kiss and tumble effect (see figure 4.14). They are in agreement with the experimental description of the DKT effect. For better understanding of the DKT effect, the time history of three distances between the grains (normalised to the diameter of the grain D) are tracked i.e., the difference in the transverse coordinates  $\delta_x/D$  and longitudinal coordinates  $\delta_y/D$  of the two grain centres, and the gap between the two surfaces  $\delta = \sqrt{\delta_x^2 + \delta_y^2} - 1$  (see figure 4.15c).

As shown in figure 4.14, grain 1 trails grain 2. As grain 2 approaches the depression zone, corresponding to negative fluid pressure behind grain 1, the velocity of the trailing grain increases as the grains approach closer, this is in agreement with the experimental description of the draft. Grain 2 increases its vertical velocity more than grain 1 until it touches grain 1. The kiss happens at a normalised time  $(t / \sqrt{(D/g)}) = 25$ . At this stage, the gap  $\delta$  between the

<sup>36</sup> grains is zero, but the actual gap is about one lattice spacing for the LBM collision model. After  
this time, the vertical velocity of grain 1 decreases and its horizontal velocity increases as the  
grains tumble. At this stage, the grains still remain in contact, i.e., the gap remains unchanged  
 $\delta = 0$ . Subsequently, the two grains separate and move away from each other. Figure 4.15b  
shows that the terminal velocities of the two grains are in good agreement with the terminal  
velocity of a single grain found by an independent simulation and calculated using the empirical  
Schiller and Nauman formula ([Komiwas et al., 2005](#)).  
<sup>1</sup>  
<sup>2</sup>  
<sup>3</sup>  
<sup>4</sup>  
<sup>5</sup>  
<sup>6</sup>

## 4.5 GP-GPU Implementation

Graphics Processing Unit (GPU) is a massively multi-threaded architecture that is widely used for graphical and now non-graphical computations. Today's GPUs are general purpose processors with support for an accessible programming interface. The main advantage of GPUs is their ability to perform significantly more floating point operations (FLOPs) per unit time than a CPU. General Purpose computations on GPUs (GPGPUs) often achieve speedups of orders of magnitude in comparison with optimised CPU implementations.  
<sup>7</sup>  
<sup>8</sup>  
<sup>9</sup>  
<sup>10</sup>  
<sup>11</sup>  
<sup>12</sup>  
<sup>13</sup>

A GPU consists of several *Streaming Multiprocessors* (SMs). Each SM contains 32 CUDA processors. Each CUDA processor has a fully pipelined integer arithmetic logic unit (ALU)  
<sup>1</sup> and a floating point unit (FPU). The FPU complies with the IEEE 754-2008 industry standard  
<sup>2</sup> for floating-point arithmetic, capable of double precision computations. The SM schedules  
<sup>3</sup> work in groups of 32 threads called warps. Each SM features two warp schedulers and two  
<sup>4</sup> instruction dispatch units, allowing two warps to be issued and executed concurrently. Each  
<sup>5</sup> thread has access to both L1 and L2 caches, which improves the performance for programs  
<sup>6</sup> with random memory access.  
<sup>14</sup>  
<sup>15</sup>

The occupancy rate of the SPs, i.e. the ratio between the number of threads run and the maximum number of executable threads, is an important aspect to take into consideration for the optimisation of a CUDA kernel. Even though a block may only be run on a single SM, it is possible to execute several blocks concurrently on the same SM. Hence, tuning the execution grid layout allows one to increase the occupancy rate. Nevertheless, reaching the maximum occupancy is usually not possible, as the threads executed in parallel on one SM have to share the available registers ([Obrecht and Kuznik, 2011](#)).  
<sup>7</sup>  
<sup>8</sup>  
<sup>9</sup>  
<sup>10</sup>  
<sup>11</sup>  
<sup>12</sup>  
<sup>13</sup>

Many-core processors are promising platforms for intrinsically parallel algorithms such as the lattice Boltzmann method. Since the global memory for GPU devices shows high latency and LBM is data intensive, the memory access pattern is an important issue for achieving good performances. Whenever possible, global memory loads and stores should be coalescent and aligned, but the propagation phase in LBM can lead to frequent misaligned memory accesses.  
<sup>14</sup>  
<sup>15</sup>  
<sup>16</sup>  
<sup>17</sup>  
<sup>18</sup>

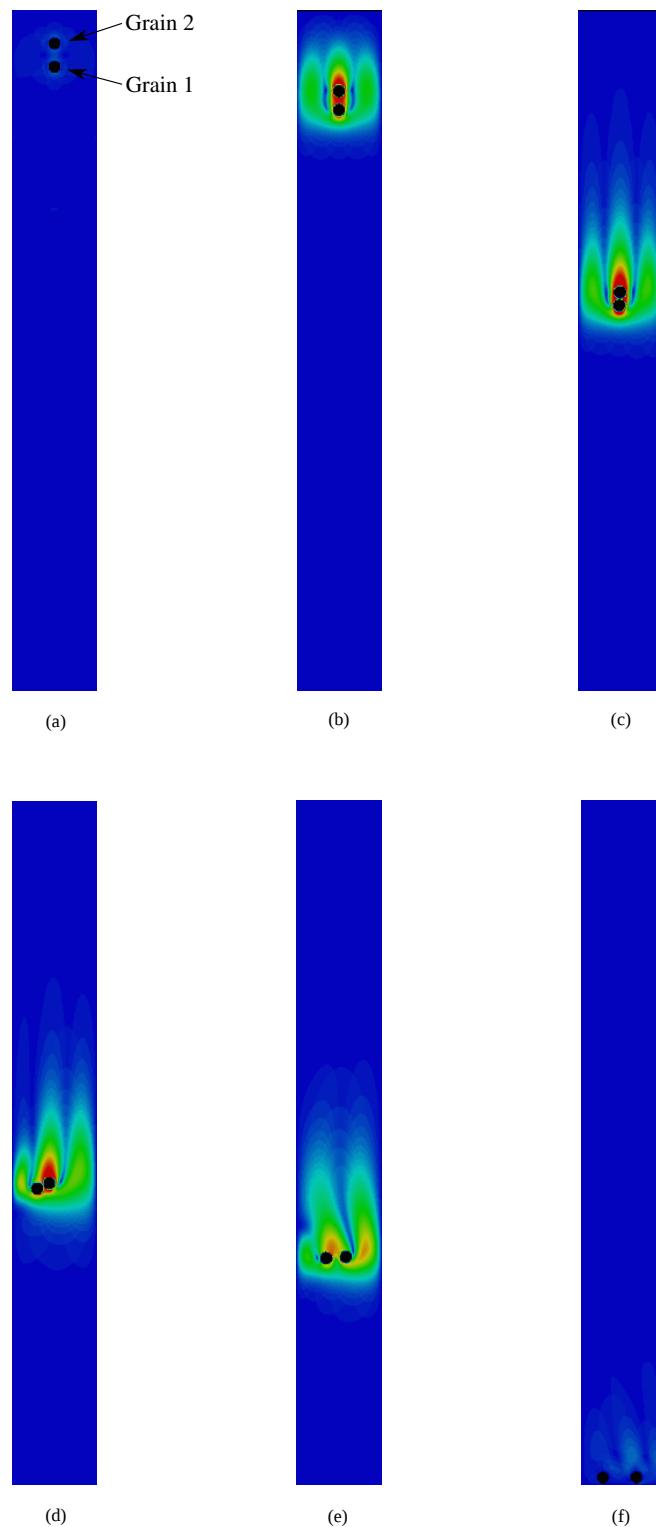


Figure 4.14 Time series of draft, kiss and tumble of two grains during sedimentation in a viscous fluid.

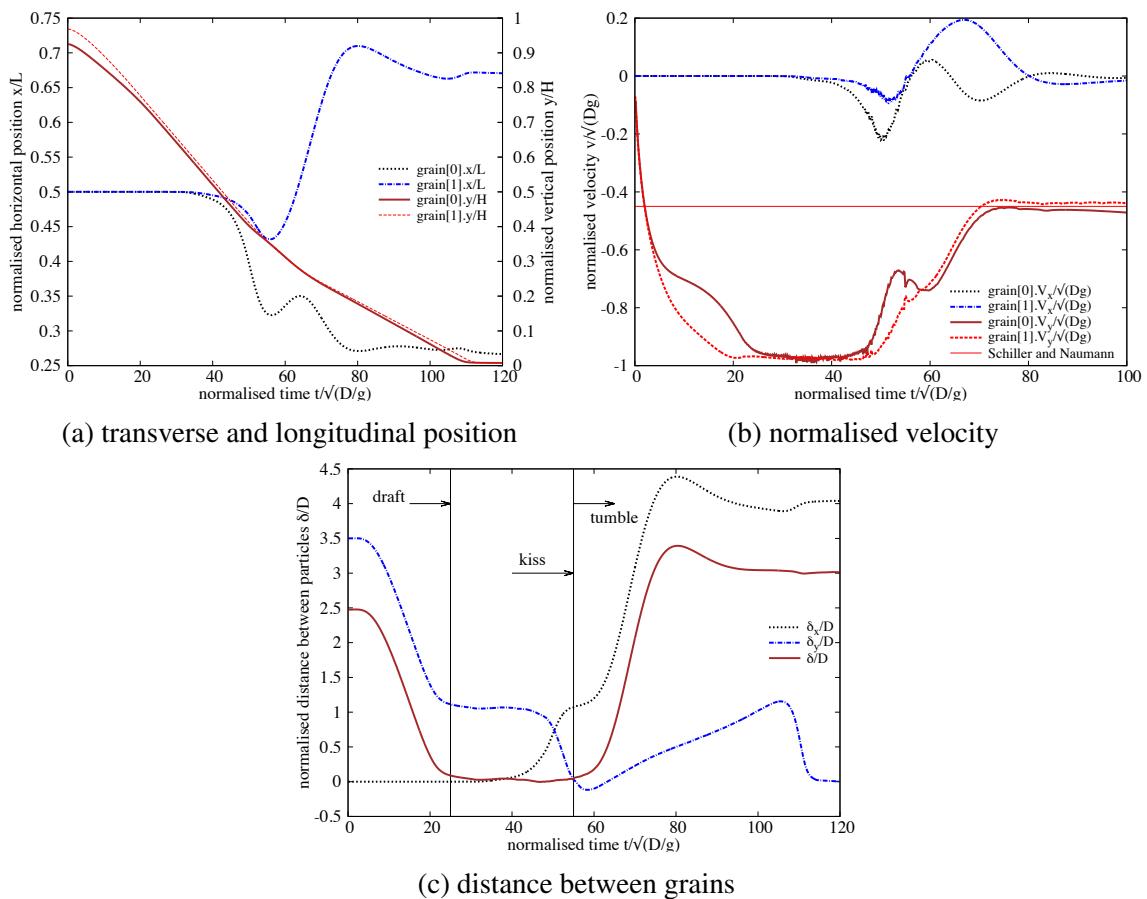


Figure 4.15 Time history of two circular grains during sedimentation.

19 Also, the data transfer between the host and the device is very expensive. In the present study,  
20 the LBM implementation follows carefully chosen data transfer schemes in global memory.

