

## A Hybrid Algorithm of Event-Driven and Time-Driven Methods for Simulations of Granular Flows

Jun Huang\* and Ole Jørgen Nydal

*Department of Energy and Process Engineering, Norwegian University of Science and Technology, Trondheim, Norway.*

Received 16 June 2010; Accepted (in revised version) 21 December 2010

Communicated by Boo Cheong Khoo

Available online 24 June 2011

---

**Abstract.** The classical discrete element approach (DEM) based on Newtonian dynamics can be divided into two major groups, event-driven methods (EDM) and time-driven methods (TDM). Generally speaking, TDM simulations are suited for cases with high volume fractions where there are collisions between multiple objects. EDM simulations are suited for cases with low volume fractions from the viewpoint of CPU time. A method combining EDM and TDM called Hybrid Algorithm of event-driven and time-driven methods (HAET) is presented in this paper. The HAET method employs TDM for the areas with high volume fractions and EDM for the remaining areas with low volume fractions. It can decrease the CPU time for simulating granular flows with strongly non-uniform volume fractions. In addition, a modified EDM algorithm using a constant time as the lower time step limit is presented. Finally, an example is presented to demonstrate the hybrid algorithm.

**AMS subject classifications:** 70F05, 65K05, 74S30, 70F10

**Key words:** Time-driven method (TDM), event-driven method (EDM), Hybrid algorithm of event-driven and time-driven methods (HAET), granular flow.

---

## 1 Introduction

Granular materials are very commonly used in industry and in our daily life. Yet many aspects of granular flows are poorly understood. Granular materials are complex systems composed of a very large number of solid particles. The motion of each particle is defined by classical Newtonian mechanics and contact mechanics. Based on the relative velocity between particles, granular material is classified into three phases, as gas-like, liquid-like and solid-like [1]. Important parameters used to determine the phase are

---

\*Corresponding author. *Email addresses:* jun.huang@ntnu.no (J. Huang), ole.j.nydal@ntnu.no (O. J. Nydal)

the volume fraction  $\phi$  and coefficients of restitution (COR) [2]. With respect to the flow cases and the accuracy required, various methods can be used to simulate the motion of granular materials numerically. According to Hogue and Newland [3], the methods are classified into two approaches: continuum mechanics methods (CMM) and discrete element methods (DEM). The CMM uses the Eulerian approach to simulate granular material behavior. Several reviews of CMM are available [4–6]. The DEM is based on the Lagrangian approach to simulate the motion of each particle at the microscopic scale. It can be divided into three main classes [4]: *Statistical mechanics models* [7], *Newtonian dynamics models*, and *Hybrid models* [3]. Furthermore, the Newtonian dynamics models can be divided into two major groups, *Event-driven methods* (EDM) and *Time-driven methods* (TDM). A detailed review of the Newtonian dynamics models is presented in the next sections. This paper presents a new hybrid algorithm EDM and TDM. It can be used for strongly non-uniform flow field with a higher efficiency than TDM. In this hybrid algorithm, the choice of using either EDM or TDM depends on the local volume fraction. Hence it is required that the velocity changes determined from these two models should be close to each other for the same binary collision. To characterize the velocity changes, two COR,  $e$  and  $\beta$  are introduced for the normal direction and the tangential direction respectively. The definitions of these two coefficients are expressed as the ratio of the post-collisional relative velocities over the pre-collisional relative velocities at the contact point, given by

$$e = -\frac{\mathbf{k} \cdot \mathbf{u}_{ij}^{lo}}{\mathbf{k} \cdot \mathbf{u}_{ij}^o}, \quad (1.1a)$$

$$\beta = -\frac{\mathbf{k} \times \mathbf{u}_{ij}^{lo}}{\mathbf{k} \times \mathbf{u}_{ij}^o}, \quad (1.1b)$$

where  $\mathbf{k}$  is the unit vector along the center line from particle  $i$  to  $j$  and  $\mathbf{u}_{ij}^o$  is the relative velocity at the contact point. In this paper, vector variables are represented by bold font and scalar variables are represented by italic font. The ranges of the two COR are  $0 \leq e \leq 1$ , and  $-1 \leq \beta \leq 1$  respectively.