21 There are three ways to accelerate GPGPU applications: (a) Using ‘drop-in’ libraries,  
22 (b) using directives by exposing parallelism, and (c) using dedicated GPGPU programming  
23 languages. OpenACC (Open Accelerators) is an open GPU directives programming standard  
24 for parallel computing on heterogeneous CPU/GPU systems. Unlike conventional GPU pro-  
25 gramming languages, such as CUDA, OpenACC uses directives to specify parallel regions in  
26 the code and performance tuning works on exposing parallelism. OpenACC targets a host-  
27 directed execution model where the sequential code runs on a conventional processor and  
28 computationally intensive parallel pieces of code (kernels) run on an accelerator such as a GPU  
29 (see figure 4.16).

30 The original GPGPU LBM – DEM code was implemented in C using OpenACC API v1.0,  
31 which was released in November 2011. The current implementation in C++ uses OpenACC API  
32 v2.0a ([OpenACC-Members, 2013](#)) and has two compute constructs, the kernels construct and  
33 the parallel construct. LBM – DEM implementation predominantly uses the OpenACC gang and  
34 vector parallelism. The LBM – DEM code runs sequential and computationally less intensive  
functions on the CPU, OpenMP multi-threading is used when possible. Computationally  
intensive functions are converted to a target accelerator specific GPU parallel code. Schematics  
of a heterogeneous CPU/GPU system is shown in figure 4.16.

OpenACC offers kernel and parallel constructs to parallelise algorithms on CUDA kernels.  
The loop nests in a kernel construct are converted by the compiler into parallel kernels that  
run efficiently on a GPU. There are three steps to this process. The first is to identify the  
loops that can be executed in parallel. The second is to map that abstract loop parallelism  
onto a concrete hardware parallelism. In OpenACC terms, gang parallelism maps to grid-level  
parallelism (equivalent to a CUDA blockIdx), and vector parallelism maps to thread-level  
parallelism (equivalent to a CUDA threadIdx). The compiler normally maps a single loop  
across multiple levels of parallelism using strip-mining. Finally, in step three the compiler  
generates and optimizes the actual code to implement the selected parallelism mapping.

An OpenACC parallel construct creates a number of parallel threads that immediately begin  
executing the body of the parallel construct redundantly. When a thread reaches a work-sharing  
loop, that thread will execute some subset of the loop iterations, depending on the scheduling  
policy as specified by the program or at the runtime. The code generation and optimization  
for a parallel construct is essentially the same as for the kernel construct. A key difference  
is that unlike a kernel construct, the entire parallel construct becomes a single target parallel  
operation, aka a single CUDA kernel. Both constructs allow for automatic vectorization within  
the loops ([Wolfe, 2012](#)).

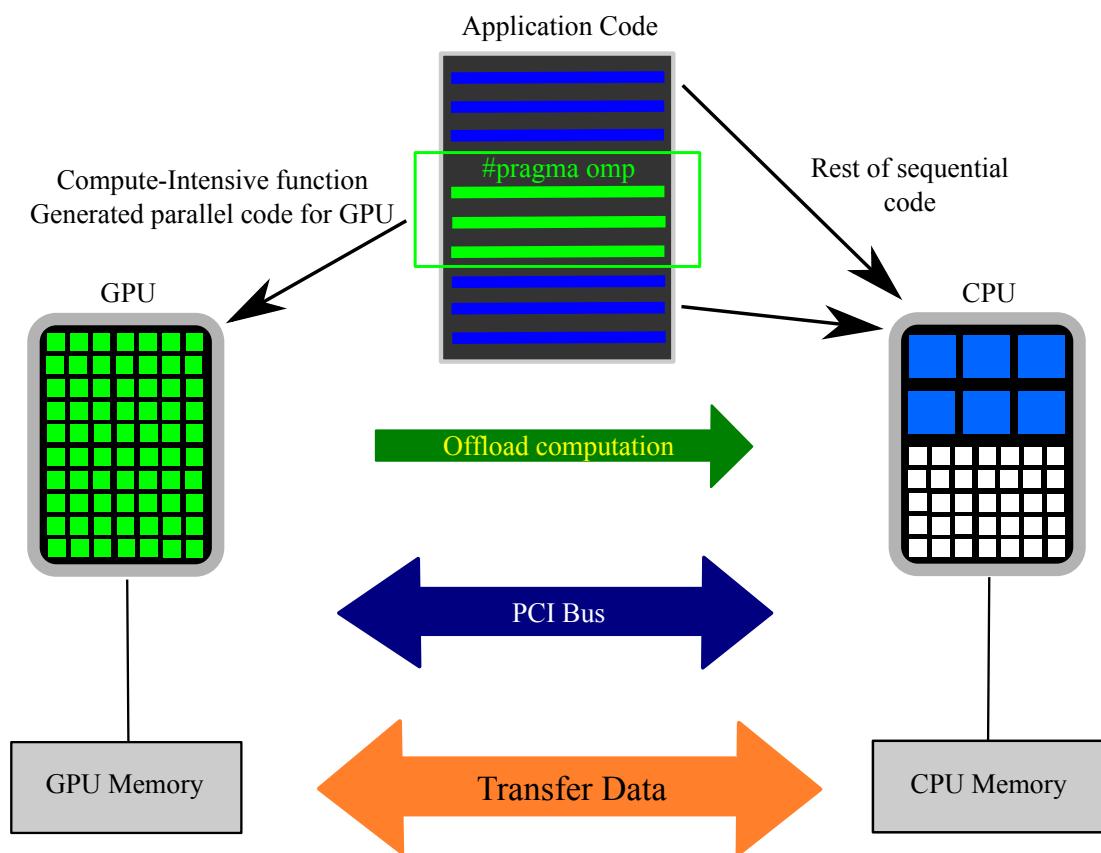


Figure 4.16 Schematics of a heterogeneous CPU/GPU system.

An excerpt from the LBM-DEM code showing the OpenACC GPU implementation of the hydrodynamic force computation is presented in Listing 4.1. The kernels loop construct tells the compiler to map the body of the following loop into an accelerator kernel. The GPU implementation uses a two-dimensional grid splitting the iterations across both the vector and gang modes. The kernel is mapped to a vector mode mapped (aligned with CUDA threadidx% $x$ ) with a vector length (thread block size) of 128. The kernel is also mapped to gang parallelism, aligned to CUDA blockidx% $x$ , to avoid partition camping by mapping the stride-1 loop to the  $x$  dimension. The compiler strip-mines the loop into chunks of 256 iterations, mapping the 256 iterations of a chunk in vector mode across the threads of a CUDA thread block, and maps the  $n/256$  chunks in gang mode across the thread blocks of the CUDA grid. The consecutive iterations ( $i$  and  $i+1$ ), which refer to contiguous array elements ( $f_{hf}[i]$  and  $f_{hf}[i+1]$ ), are mapped to adjacent CUDA threads in the same thread block, to optimize for coalesced memory accesses.

Listing 4.1 OpenACC GPU implementation of the hydrodynamic force computation.

```

1 //OpenACC Kernels copy data between the host and the device
2 #pragma acc kernels
3 copyout(fhf1 [0: nbgrains ], f hf2 [0: nbgrains ], f hf3 [0: nbgrains ])
1 4 copyin(obst [0:][0:], g [0: nbgrains ], ey [0:], f [0:][0:][0:], ex [0:])
2 5 //Create individual threads for each DEM grain
3 6 #pragma acc parallel for
4 7 for ( i=0; i<nbgrains ;i++) {
5 8     // Reset hydrodynamic forces to zero at the start of time step
6 9     f hf1 [ i]=f hf2 [ i]=f hf3 [ i ]=0. ;
7 10    // Iterate through all lattice nodes
8 11    for ( y=0; y<ly;y++) {
9 12        for ( x=0; x<lx;x++) {
10 13            if (obst [x ][y]==i) {
11 14                // generate code to execute the iterations in parallel with
12 15                // no synchronization
13 16                #pragma acc for independent
14 17                for ( iLB=1; iLB<Q; iLB++) {
15 18                    next_x=x+ex[iLB];
16 19                    next_y=y+ey[iLB];
17 20                    if (iLB<=half) halfq=half ;
18 21                    else halfq= -half;
```

```

19   22     if (obst[next_x][next_y]!=i) {
20   23       fnx=(f[x][y][iLB+halfq]+f[next_x][next_y][iLB])*ex[iLB+halfq];
21   24       fny=(f[x][y][iLB+halfq]+f[next_x][next_y][iLB])*ey[iLB+halfq];
22   25       fhf1[i]=fhf1[i]+fnx;
23   26       fhf2[i]=fhf2[i]+fny;
24   27       fhf3[i]=fhf3[i]-fnx*(y-(g[i].x2-wall_bottom_y)/dx)
28           +fny*(x-(g[i].x1-wall_left_x)/dx);
29   30     }
30   31   }
31   32   }
32   33   }
33   34 }
```

Memory transaction optimisation is more important than computation optimisation. Registers do not give rise to any specific problem apart from their limited amount. Global memory, being the only one accessible by both the CPU and the GPU, is the critical path as it suffers from high latency. However, this latency is mostly hidden by the scheduler which stalls inactive warps until data are available. For data intensive LBM, this aspect is generally the limiting factor ([Obrecht and Kuznik, 2011](#)). To optimise the global memory transactions, the memory access is coalesced and aligned, as explained above. The memory transactions between the host and the target through a PCI bus are kept to a minimum.

A two-dimensional fluid – grain system, which consists of 7.2 million LBM nodes and 2500 DEM grains is used to demonstrate the ability of the GPGPU LBM – DEM code. The wall time required to compute 100 iterations of the given LBM – DEM problem is compared for executions running on a single CPU thread, multi-threaded CPU (using OpenMP) and the GPGPU implementations (see table 4.3). The speed-up of parallel implementations are measured against the single CPU thread execution time. OpenMP parallelised multi-threaded CPU execution running on 12 cores achieved a speed-up of 13.5x in comparison to a serial implementation. GPGPU implementation using OpenACC delivered an impressive 126x speed-up in comparison to a single thread CPU execution and about 10 times quicker than a CPU parallel code. In other words, a simulation that would have ordinarily taken 126 days to compute, could now be finished in a day using a GPU.

Scalability is an important criterion when developing high-performance computing codes. Scalability in GPUs is measured in terms of SM utilisation. It is important to distribute sufficient work to all SMs such that on every cycle the warp scheduler has at least one warp eligible to issue instructions. In general, sufficient warps on each SM should be available to

Table 4.3 GPU vs CPU parallelisation

Execution	Computational Time (s)	Speedup
CPU 1 OpenMP thread	2016	–
CPU 2 OpenMP threads	1035	1.5 x
CPU 4 OpenMP threads	660	3.0 x
CPU 12 OpenMP threads	150	13.5 x
GPU OpenACC	16	126.0 x

# Wall time for 100 iteration for 7.2 Million LBM nodes and 2500 DEM grains.

\* CPU OpenMP threads - 6 core Intel Xeon @ 3.3GHz

† GPU threads - GeForce GTX 580 - 512 CUDA cores

3 hide instruction and memory latency and to provide a variety of instruction types to fill the  
 4 execution pipeline. Figure 4.17 shows the scalability of GP-GPU implementation as the LBM  
 5 domain size is increased from 500,000 to 9 million nodes. With increase in LBM nodes the  
 6 computation time increases linearly with a slope of about 2, which shows that the LBM-DEM  
 7 implementation algorithm scales with the domain size.