## 2 Time-driven method

The calculation of the contact force between each discrete element in the time-driven method used in the HAET algorithm is based on the original method of Cundall and Strack [8]. An integration method, which is introduced in Section 4, can be employed to calculate the changes in velocity and position for each particle after a certain time by considering all the forces on the particle. Some of the forces acting between particles originate from the deformation of the particles when they are in contact with their neighbors, as shown in the Fig. 1. Ramirez [9] suggested that the normal particle-particle collision process could be modeled as a spring-dashpot system, where the normal force,  $F_n$ , is the

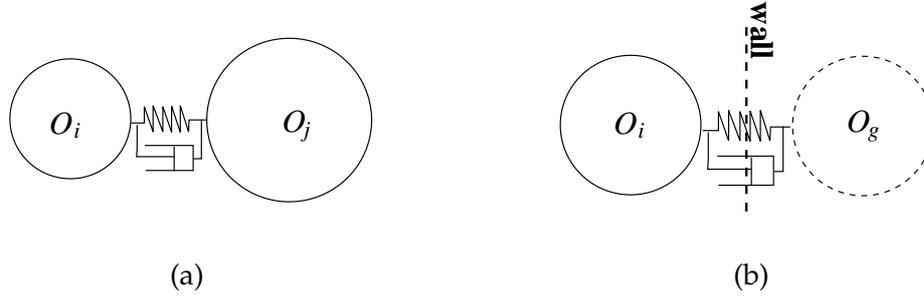


Figure 1: Spring-dashpot models: (a) binary collision and (b) collision with the wall.

sum of an elastic term,  $F_n^e$ , and a viscous term,  $F_n^d$ ,

$$F_n = F_n^e + F_n^d = K_n \delta_n^\zeta + \eta_n \delta_n^\xi \dot{\delta}_n, \tag{2.1}$$

where  $\zeta=3/2$  (see [10,11]),  $\xi=0.5$  (see [12]),  $\delta_n$  is the overlap,  $\delta_n=r_i+r_j-d_{ij}$  ( $\delta_n>0$ ),  $K_n$  is the effective stiffness,  $\eta_n$  is the damping coefficient,  $r_{i,j}$  is the radius of each particle and  $d_{ij}$  is the distance between the two centers of the particles. The coefficients in Eq. (2.1) are described in more detail in [13, 14] for a constant COR in the normal direction,  $e$ . The magnitude of tangential force,  $|\mathbf{F}_{t,i}|$ , is obtained from Coulomb's friction law where  $|\mathbf{F}_{t,i}|=\mu|\mathbf{F}_{n,i}|$ .

The normal and tangential components of impulse are given by  $d\mathbf{J}_n = \mathbf{F}_n \cdot dt$ , and  $d\mathbf{J}_t = \mathbf{F}_t \cdot dt$ , respectively, where  $dt$  is the time step in the integration. Considering the balance of impulses and the exchanges of momentum due to collisions of all particles in contact, the velocities of a particle after the time step,  $dt$ , are given by

$$\mathbf{u}'_{n,t} = \mathbf{u}_{n,t} + \frac{\sum \mathbf{J}_{n,t}}{m}, \tag{2.2a}$$

$$\boldsymbol{\omega}' = \boldsymbol{\omega} + \frac{\sum \mathbf{J}_t}{I}, \tag{2.2b}$$

where  $\mathbf{u}$  is velocity in translation and  $\boldsymbol{\omega}$  is the angular velocity in rotation. The prime means the status at the end of the time step and the corresponding symbols without prime are the status at the beginning of the time step.  $I=0.4mr^2$  is the moment of inertia for a spherical particle.

According to [15, 16], the time step in TDM should be less than 2% of the shortest binary collision time, and it cannot exceed the Rayleigh time. The Rayleigh time is the time required for a Rayleigh wave to travel the diameter of an elastic particle. Since the Rayleigh time is proportional to the binary collision time, the time steps calculated from the collision time and the Rayleigh time are very close [15,17].

### 3 Event-driven method

In the Event-driven method (EDM), also called hard particle model, there is an event, e.g., collision, inside the system which controls the system dynamics. When an event occurs, there is a change in the system dynamics. This change needs to be updated at the next time step. The Event-driven method typically has two functions to perform, calculating the next collision time and the implementing of collision dynamics. Whenever the distance between spherical particles equals the sum of the two radii, then an event, a collision, occurs. The velocities of the particles will change suddenly [18]. Consider a pair of spherical particles, as shown in Fig. 2, with radii  $r_i$  and  $r_j$ , and positions at time  $t$ ,  $\mathbf{p}_i$  and  $\mathbf{p}_j$  respectively. The time gap for the coming collision is given by the smaller real root of the equation

$$\delta t_{ij} = \frac{-\mathbf{b}_{ij} - \sqrt{-\mathbf{b}_{ij}^2 - \mathbf{u}_{ij}^2 (\mathbf{p}_{ij}^2 - (r_i + r_j)^2)}}{\mathbf{u}_{ij}}, \tag{3.1}$$

where  $\mathbf{u}_{ij}$  is the magnitude of the relative velocity between the two particles,  $\mathbf{u}_{ij} = \mathbf{u}_i - \mathbf{u}_j$ , and  $\mathbf{b}_{ij}$  is defined as  $\mathbf{b}_{ij} = \mathbf{u}_{ij} \mathbf{p}_{ij}$  with  $\mathbf{p}_{ij} = \mathbf{p}_i - \mathbf{p}_j$ . For a multi-body system, the time step is the time gap between consecutive events, that  $dt_{EDM} = \min(\delta t_{ij})$ . In other words, it is determined by the shortest free flight time between all the particles [4]. The number of particles in the system also affects the time step in the EDM simulations. From a statistical point of view, the larger the number of particles in the system, the higher probability of the existence of a short free flight path even when the global volume fraction is the same.

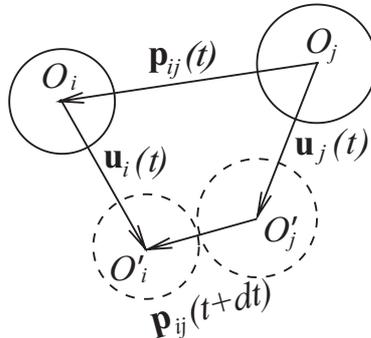


Figure 2: Illustration of relative position vectors before and at the collision time with the corresponding velocity vectors.

There are several EDM models for the relationships between the coefficients  $e$  and  $\beta$ . According to Zamankhan and Huang [19],  $\beta$  can be written as

$$\beta = -1 + \mu \frac{1+e}{m_r} \left( m_r + \frac{m_i}{I_{nd,i}} + \frac{m_j}{I_{nd,j}} \right) \left| \frac{u_{ij}^n}{u_{ij}^t} \right|, \tag{3.2}$$

where  $I_{nd}$  is the non-dimensional moment of inertia. For spherical particles,  $I_{nd} = 0.4$ .  $u_{ij}^n$  and  $u_{ij}^t$  are the two components of relative velocity in the normal and tangential direction

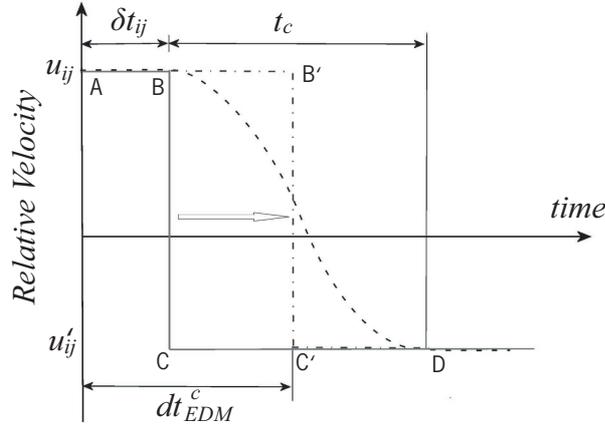


Figure 3: Illustration of different collision models: dashed line for TDM, solid line for the original EDM algorithm using  $\delta t_{ij}$  as time step and the dash-dotted line for the modified EDM algorithm in the HAET method using a constant time step  $dt_{EDM}^c$ .

respectively. Lun and Bent [20] suggested a phenomenological constant  $\beta_0$  characterizing the COR in the tangential direction for a sticking collision. The measurements of Maw [21], suggest that  $\beta$  cannot be greater than 0.4. Follow the spring-dashpot model for the normal direction given above, a similar form for the tangential component is given by Zamankhan and Bordbar [22].

Combining the *law of conservation of momentum* and the relations between the velocity changes in the tangential direction and in rotation [23],

$$\left| \frac{\Delta \mathbf{u}_{i,t}}{\Delta \boldsymbol{\omega}_i} \right| = \frac{r_i}{2.5}. \tag{3.3}$$

The details of the derivation are given in the appendix.