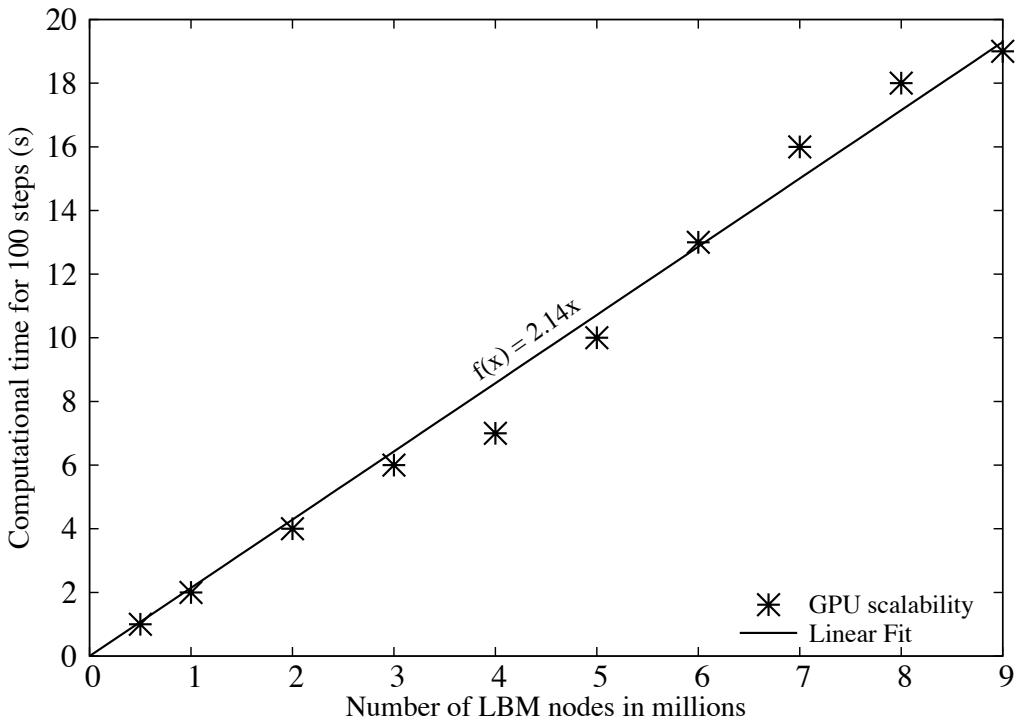


Figure 4.17 GPU scalability with increase in LBM nodes

8 A two-dimensional coupled LBM-DEM technique is developed to understand the local  
 9 rheology of granular flows in fluid. The coupled LBM-DEM technique offers the possibility to

10 capture the intricate microscale effects such as the hydrodynamic instabilities. The Smagorinsky  
11 turbulence model is implement in LBM to capture the unsteady flow dynamics in underwater  
12 granular avalanches. The GPGPU implementation of the coupled LBM – DEM technique  
13 offers the capability to model large scale fluid – grain systems, which are otherwise impossible  
14 to model using conventional computation techniques. Efficient data transfer mechanisms that  
15 achieves coalesced global memory ensures that the GPGPU implementation scales linearly  
16 with the domain size.

<sup>17</sup>

# Chapter 5

<sup>18</sup>

## Underwater granular flows

<sup>19</sup>

### 5.1 Submarine granular flows down incline plane

<sup>20</sup> The flow of dense granular material is a common phenomenon in engineering predictions, such  
<sup>21</sup> as avalanches, landslides, and debris-flow modelling. Despite the huge amount of research  
<sup>22</sup> that has gone into describing the behaviour of granular flows, a constitutive equation that  
<sup>23</sup> describes the overall behaviour of a flowing granular material is still lacking. The initiation and  
<sup>24</sup> propagation of submarine granular flows depend mainly on the slope, density, and quantity of  
1 the material destabilised. Although certain macroscopic models are able to capture the simple  
2 mechanical behaviours, the complex physical mechanisms that occur at the grain scale, such as  
3 hydrodynamic instabilities, the formation of clusters, collapse, and transport, have largely been  
4 ignored ([Topin et al., 2011](#)). The momentum transfer between the discrete and the continuous  
5 phases significantly affects the dynamics of the flow ([Peker and Helvacı, 2007](#)). Grain-scale  
6 description of the granular material enriches the macro-scale variables, which poorly account  
7 for the local rheology of the materials. In order to describe the mechanism of saturated and/or  
8 immersed granular flows, it is important to consider both the dynamics of the solid phase and  
9 the role of the ambient fluid ([Denlinger and Iverson, 2001](#)). In particular, when the solid phase  
10 reaches a high volume fraction, it is important to consider the strong heterogeneity arising from  
11 the contact forces between the grains, the drag interactions which counteract the movement  
12 of the grains, and the hydrodynamic forces that reduce the weight of the solids inducing a  
13 transition from dense compacted to a dense suspended flow ([Meruane et al., 2010](#)). The case of  
14 the collapse in presence of an interstitial fluid has been less studied. In this paper, we study the  
15 submarine granular flows in the inclined configuration. We study the effect of permeability,  
1 density and slope angle on the run-out evolution.

<sup>2</sup> In this study, a 2D poly-disperse system ( $d_{max}/d_{min} = 1.8$ ) of circular discs in fluid was  
<sup>3</sup> used to understand the behaviour of granular flows on inclined planes (see Figure [5.1](#)). The soil

4 column was modelled using 1000 discs of density  $2650 \text{ kg m}^{-3}$  and a contact friction angle of  
 5  $26^\circ$ . The collapse of the column was simulated inside a fluid with a density of  $1000 \text{ kg m}^{-3}$  and  
 6 a kinematic viscosity of  $1 \times 10^{-6} \text{ m}^2 \text{ s}^{-1}$ . The choice of a 2D geometry has the advantage of  
 7 cheaper computational effort than a 3D case, making it feasible to simulate very large systems.  
 8 A granular column of aspect ratio ‘a’ of 0.8 was used. A hydrodynamic radius  $r = 0.9R$  was  
 9 adopted during the LBM computations. Dry analyses were also performed to study the effect  
 10 of hydrodynamic forces on the run-out distance.

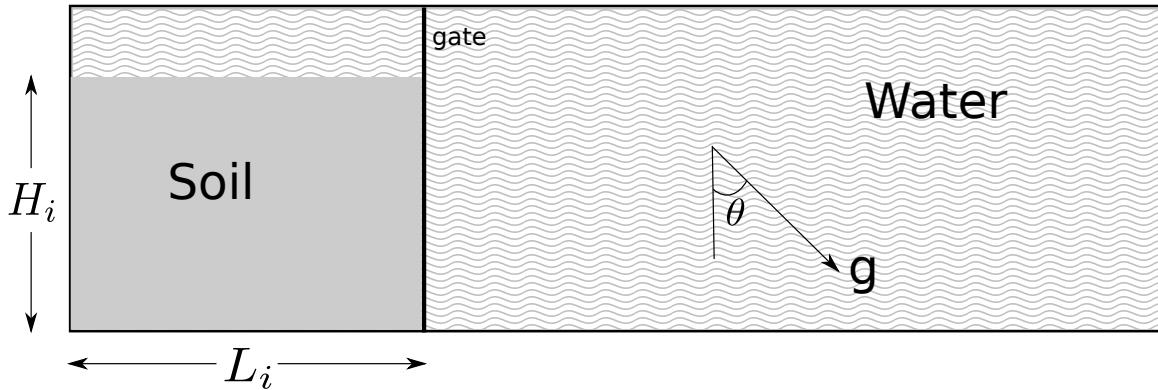


Figure 5.1 Underwater granular collapse set-up

### 11 5.1.1 Effect of initial density

12 The morphology of the granular deposits in fluid is shown to be mainly controlled by the initial  
 13 volume fraction of the granular mass and not by the aspect ratio of the column (Pailha et al.,  
 14 2008; Rondon et al., 2011). In order to understand the influence of the initial packing density  
 15 on the run-out behaviour, a dense sand column (initial packing density,  $\Phi = 83\%$ ) and a loose  
 16 sand column ( $\Phi = 79\%$ ) were used. The granular columns collapse and flow down slopes of  
 17 varying inclinations ( $2.5^\circ$ ,  $5^\circ$  and  $7.5^\circ$ ).

18 The evolution of run-out distances for a dense sand column with time in dry and submerged  
 19 conditions for varying slope inclinations are presented in figure 5.2. The run-out distance  
 20 is longer in submerged condition than the dry condition for a flow on a horizontal surface.  
 21 However, with increase in the slope angle the run-out in the fluid decreases.

22 Dense granular columns in fluid take a longer time to collapse and flow, due to the develop-  
 23 ment of large negative pore-pressure, as the dense granular material dilates during the initial  
 24 phase of the flow. The morphology of dense granular flows down slopes of varying inclinations  
 25 at the critical time ( $t = \tau_c = \sqrt{H/g}$ , when the flow is fully mobilised) are shown in figure 5.4.

26 It can be seen that the viscous drag on the dense column tend to predominate over the  
 27 influence of hydroplaning on the run-out behaviour. This influence can be observed in the

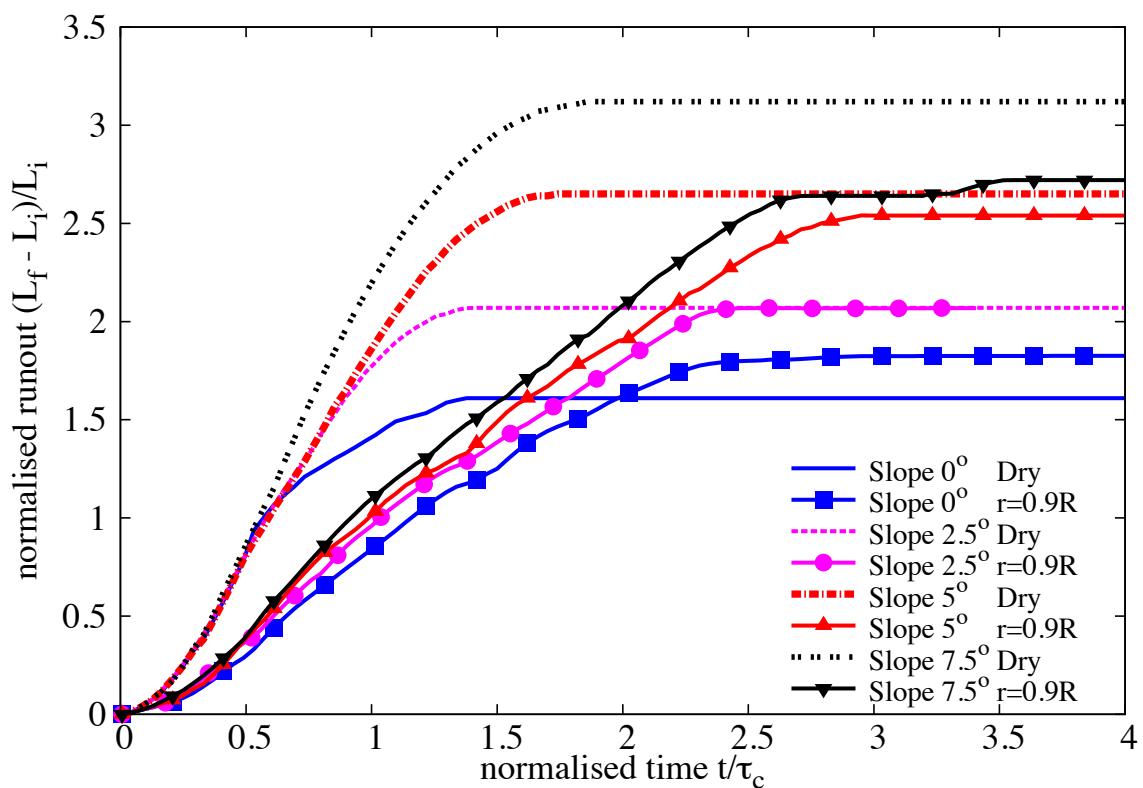


Figure 5.2 Evolution of run-out with time (dense)