The omission of the collision process in the original EDM algorithm also causes some difference with the TDM model. As shown in Fig. 3, the relative velocity from TDM is plotted by a dashed line and that of the original EDM algorithm is by a solid line. The origin of the time axis starts at the last time step. In the original EDM algorithm, the time step is obtained from Eq. (3.1). The difference in the distances between the two particles after collisions obtained from the two methods is the area between these two curves, which is given by

$$\varepsilon = \frac{u_{ij}(1+e)}{4} t_c + \mathcal{O}(u_{ij}). \tag{3.4}$$

If a constant time  $dt_{EDM}^c$  is used as the time step instead of  $\delta t_{ij}$ , such that

$$\delta t_{ij} < dt_{EDM}^c < \delta t_{ij} + t_c, \tag{3.5}$$

the EDM curve shifts from BC to B'C' and the difference  $\varepsilon$  becomes smaller. However, this longer time step may also cause the two particles to overlap with each other. This is not allowed in the hard particle model, if the integration method introduced in Section 4 is used. To avoid the overlap, two artificial displacements are introduced at the end of time step,  $dt_{EDM}^c$ . The artificial displacements for particles  $i$  and  $j$  are given as follows:

$$\mathbf{p}_i(t+dt_{EDM}^c) = \mathbf{u}_i \left( \frac{\mathbf{p}_{ij}(t)}{\mathbf{u}_j^n - \mathbf{u}_i^n} \right), \quad (3.6a)$$

$$\mathbf{p}_j(t+dt_{EDM}^c) = \mathbf{u}_j \left( \frac{\mathbf{p}_{ij}(t)}{\mathbf{u}_j^n - \mathbf{u}_i^n} \right). \quad (3.6b)$$

This modification to the original EDM algorithm is only valid for particles in contact with one neighbor at the end of a particular time step, because the artificial displacements cannot ensure that multi-body systems do not overlap. The solution to this limitation is discussed in Section 6. To avoid solving the complex quadratic equation, Eq. (3.1), and the comparison for  $\min(\delta t_{ij})$ , a constant time step,  $dt_{EDM}^c$ , can be used in the modified EDM algorithm. When the number of particles is huge, the probability of obtaining  $\min(\delta t_{ij}) \ll t_c$  is high, which is undesirable.

## 4 Integration algorithm

The Verlet algorithm is a second-order method for integrating the equations of motion [24]. The original equation for the position is given by

$$\mathbf{p}_{(t+dt)} = 2\mathbf{p}_t - \mathbf{p}_{(t-dt)} + \mathbf{a}_t dt^2, \quad (4.1)$$

and the velocity can be obtained from the formula

$$\mathbf{u}_t = \frac{\mathbf{p}_{(t+dt)} - \mathbf{p}_{(t-dt)}}{2dt}. \quad (4.2)$$

Various modifications have led to several different forms of the Verlet algorithm. Illustrators of these forms are shown in Fig. 4. No matter which form is used, the acceleration plays an important role in the integration. For well-packed particles located in an acceleration field, all the forces should be balanced at rest,  $\sum F = 0$ . However, due to the lack of contact forces the forces at rest do not sum to zero again when EDM is used. Hence, the acceleration,  $\mathbf{a}$ , the changes of velocity,  $d\mathbf{u}$ , and position,  $d\mathbf{p}$ , will not be zero. These errors are proportional to the time step  $dt_{EDM}$ , which may increase the total energy in the whole system meaning the simulation will not converge, unless  $dt_{EDM}$  approaches zero.

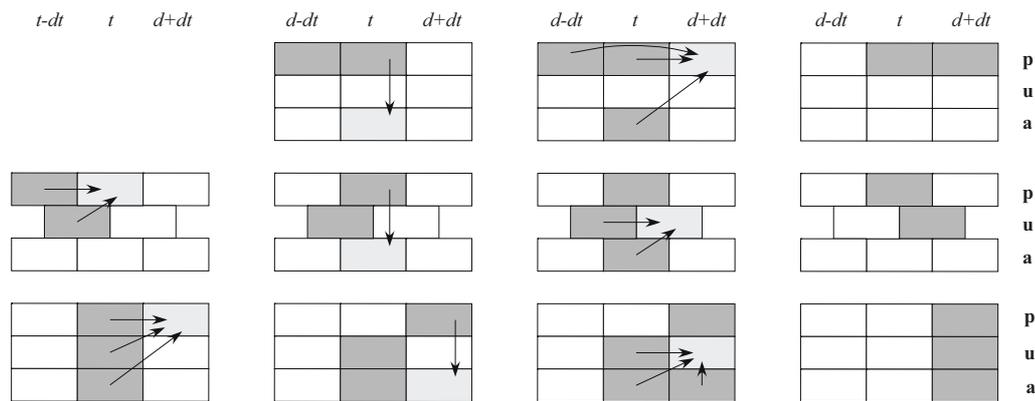


Figure 4: Various forms of the Verlet algorithm. (a) The original form, (b) the leapfrog form, and (c) the velocity form.

## 5 Neighbor list

If there is a possibility of binary interaction between any pair of particles in an  $N$ -particle system, it needs  $N(N-1)/2$  operations to calculate the interactions in each time step. This procedure takes up most of the CPU time. Fortunately, the only interaction among particles in granular material when they contact. This is a kind of short-range force. The conventionally *Verlet table method* and *Link cell method* have been introduced by Allen and Tildesley [18] for such short-range potential.

The Verlet table method sets up a list of neighbors for every particle in the system [18, 24]. This list is updated periodically after a certain number of time steps as some particles move in and out of the neighboring region. Once the neighbor list is constructed, the evaluation of the position of a colliding pair is efficient. This procedure requires on the order of  $N \cdot N_{neighbor}$  steps, where  $N_{neighbor}$  is the average number of neighbors for each particle in the Verlet table.

One important problem in the Verlet table method is how to set it up effectively. The link cell list algorithm is a method that can reduce that when the number of particle is huge. In this method, the simulation domain is partitioned into several cells where the size of the cells is larger than that of the particles. The particles are assigned a cell according to their positions. There are eight neighboring cells for a 2-D problem (and 26 cells for a 3-D problem) as shown in Fig. 5. Thus for each particle, all particles in the same cell and the neighboring cells are added to the neighbor list and are evaluated [25]. The amount of time between consecutive Verlet table searches is given by

$$dt_{BL} = \frac{L_{cell} - (r_i + r_j)}{2|\mathbf{u}|_{\max}}, \tag{5.1}$$

where  $L_{cell}$  is the side length of the cells and  $|\mathbf{u}|_{\max}$  is the maximum magnitude of the velocity of the particles in the system. To prevent  $|\mathbf{u}|_{\max}$  increase in the time  $dt_{BL}$ , a

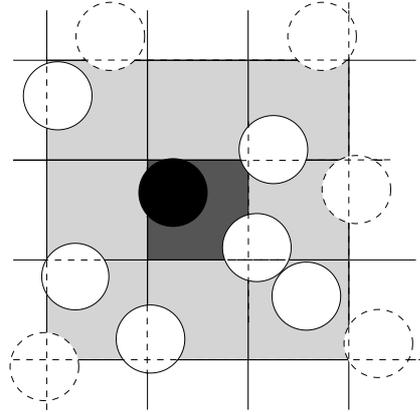


Figure 5: Illustration of the link cell method: for the black particle located in the central dark gray cell, only those solid particles in the gray cells are added to its Verlet table.

conservative estimation  $dt_{BL}$  is given by

$$dt_{BL} = C \frac{L_{cell} - (r_i + r_j)}{2|\mathbf{u}|_{\max}}, \quad (5.2)$$

where  $C$  is a coefficient less than 1.

## 6 Hybrid algorithm

Usually, EDM simulations are faster than TDM simulations, because the colliding process is modeled and the time step  $dt_{EDM}$  can be much larger than  $dt_{TDM}$ . However, particles in EDM cannot overlap with each other due to the lack of some mechanics to balance the overlap and interaction force. This is a particular disadvantage for well-packed particles in an accelerating field at quasi-steady state. Because there is no feedback system to control the magnitude of overlap, the volume of particles becomes un-physical if the overlap approaches an unreasonable value. Controlling  $dt_{EDM}$  is a valid way to avoid the overlap in the original EDM algorithm. For example, a particle arrives at steady state,  $|\mathbf{u}| = 0$  at time,  $t$ . The displacement of particle is  $|\mathbf{p}| = 0.5|\mathbf{a}|dt_{EDM}^2$  after a time step, and therefore  $dt_{EDM}$  approach finite to avoid overlap. TDM is better suited for such cases.

The volume fraction of granular flow is not always uniform, especially when there are obstacles or corners in the flow field. For example, when granular flow passes over an obstacle, a dynamic dune with an inner triangular region with zero velocity and high volume fraction is reported [26–28]. If the advantages of both EDM and TDM can be combined together, the simulation can be more effective. The basic idea of the improved algorithm, HAET, is to use the TDM for those areas with high volume fraction,  $\phi$ , and the EDM for the remaining areas. A flow chart of the decision part in the new algorithm is given in Fig. 6. Here, the output,  $K$ , is an  $N$ -element array, in which every element is a