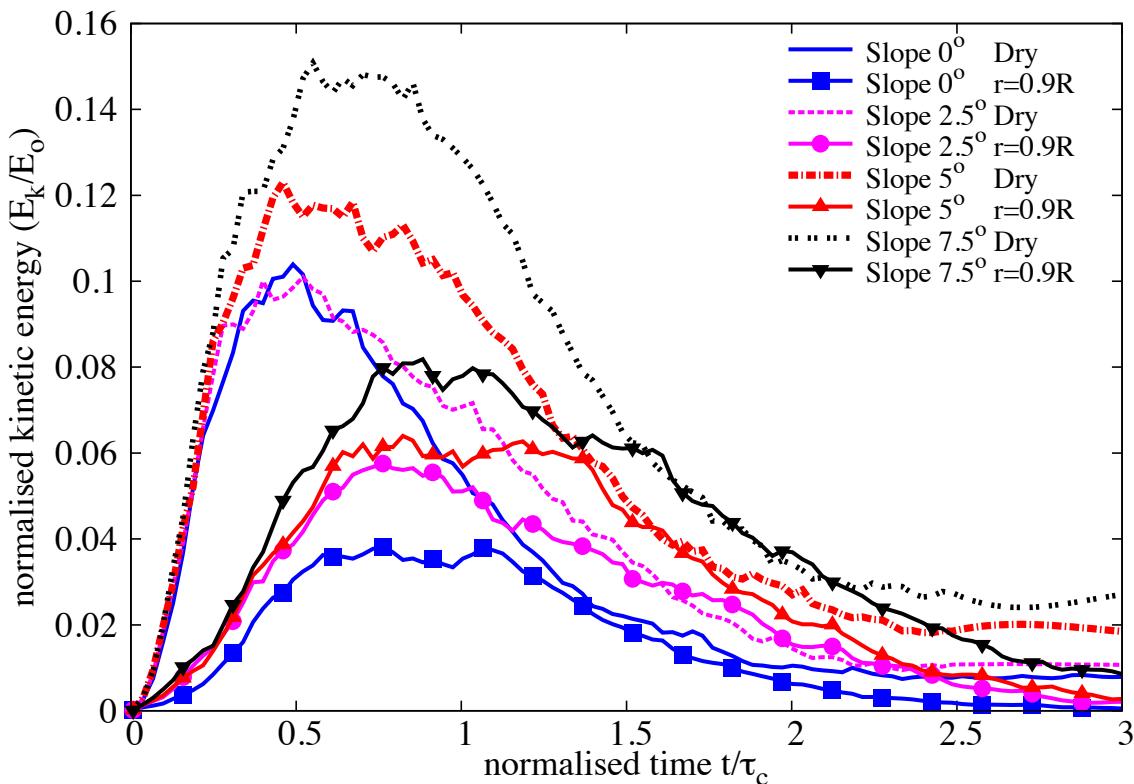
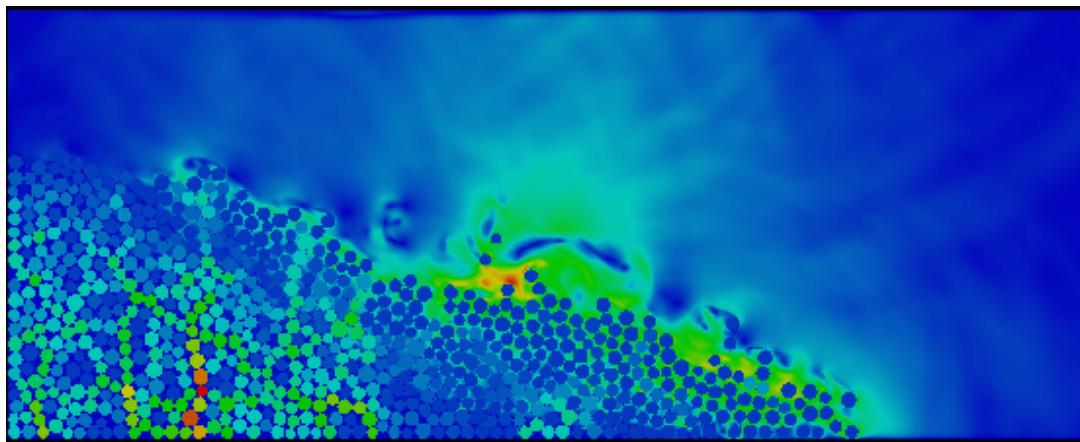


Figure 5.3 Evolution of Kinetic Energy with time (dense case)

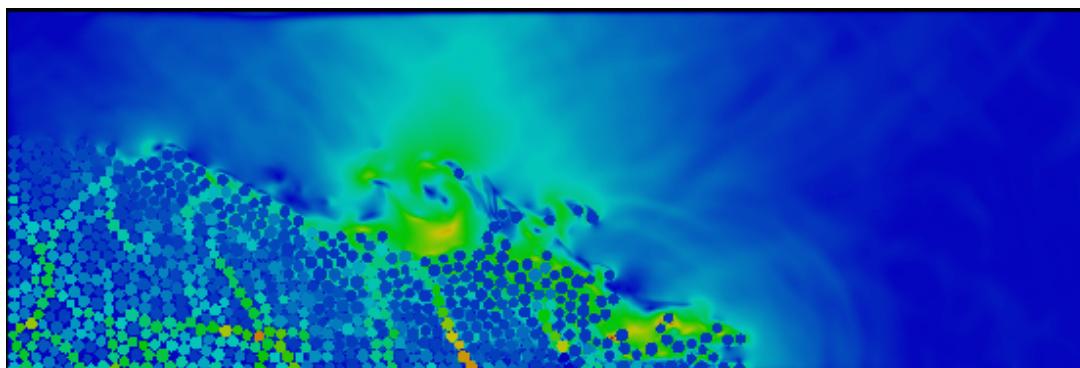
28 smaller peak kinetic energy for granular column in fluid compared to it's dry counterpart  
 29 (see Figure 5.3). With increase in slope angle, the volume of material that dilates increases.  
 30 This results in large negative pore pressures and more viscous drag on the granular material.  
 31 Hence, the difference in the run-out between the dry and the submerged condition, for a dense  
 32 granular assembly, increases with increase in the slope angle.

In contrast to the dense granular columns, the loose granular columns (relative density  $I_D = 30\%$ ) show longer run-out distance in immersed conditions (see Figure 5.5). The run-out distance in fluid increases with increase in the slope angle compared to the dry cases. Loose granular material tends to entrain more water at the base of the flow front, creating a lubricating surface, which causes longer run-out distance (see Figure 5.6). The hydroplaning effect causes an increase in the velocity the loose condition in comparison with the dense condition (see Figure 5.7).

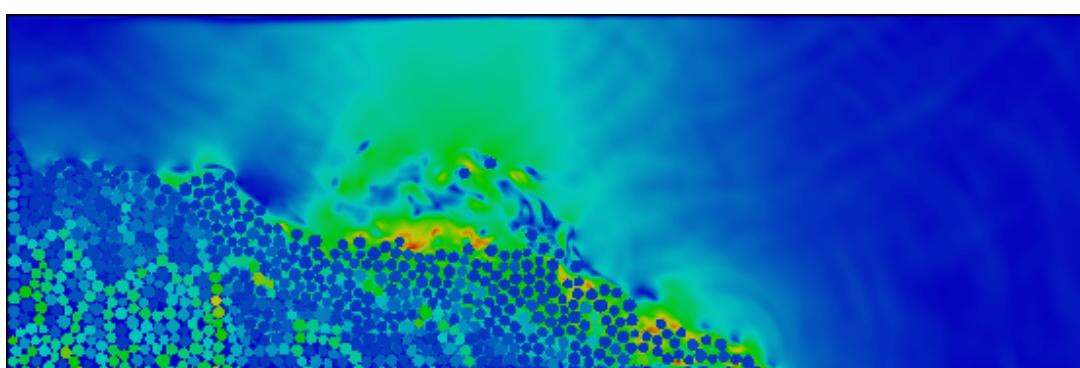
The evolution of packing density (see Figure 5.8) shows that dense and the loose conditions reach similar packing density. This indicates that the dense granular column dilates more and is susceptible to higher viscous drag forces. Whereas in the loose condition, a positive



(a) Slope 2.5



(b) Slope 5.0



(c) Slope 7.5

Figure 5.4 Flow morphology at critical time for different slope angles (dense)

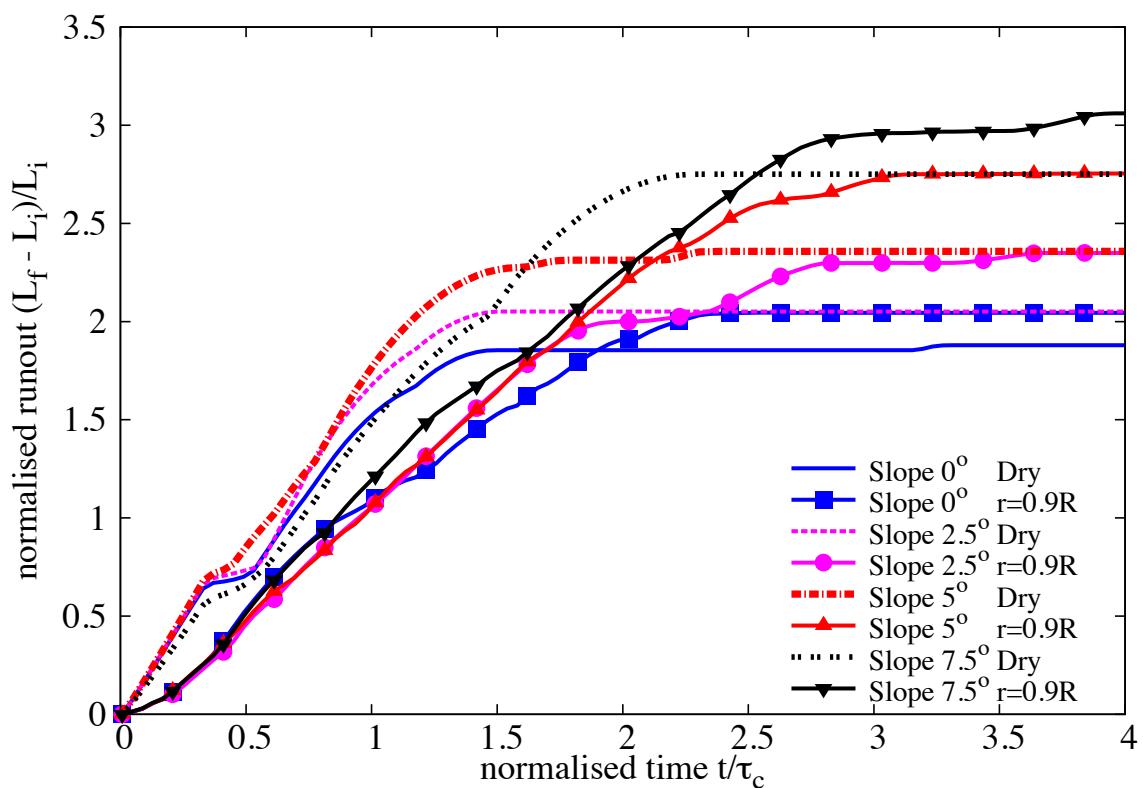
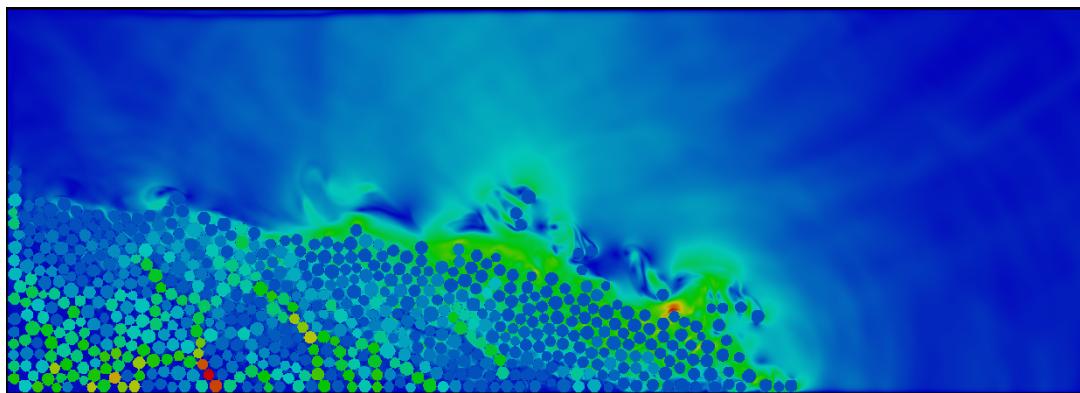
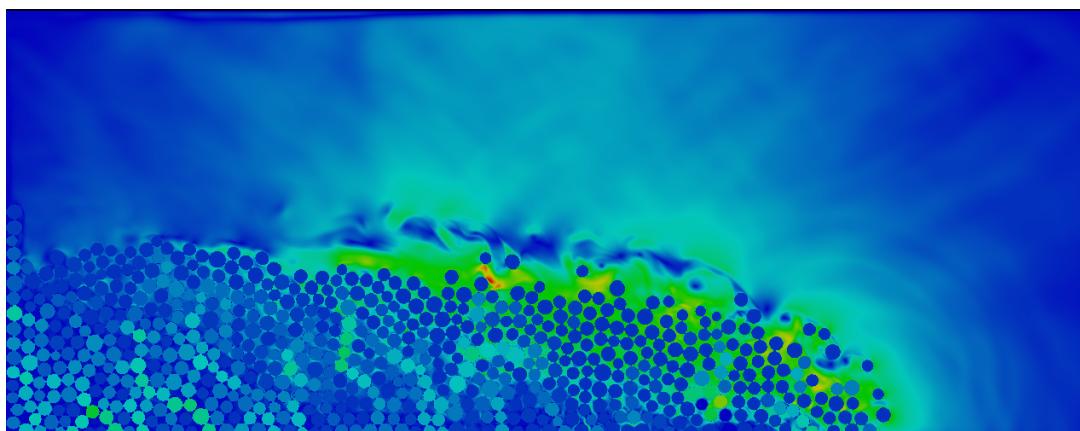


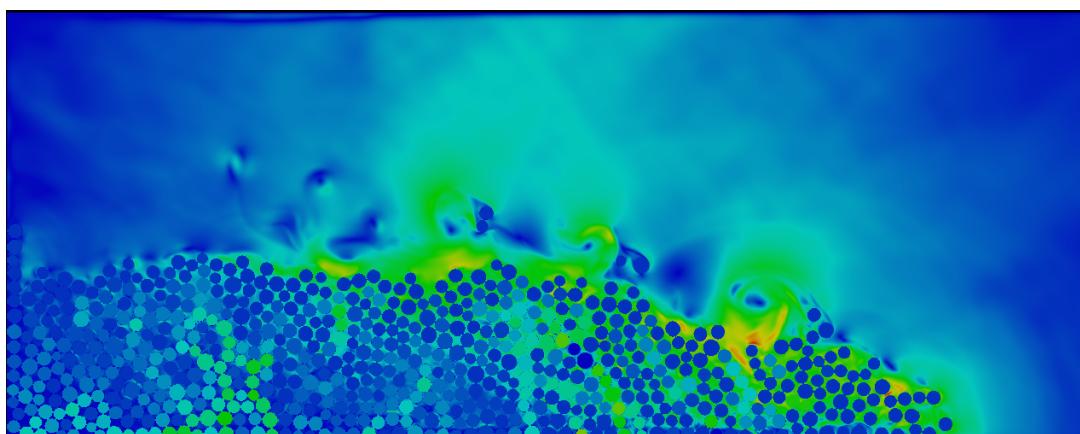
Figure 5.5 Evolution of run-out with time (loose)



(a) Slope 2.5



(b) Slope 5.0



(c) Slope 7.5

Figure 5.6 Flow morphology at critical time for different slope angles (loose)

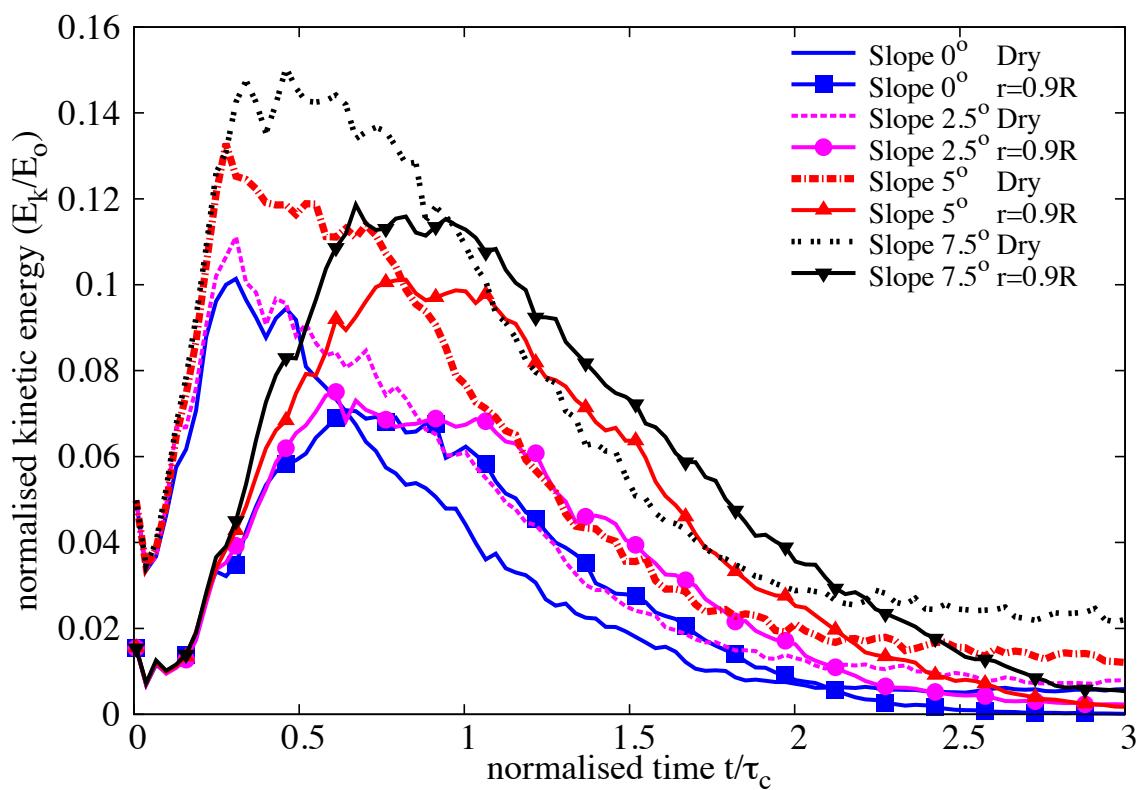


Figure 5.7 Evolution of Kinetic Energy with time (loose)

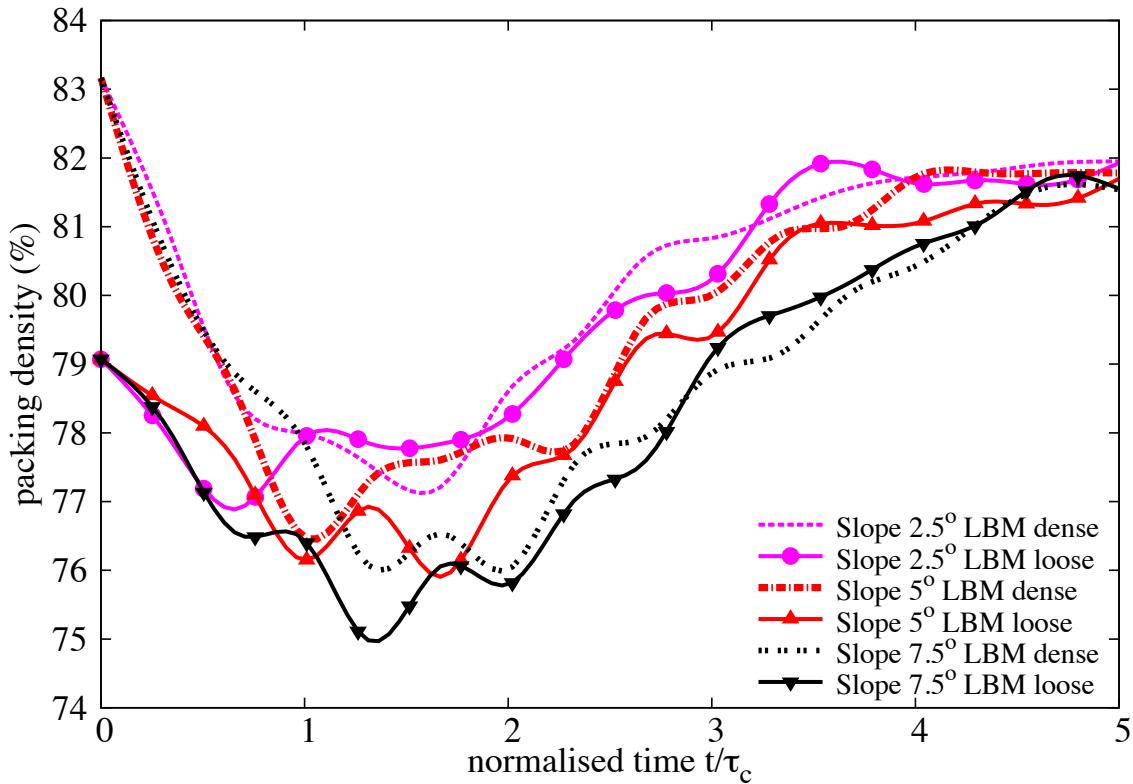


Figure 5.8 Evolution of packing density with time

pore-pressure is observed at the base of the flow, indicating entrainment of water at the base, i.e. hydroplaning resulting in longer run-out distance.

### 5.1.2 Effect of permeability

In DEM, the grain – grain interaction is described based on the overlap between the grains at the contact surface. In a 3D granular assembly, the pore spaces between grains are interconnected. Whereas in a 2-D assembly, the grains are in contact with each other that result in a non-interconnected pore-fluid space. This causes a no flow condition in a 2-D case. In order to overcome this difficulty, a reduction in radius is assumed only during the LBM computation phase (fluid and fluid – solid interaction). The reduction in radius allows interconnected pore space through which the surrounding fluid can flow. This technique has no effect on the grain – grain interactions computed using DEM. See Kumar et al. (2012) for more details about the relationship between reduction in radius and permeability of the granular assembly.