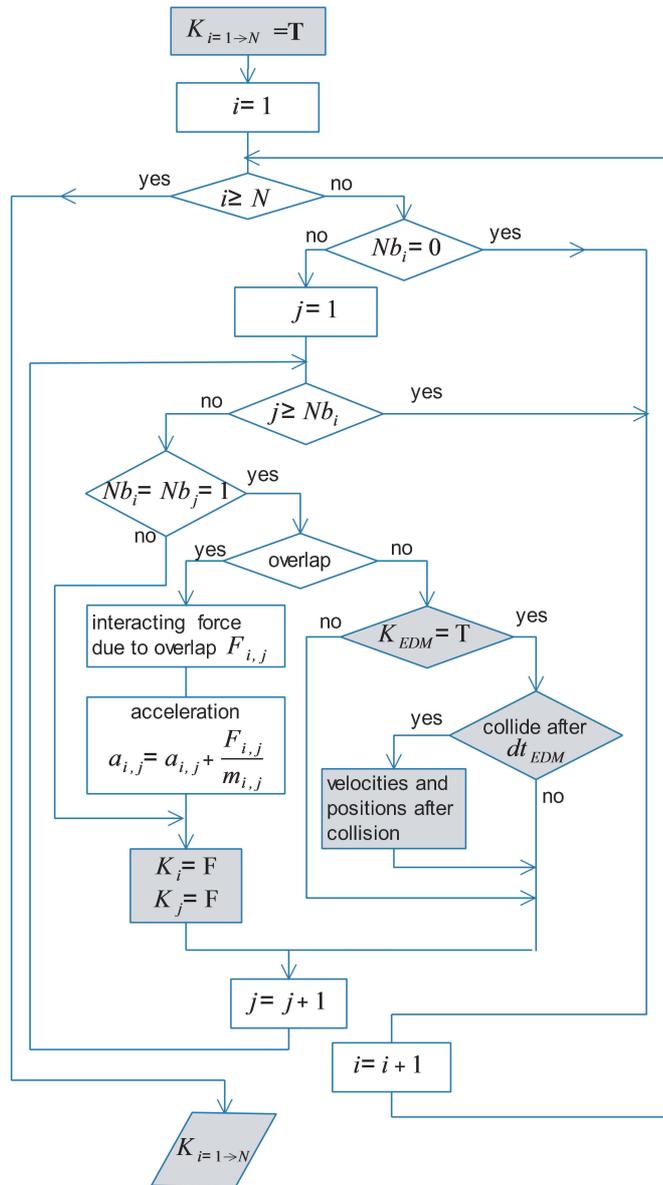


Figure 6: Flowchart about the conditional parameter  $K_i$ . The flowchart for TDM does not include the dark parts and only the accelerations due to collision are obtained. Here, T for True and F for Fail.

conditional parameter that chooses whether to use TDM or EDM for the corresponding particle. The flow chart of the integration part is shown in Fig. 7.

The main challenge of this hybrid algorithm is to define the critical volume fraction,  $\phi_c$ . As discussed previously, for a particle, the next collision must occur with one of the particles located in the link cells. For the EDM algorithm, the binary collision, the event,

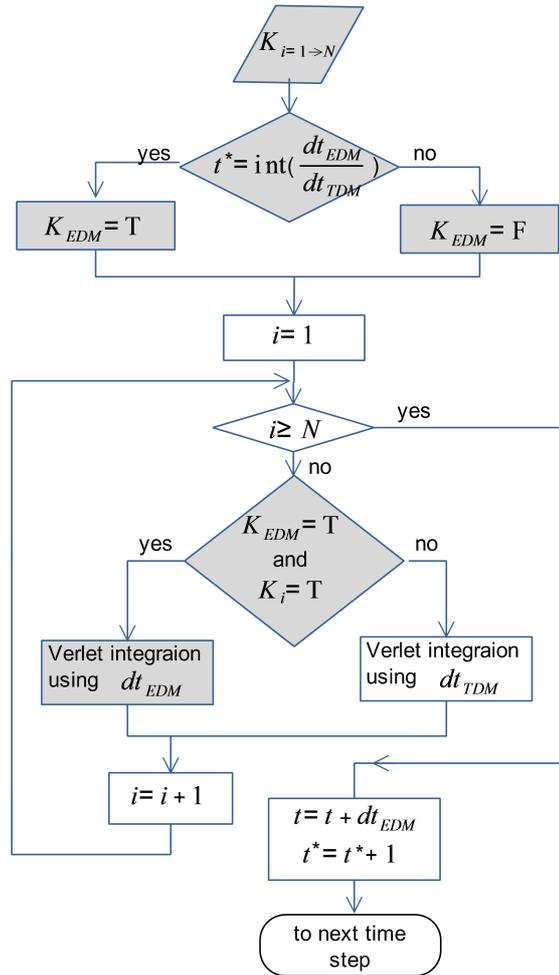


Figure 7: Flowchart of the integration part in HAET. The flowchart for TDM without the dark parts. Here, T for True and F for Fail.

happens instantaneously. The simplest solution is that the number of neighbors in the link cells is one for each of the two colliding particles. In other words, the particles are each other’s only neighbor, and, therefore, the modified EDM algorithm can be used in HAET. Because of the valid range shown in Eq. (3.5), the time step for the EDM algorithm can be as long as the collision time, and

$$dt_{TDM} \ll dt_{EDM}^c \ll dt_{BL}. \tag{6.1}$$

Secondly, avoiding overlap is necessary in the EDM algorithm. Hence, another necessary condition for using the EDM in the hybrid algorithm is that the distance between the two colliding particles should be no less than the sum of the radii before collision,

$$|\mathbf{p}_{ij}| \geq r_i + r_j. \tag{6.2}$$

Thirdly, the technique for solving the particle-wall contact can also be very important. Like the methods for the particle-particle collision, the particle-wall collision also can be classified as EDM or TDM. When a particle collides with the wall, an imaginary particle is created temporarily. The imaginary particle is a mirror of the real one and the surface of the wall is the plane of symmetry. A collision happens between the real and the imaginary particles.

Lastly, the time step  $dt_{EDM}$  in the hybrid algorithm is used for describing the instantaneous collision. The forces at rest, e.g., gravity and interaction with the wall, are still integrated using the time step of TDM,  $dt_{TDM}$ . In other words, only some of the CPU time is reduced by using the EDM model.

## 7 Effect of the hybrid algorithm

Although hoppers are very common industrial and agricultural appliances, the dynamics of the granular materials flowing inside them is not clearly understood [29]. The dilute-to-dense and dense-jamming transition of granular flow was investigated in a two dimensional channel by Hou et al. [30]. If the size of grain is smaller than the exit, two wedges will be formed at both sides of the exit for the 2-D problem shown in Fig. 8(a). According to the velocity profile and volume fraction, the entire system can be classified into three regions: *free drop zone*; *condensed zone* in the wedges, also called *fast collision zone*; and *chute zone*. For these two wedges, TDM is more suited than EDM because of the high volume fraction. However, for the simulation of the other regions, EDM is much better.

Here, three simulations of the hopper are run by using TDM, EDM and HAET respectively with the same geometry and initial conditions, as shown in Fig. 8(b). The hopper is 0.25m high and 0.06m wide. A total of 1200 particles with diameter 0.5mm were put into the hopper. The initial positions of these particles are random and the initial velocities are 10m/s downward. In other words, the initial area fraction of the solid phase is 0.0157. The gravity force is also downward. The normal coefficient of restitution  $e=0.8$ , the surface roughness  $\mu=0.2$  and the binary collision models introduced in Sections 2 and 3 were used in the simulations. The size of link cell used is  $L_{cell}=4r$  and the three different time steps are  $dt_{BL}=2.5\times 10^{-5}s$ ,  $dt_{EDM}^c=t_c=2.8\times 10^{-8}s$ , and  $dt_{TDM}^c=5.6\times 10^{-10}s$  respectively. Because the size of the particles is much greater than that suggested by Midi [31], the drag force from the interstitial air was omitted. The result indicates that EDM cannot be used for such strongly nonuniform volume fraction distribution. Figs. 8(c) and (d) show the instantaneous particle distributions at  $t=0.02s$  by using TDM and HAET methods, respectively. The contour figures of volume fraction  $\phi$  and averaged magnitude of velocity are presented in Figs. 8(e)-(h), for TDM and HAET at the same time. The units for the magnitude of velocity are m/s. The results of the two methods are close to each other and the condensed zones in the wedges are obvious.

Fig. 9(a) presents the dimensionless CPU time,  $t_{CPU}^*$ , from the three methods. Here,