For a slope angle of 5°, the hydrodynamic radius of the loosely packed grains was varied from  $r = 0.7R$  (high permeability), 0.75R, 0.8R, 0.85R to 0.9R (low permeability). The run-out distance is found to increase with decrease in the permeability of the granular assembly

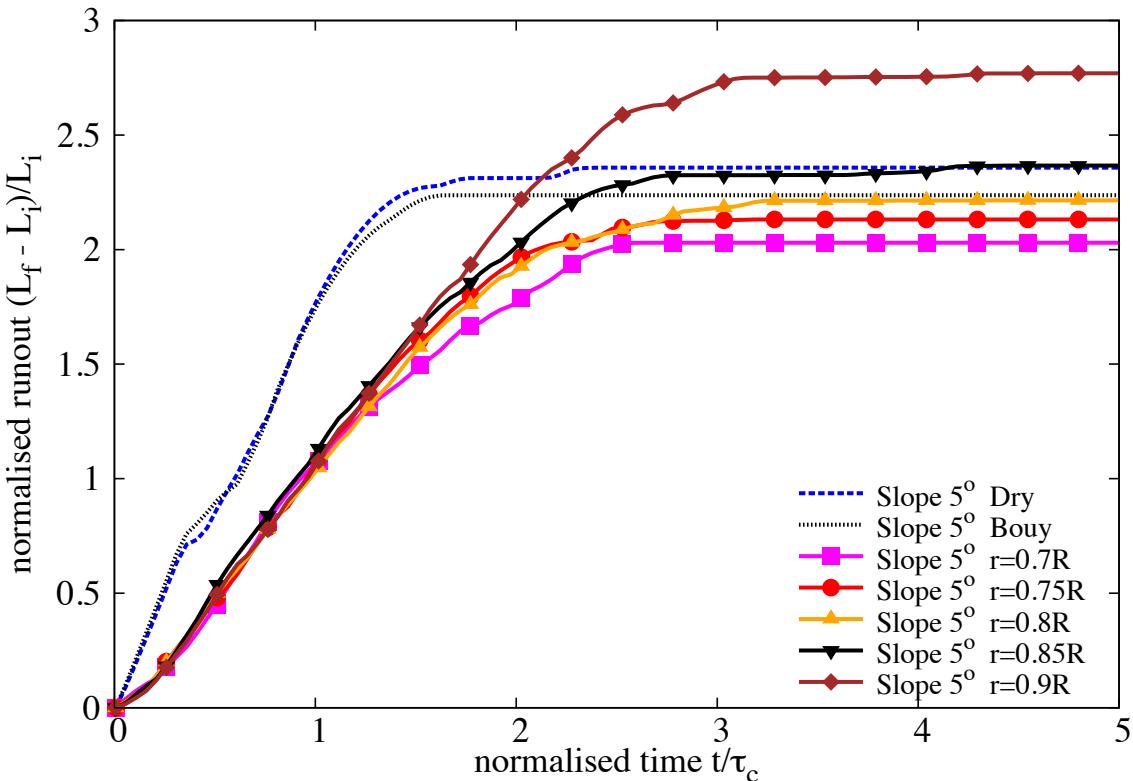


Figure 5.9 Evolution of run-out with time for different permeability (loose slope 5°)

(see Figure 5.9). The run-out distance for high permeable conditions ( $r = 0.7R - 0.8R$ ) were lower than their dry counterparts. Although, decrease in permeability resulted in an increase in the run-out distance, no significant change in the run-out behaviour was observed for a hydrodynamic radius of up to 0.8R.

With further decrease in permeability ( $r = 0.85R$  and  $0.9R$ ), the run-out distance in the fluid was greater than the run-out observed in the dry condition. At very low permeability ( $r = 0.9R$ ), granular material started to entrain more water at the base, which causes a reduction in the effective stress accompanied by a lubrication effect on the flowing granular media. This can be seen as a significant increase in the peak kinetic energy and the duration of the peak energy, in comparison with dry and high permeable conditions (see Figure 5.11).

The permeability of the granular column did not have an influence on the evolution of height during the flow. However, dry granular column tends to collapse more than the immersed granular column (see Figure 5.10).

Positive pore-pressure generation at the base of the flow was observed for low permeable conditions. Inspection of the local packing density showed entrainment of water at the base of the flow, which can also be observed by the steep decrease in the packing density (see Fig-

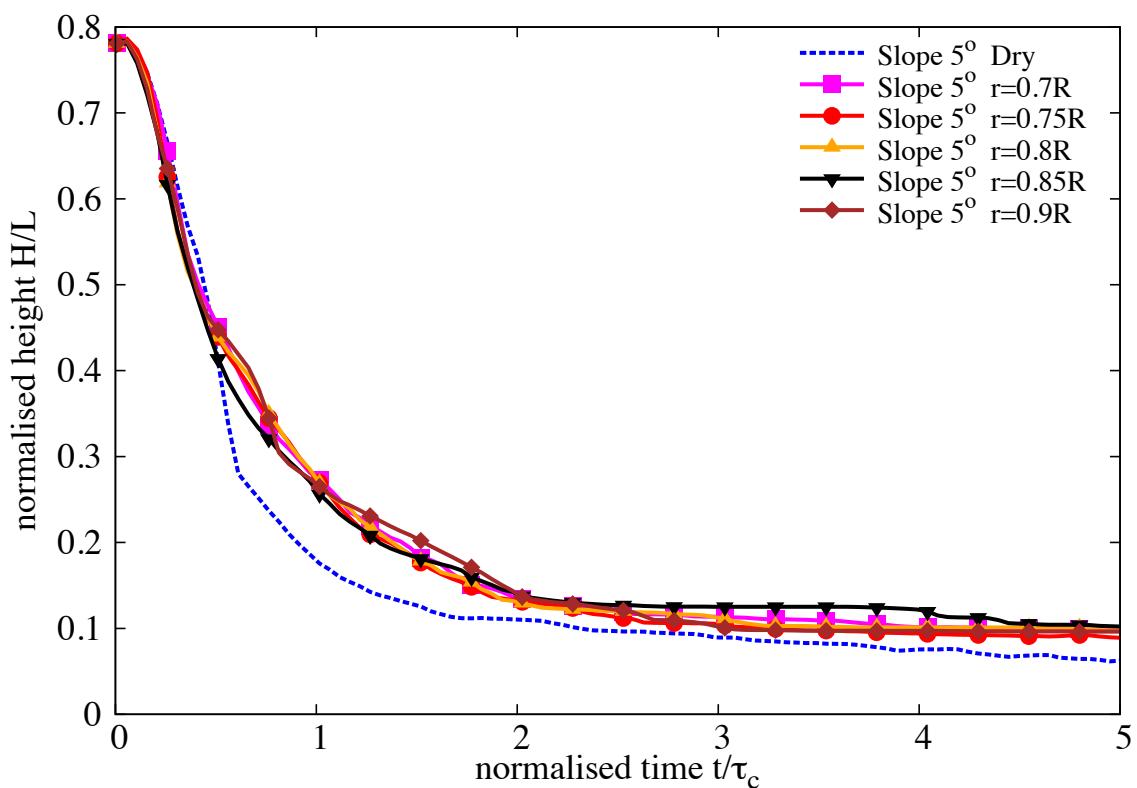


Figure 5.10 Evolution of height with time for different permeability (loose slope  $5^\circ$ )

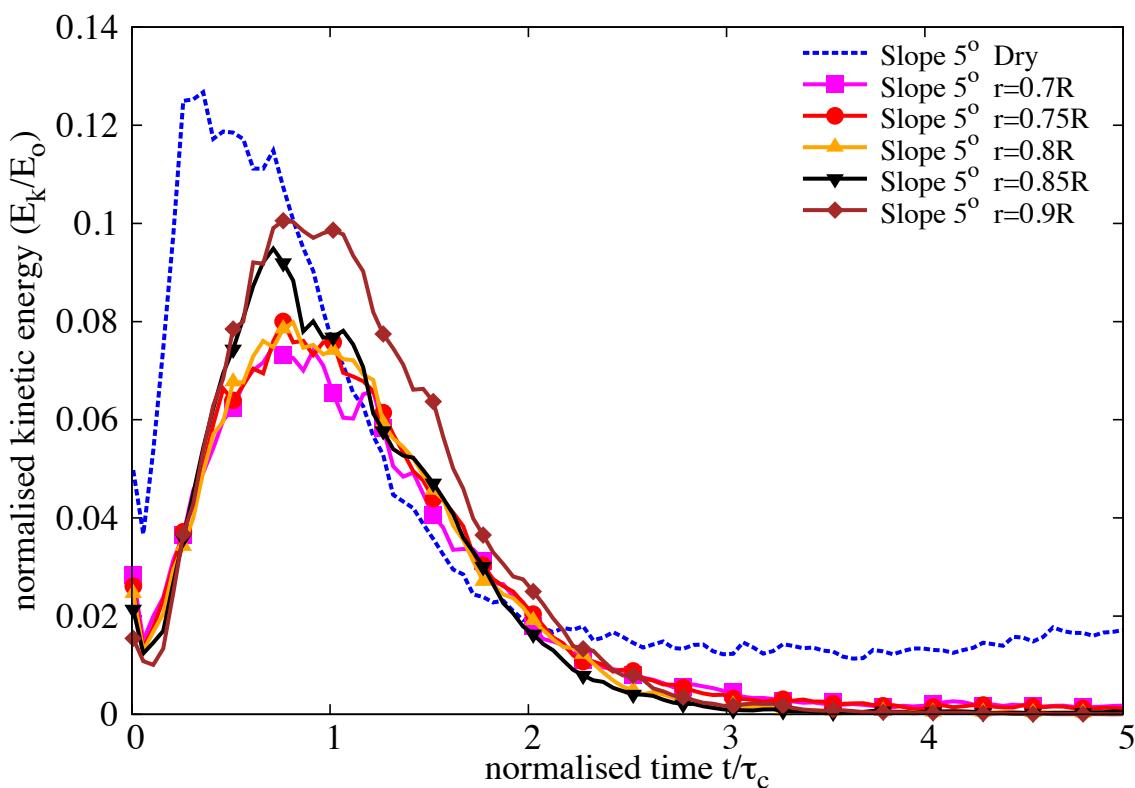


Figure 5.11 Evolution of Kinetic Energy with time for different permeability (loose slope 5°)

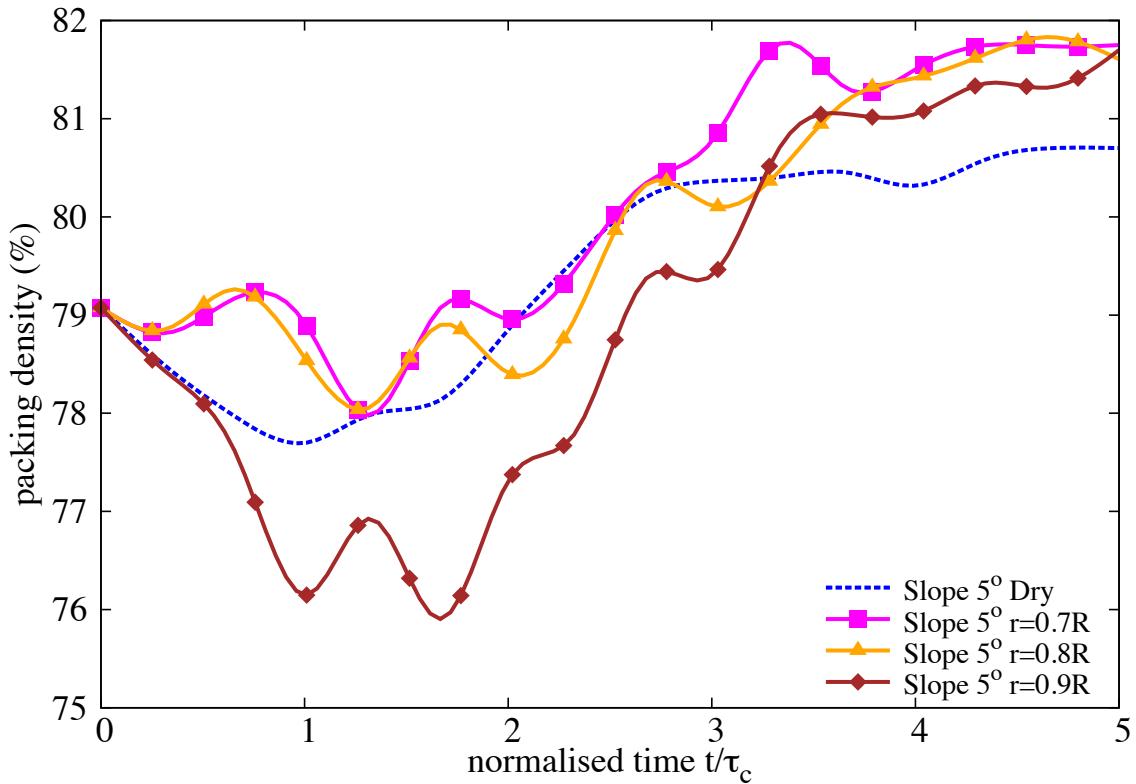


Figure 5.12 Evolution of packing density with time for different permeability (loose slope 5°)

Figure 5.12) for the very low permeability condition ( $r = 0.9R$ ). At the end of the flow ( $t \geq 3 \times \tau_c$ ), the excess pore-pressure dissipates and the granular material, irrespective of their permeability, reaches almost the same packing density.

### 5.1.3 Summary

Two-dimensional LB-DEM simulations were performed to understand the behaviour of submarine granular flows. Unlike dry granular collapse, the run-out behaviour in fluid is dictated by the initial volume fraction. Granular columns with loose packing tend to flow longer in comparison to dense columns, due to entrainment of water at the base resulting in lubrication. The loose column when it starts flowing expands and ejects liquid, leading to a partial fluidization of the material. However, with increase in the slope angle, the run-out in fluid is influenced by the viscous drag on the granular materials. The run-out distance in fluid increases with decrease in permeability. More research work is required to characterise the flow behaviour of granular materials, especially in submerged conditions.



<sup>15</sup> **References**

- <sup>16</sup> Abe, K., Soga, K., and Bandara, S. (2013). Material Point Method for Coupled Hydromechanical Problems. *Journal of Geotechnical and Geoenvironmental Engineering*, page 04013033.
- <sup>19</sup> Aidun, C., Lu, Y., and Ding, E. (1998). Direct analysis of particulate suspensions with inertia using the discrete Boltzmann equation. *Journal of Fluid Mechanics*, 373(-1):287–311.
- <sup>21</sup> 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.
- <sup>24</sup> Balmforth, N. J. and Kerswell, R. R. (2005). Granular collapse in two dimensions. *Journal of Fluid Mechanics*, 538:399–428.
- <sup>26</sup> Bandara, S. (2013). *Material Point Method to simulate Large Deformation Problems in Fluid-saturated Granular Medium*. PhD thesis, University of Cambridge.
- <sup>28</sup> Bardenhagen, S. and Kober, E. (2004). The generalized interpolation material point method. *Computer Modeling in Engineering and Sciences*, 5(6):477–496.
- <sup>30</sup> 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.
- <sup>33</sup> Bathe, K. and Zhang, H. (2004). Finite element developments for general fluid flows with structural interactions. *International Journal for Numerical Methods in Engineering*, 60(1):213–232.
- <sup>36</sup> 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.
- Breuer, M., Bernsdorf, J., Zeiser, T., and Durst, F. (2000). Accurate computations of the laminar flow past a square cylinder based on two different methods: lattice-Boltzmann and finite-volume. *Journal of Heat and Fluid Flow*, 21(2):186–196.  
<sup>1</sup>  
<sup>2</sup>  
<sup>3</sup>
- Cambou, B., Jean, M., and Radjaï, F. (2009). *Micromechanics of granular materials*. Wiley-ISTE.  
<sup>4</sup>  
<sup>5</sup>
- Capecelatro, J. and Desjardins, O. (2013). An Euler–Lagrange strategy for simulating particle-laden flows. *Journal of Computational Physics*, 238(0):1–31.  
<sup>6</sup>  
<sup>7</sup>

- 
- Chen, S. and Doolen, G. G. D. (1998). Lattice Boltzmann method for fluid flows. *Annual review of fluid mechanics*, 30(1):329–364. 8  
9
- Chen, Z. and Brannon, R. (2002). An evaluation of the material point method. *Sandia National Laboratories (SAND2002-0482)*. 10  
11
- Cook, B., Noble, D., and Williams, J. (2004). A direct simulation method for particle-fluid systems. *Engineering Computations*, 21(2/3/4):151–168. 12  
13
- Crosta, G. B., Imposimato, S., and Roddeman, D. (2009). Numerical modeling of 2-D granular step collapse on erodible and nonerodible surface. *J. Geophys. Res.*, 114(F3):F03020. 14  
15
- Daerr, A. and Douady, S. (1999). Two types of avalanche behaviour in granular media. *Nature*, 399(6733):241–243. 16  
17
- Denlinger, R. and Iverson, R. (2001). Flow of variably fluidized granular masses across three-dimensional terrain, ii: Numerical predictions and experimental tests. *J. Geophys. Res.*, 106(B1):553–566. 18  
19  
20
- Du, R., Shi, B., and Chen, X. (2006). Multi-relaxation-time lattice Boltzmann model for incompressible flow. *Physics Letters A*, 359(6):564–572. 21  
22
- Durst, F., Ray, S., Unsal, B., and Bayoumi, O. (2005). The development lengths of laminar pipe and channel flows. *Journal of fluids engineering*, 127:1154. 23  
24
- Estrada, N., Taboada, A., and Radjai, F. (2008). Shear strength and force transmission in granular media with rolling resistance. *Physical Review E*, 78(2):021301. 25  
26
- Feng, Y. T., Han, K., and Owen, D. R. J. (2007). Coupled lattice Boltzmann method and discrete element modelling of particle transport in turbulent fluid flows: Computational issues. *International Journal for Numerical Methods in Engineering*, 72(9):1111–1134. 27  
28  
29
- Fortes, A. (1987). Nonlinear mechanics of fluidization of beds of spherical particles. *Journal of Fluid Mechanics*. 30  
31
- Frisch, U. and Kolmogorov, A. (1995). *Turbulence: the legacy of AN Kolmogorov*. Cambridge University Press. 32  
33
- Girolami, L., Hergault, V., Vinay, G., and Wachs, A. (2012). A three-dimensional discrete-grain model for the simulation of dam-break rectangular collapses: comparison between numerical results and experiments. *Granular Matter*, pages 1–12. 34  
35  
36
- <sup>1</sup> Grubmuller, H., Heller, H., Windemuth, A., and Schulten, K. (1991). Generalized verlet algorithm for efficient molecular dynamics simulations with long-range interactions. *Molecular simulations*, 6:121–142.
- <sup>4</sup> Guilkey, J., Harman, T., Xia, A., Kashiwa, B., and McMurtry, P. (2003). An Eulerian-Lagrangian approach for large deformation fluid structure interaction problems, Part 1: algorithm development. *Advances in Fluid Mechanics*, 36:143–156.
- <sup>7</sup> Gumhold, S. (2003). Splatting illuminated ellipsoids with depth correction. volume 2003, pages 245–252.

- 9 Han, K., Feng, Y., and Owen, D. (2007a). Coupled lattice Boltzmann and discrete element  
10 modelling of fluid–particle interaction problems. *Computers & Structures*, 85(11-14):1080–  
11 1088.
- 12 Han, S., Zhu, P., and Lin, Z. (2007b). Two-dimensional interpolation-supplemented and  
13 Taylor-series expansion-based lattice Boltzmann method and its application. . . in *Nonlinear  
14 Science and Numerical Simulation*, 12(7):1162–1171.
- 15 Harlow, F. H. (1964). The particle-in-cell computing method for fluid dynamics. *Computer  
16 Methods in Physics*, 3:319–343.
- 17 He, X. and Luo, L. L.-S. (1997a). Theory of the lattice Boltzmann method: From the Boltzmann  
18 equation to the lattice Boltzmann equation. *Physical Review E*, 56(6):6811.
- 19 He, X. and Luo, L.-S. (1997b). A priori derivation of the lattice Boltzmann equation. *Physical  
20 Review E*, 55(6):R6333–R6336.
- 21 He, X., Zou, Q., Luo, L. S., and Dembo, M. (1997). Analytic solutions of simple flows and  
22 analysis of nonslip boundary conditions for the lattice Boltzmann BGK model. *Journal of  
23 Statistical Physics*, 87(1):115–136.
- 24 Hertz, H. (1882). Über die Berührung fester elastischer Körper. *Journal für die Reinen und  
25 Angewandte Mathematik*, 92:156–171.
- 26 Hogg, A. J. (2007). Two-dimensional granular slumps down slopes. *Physics of Fluids*, 19(9):9.
- 27 Iglberger, K., Thurey, N., and Rude, U. (2008). Simulation of moving particles in 3D with the  
28 Lattice Boltzmann method. *Computers & Mathematics with Applications*, 55(7):1461–1468.
- 29 Iverson, R. M. (1997). The physics of debris flows. *Rev. Geophys.*, 35(3):245–296.
- 30 Jaeger, H., Nagel, S., and Behringer, R. (1996). Granular solids, liquids, and gases. *Reviews of  
31 Modern Physics*, 68(4):1259–1273.
- 32 Jean, M. (1999). The non-smooth contact dynamics method. *Computer Methods in Applied  
33 Mechanics and Engineering*, 177(3-4):235–257.
- 34 Jenkins, J. T. and Savage, S. B. (1983). A theory for the rapid flow of identical, smooth, nearly  
35 elastic, spherical particles. *Journal of Fluid Mechanics*, 130:187–202.
- 36 Josserand, C., Lagree, P. Y., and Lhuillier, D. (2004). Stationary shear flows of dense granular  
37 materials: A tentative continuum modelling. *European Physical Journal E*, 14(2):127–135.
- Jullien, R., Meakin, P., and Pavlovitch, A. (1992). Random packings of spheres built with  
sequential models. *Journal of Applied Physics*, 25:4103. 1  
2
- 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. 3  
4  
5
- Kerswell, R. (2005). Dam break with Coulomb friction: A model for granular slumping?  
*Physics of Fluids*, 17:057101. 6  
7

- 
- Komiwas, V., Mege, P., Meimon, Y., and Herrmann, H. (2005). Simulation of granular flow in a fluid applied to sedimentation. *Granular Matter*, 8(1):41–54. 8  
9
- Krogh, M., Painter, J., and Hansen, C. (1997). Parallel sphere rendering. *Parallel Computing*, 23(7):961–974. 10  
11
- Kuester, F., Bruckschen, R., Hamann, B., and Joy, K. (2001). Visualization of particle traces in virtual environments. pages 151–157. 12  
13
- Kumar, K., Soga, K., and Delenne, J.-Y. (2012). *Discrete Element Modelling of Particulate Media*. Special Publication. Royal Society of Chemistry, Cambridge. 14  
15
- Lacaze, L. and Kerswell, R. R. (2009). Axisymmetric Granular Collapse: A Transient 3D Flow Test of Viscoplasticity. *Physical Review Letters*, 102(10):108305. 16  
17
- Ladd, A. J. (1994). Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation. *Journal of Fluid Mechanics*, 271:285–309. 18  
19
- Ladd, A. J. C. and Verberg, R. (2001). Lattice-Boltzmann Simulations of Particle-Fluid Suspensions. *Journal of Statistical Physics*, 104(5):1191–1251. 20  
21
- Lajeunesse, E., Mangeney-Castelnau, A., and Vilotte, J. P. (2004). Spreading of a granular mass on a horizontal plane. *Physics of Fluids*, 16(7):2371. 22  
23
- Lajeunesse, E., Monnier, J. B., and Homsy, G. M. (2005). Granular slumping on a horizontal surface. *Physics of Fluids*, 17(10). 24  
25
- Larrieu, E., Staron, L., and Hinch, E. J. (2006). Raining into shallow water as a description of the collapse of a column of grains. *Journal of Fluid Mechanics*, 554:259–270. 26  
27
- Levoy, M. (1988). Display of surfaces from volume data. *Computer Graphics and Applications, IEEE*, 8(3):29–37. 28  
29
- Li, S. and Liu, W. (2002). Meshfree and particle methods and their applications. *Applied Mechanics Reviews*, 55:1. 30  
31
- Liu, S. H., Sun, D. A., and Wang, Y. (2003). Numerical study of soil collapse behavior by discrete element modelling. *Computers and Geotechnics*, 30(Compindex):399–408. 32  
33
- Lo, C. Y., Bolton, M., and Cheng, Y. P. (2009). Discrete element simulation of granular column collapse. In *AIP Conf. Proc. Powders and Grains 2009*, volume 1145 of *6th International Conference on Micromechanics of Granular Media, Powders and Grains 2009*, pages 627–630. 34  
35  
36
- <sup>1</sup> Lorenzen, W. and Cline, H. (1987). Marching cubes: A high resolution 3D surface construction algorithm. *ACM Siggraph Computer Graphics*, 21(4):163–169.
- <sup>2</sup> Lube, G., Huppert, H. E., Sparks, R. S. J., and Freundt, A. (2005). Collapses of two-dimensional granular columns. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 72(4):1–10.
- <sup>3</sup> Luding, S., Clément, E., and Blumen, A. (1994). Anomalous energy dissipation in molecular-dynamics simulations of grains: The "detachment" effect. *Physical Review ...*, 50:4113.

- 9 Meakin, P. and Jullien, R. (1985). Structural readjustment effects in cluster-cluster aggregation.  
10 *Journal de Physique*, 46(9):1543–1552.
- 11 Mehta, A. and Barker, G. (1994). The dynamics of sand. *Reports on Progress in Physics*,  
12 57:383.
- 13 Mei, R., Shyy, W., Yu, D., and Luo, L.-S. (2000). Lattice Boltzmann Method for 3-D Flows  
14 with Curved Boundary. *Journal of Computational Physics*, 161(2):680–699.
- 15 Meruane, C., Tamburrino, A., and Roche, O. (2010). On the role of the ambient fluid on  
16 gravitational granular flow dynamics. *Journal of Fluid Mechanics*, 648:381–404.
- 17 Mitchell, J. K. and Soga, K. (2005). *Fundamentals of soil behavior*. John Wiley & Sons.
- 18 Noble, D. and Torczynski, J. (1998). A lattice-Boltzmann method for partially saturated  
19 computational cells. *International Journal of Modern Physics C-Physics and Computer*,  
20 9(8):1189–1202.
- 21 Obrecht, C. and Kuznik, F. (2011). A new approach to the lattice Boltzmann method for  
22 graphics processing units. *Computers & Mathematics* . . . .
- 23 OpenACC-Members (2013). The OpenACC Application Programming Interface Version 2.0  
24 (June, 2013; Corrected: August, 2013). Technical report, OpenMP Standard Group.
- 25 Pailha, M., Pouliquen, O., and Nicolas, M. (2008). Initiation of Submarine Granular Avalanches:  
26 Role of the Initial Volume Fraction. *AIP Conference Proceedings*, 1027(1):935–937.
- 27 Pan, C., Luo, L., and Miller, C. (2006). An evaluation of lattice Boltzmann schemes for porous  
28 medium flow simulation. *Computers & fluids*, 35(8-9):898–909.
- 29 Peker, S. and Helvacı, (2007). *Solid-liquid two phase flow*. Elsevier.
- 30 Radjai, F. and Dubois, F. (2011). *Discrete-element modeling of granular materials*. ISTE  
31 Wiley, London; Hoboken, N.J.
- 32 Radjai, F. and Richéfeu, V. (2009). Contact dynamics as a nonsmooth discrete element method.  
33 *Mechanics of Materials*, 41(6):715–728.
- 34 Radjai, F., Voivret, C., and McNamara, S. (2011). Discrete-element modelling of granular  
35 materials. chapter Periodic b, pages 181–198. ISTE Wiley.
- 36 Rondon, L., Pouliquen, O., and Aussillous, P. (2011). Granular collapse in a fluid: Role of the  
37 initial volume fraction. *Physics of Fluids*, 23(7):073301–073301–7.
- Rycroft, C. H., Bazant, M. Z., Grest, G. S., and Landry, J. W. (2006). Dynamics of random  
1 packings in granular flow. *Physical Review E*, 73(5):051306.  
2
- Rycroft, C. H., Orpe, A. V., and Kudrolli, A. (2009). Physical test of a particle simulation  
3 model in a sheared granular system. *Physical Review E*, 80(3):031305.  
4
- Schaefer, D. G. (1990). Instability and ill-posedness in the deformation of granular materials.  
5 *International Journal for Numerical and Analytical Methods in Geomechanics*, 14(4):253–  
6 278.  
7

- 
- Smagorinsky, J. (1963). General circulation experiments with the primitive equations. *Monthly weather review*, 91(3):99–164. 8  
9
- Staron, L. and Hinch, E. J. (2005). Study of the collapse of granular columns using two-dimensional discrete-grain simulation. *Journal of Fluid Mechanics*, 545:1–27. 10  
11
- Staron, L. and Hinch, E. J. (2006). The spreading of a granular mass: role of grain properties and initial conditions. *Granular Matter*, 9(3-4):205–217. 12  
13
- Staron, L. and Hinch, E. J. (2007). The spreading of a granular mass: Role of grain properties and initial conditions. *Granular Matter*, 9(3-4):205–217. 14  
15
- Staron, L. and Lajeunesse, E. (2009). Understanding how volume affects the mobility of dry debris flows. *Geophys. Res. Lett.*, 36(12):L12402. 16  
17
- 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. 18  
19  
20
- Steffen, M., Wallstedt, P., Guilkey, J., Kirby, R., and Berzins, M. (2008). Examination and analysis of implementation choices within the material point method. *Computer Modeling in Engineering and Sciences*, 32:107–127. 21  
22  
23
- Succi, S. (2001). *The lattice Boltzmann equation for fluid dynamics and beyond*. Oxford University Press. 24  
25
- Succi, S., Foti, E., and Higuera, F. (1989). Three-Dimensional Flows in Complex Geometries with the Lattice Boltzmann Method. *Europhysics Letters (EPL)*, 10(5):433–438. 26  
27
- Sukop, M. and Thorne, D. (2006). *Lattice Boltzmann modeling: An introduction for geoscientists and engineers*. Springer Verlag. 28  
29
- 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. 2498  
2499
- Sulsky, D., Zhou, S.-J., and Schreyer, H. L. (1995). *Application of a particle-in-cell method to solid mechanics*, volume 87. 2500  
2501
- Sun, X., Sakai, M., and Yamada, Y. (2013). Three-dimensional simulation of a solid–liquid flow by the DEM–SPH method. *Journal of Computational Physics*, 248(0):147–176. 2502  
2503
- Topin, V., Dubois, F., Monerie, Y., Perales, F., and Wachs, A. (2011). Micro-rheology of dense particulate flows: Application to immersed avalanches. *Journal of Non-Newtonian Fluid Mechanics*, 166(1-2):63–72. 2504  
2505  
2506
- Topin, V., Monerie, Y., Perales, F., and Radjaï, F. (2012). Collapse Dynamics and Runout of Dense Granular Materials in a Fluid. *Physical Review Letters*, 109(18):188001. 2507  
2508
- Utili, S., Zhao, T., and Housby, G. (2014). 3D DEM investigation of granular column collapse: Evaluation of debris motion and its destructive power. *Engineering Geology*. 2509  
2510
- Verlet, L. (1967). Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. *Phys. Rev.*, 159(1):98. 2511  
2512

- Vold, M. J. (1959). A numerical approach to the problem of sediment volume. *Journal of Colloid Science*, 14(2):168–174. 2513  
2514
- Wang, L., Guo, Z., and Mi, J. (2014). Drafting, kissing and tumbling process of two particles with different sizes. *Computers & Fluids*. 2515  
2516
- Warnett, J. M., Denissenko, P., Thomas, P. J., Kiraci, E., and Williams, M. A. (2013). Scalings of axisymmetric granular column collapse. *Granular Matter*, 16(1):115–124. 2517  
2518
- Willis, A., Peixinho, J., Kerswell, R., and Mullin, T. (2008). Experimental and theoretical progress in pipe flow transition. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1876):2671. 2519  
2520  
2521
- Wolfe, M. (2012). PGInsider August 2012: OpenACC Kernels and Parallel Constructs. 2522
- Xiong, Q., Madadi-Kandjani, E., and Lorenzini, G. (2014). A LBM–DEM solver for fast discrete particle simulation of particle–fluid flows. *Continuum Mechanics and Thermodynamics*. 2523  
2524  
2525
- Yu, D., Mei, R., Luo, L., and Shyy, W. (2003). Viscous flow computations with the method of Lattice Boltzmann equation. *Progress in Aerospace Sciences*, 39(5):329–367. 2526  
2527
- Yu, H., Girimaji, S., and Luo, L. (2005). Lattice Boltzmann simulations of decaying homogeneous isotropic turbulence. *Physical Review E*, 71(1):016708. 2528  
2529
- Zenit, R. (2005). Computer simulations of the collapse of a granular column. *Physics of Fluids*, 17(Compendex):031703–1–031703–4. 2530  
2531
- Zhong, Q. and Olson, M. (1991). Finite element–algebraic closure analysis of turbulent separated-reattaching flow around a rectangular body. *Computer methods in applied mechanics and engineering*, 85(2):131–150. 2532  
2533  
2534
- Zhou, H., Mo, G., Wu, F., Zhao, J., Rui, M., and Cen, K. (2012). GPU implementation of lattice Boltzmann method for flows with curved boundaries. *Computer Methods in Applied Mechanics and Engineering*, 225-228(null):65–73. 2535  
2536  
2537
- Ziegler, D. (1993). Boundary conditions for Lattice Boltzmann simulations. *Journal of Statistical Physics*, 71(5):1171–1177. 2538  
2539
- Zou, Q. and He, X. (1997). On pressure and velocity boundary conditions for the Lattice Boltzmann BGK model. *Physics of Fluids*, 9(6):1591–1598. 2540  
2541