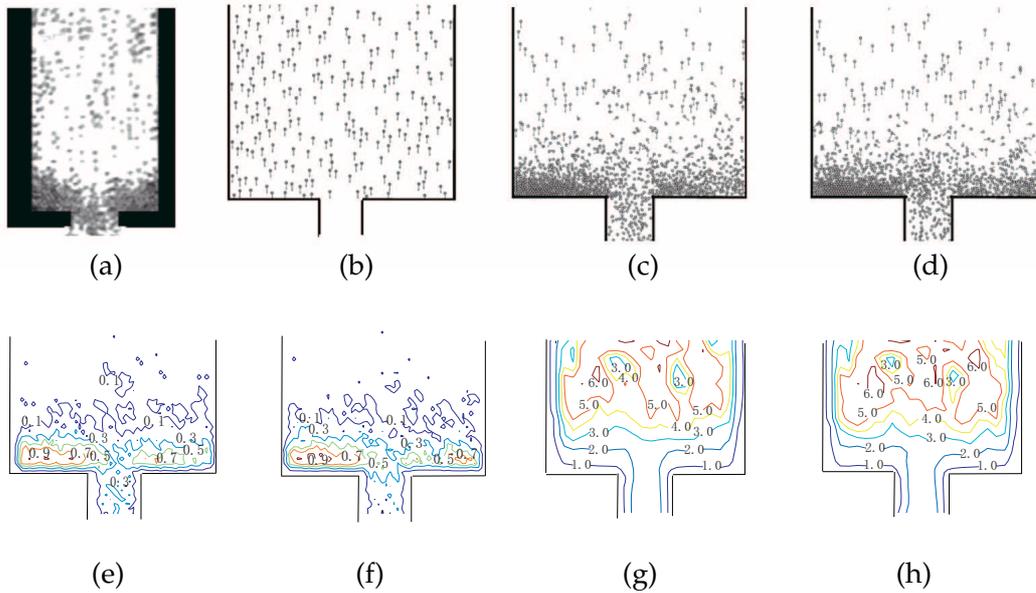


Figure 8: Comparison between the experiment and simulations (only the lower parts are shown) (a) experimental result, (b) initial condition of numerical simulations, (c) instantaneous velocity vector of TDM and (d) instantaneous velocity vector of HAET at  $t=0.02s$ , (e) contour of volume fraction of TDM, (f) contour of volume fraction of HAET, (g) contour of averaged magnitude of velocity of TDM and (h) contour of averaged magnitude of velocity of HAET (m/s).

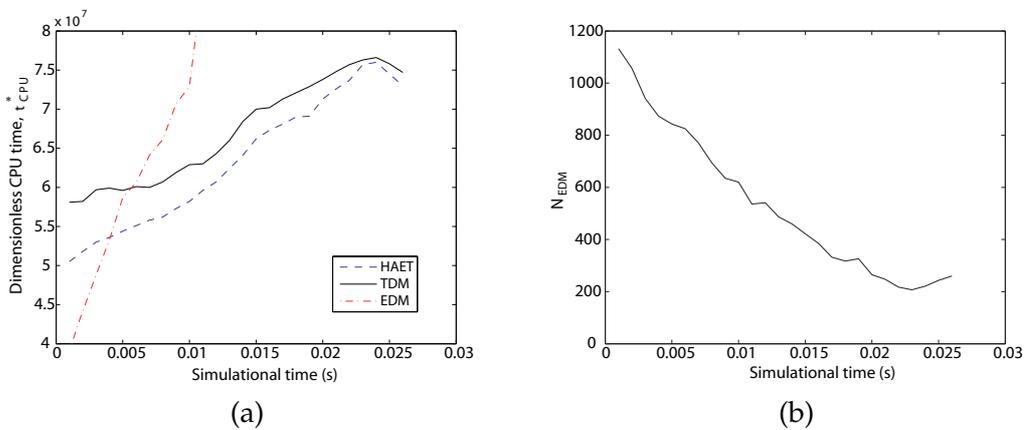


Figure 9: Analysis of CPU time: (a) comparison of dimensionless CPU time  $t_{CPU}^*$  between EDM and HAET; (b) the change of the number of particles integrated using  $dt_{EDM}$  in HAET with the simulation time.

the dimensionless CPU time is defined as the ratio between the CPU time,  $t_{CPU}$ , and the simulation time,  $t$ . These three tests were carried out with SISD (single instruction single data) implementation by the same computer (HP Compaq DC7900 Minitower with dual-core processor E8600@3.3GHz and 4G RAM). The HAET simulation used 15% less CPU

time than the TDM simulation, but used much more than EDM at the very beginning. However, with the formation and growth of the condensed zone, the time step,  $dt_{EDM}$ , decreased steeply and the simulation with EDM had to be stopped at  $t=0.01$ s. The other two curves also increased but at low rate until enough particles entered the chute zone. This is because with the growth of the wedges, more and more collisions occur in these regions and more CPU time is required to reach a solution. Fig. 9(b) shows the number of particles integrated by using the longer time step,  $dt_{EDM}^c$  in HAET. It can be seen that this number,  $N_{EDM}$ , decreases during the whole process when  $t_{CPU}^*$  is increasing. Lastly, although the HAET algorithm can reduce the CPU time in some binary collision and integration processes, more conditional decisions need be done before-hand. Therefore, if  $N_{EDM}$  is too small, a simulation using TDM is faster than that using HAET.

## 8 Conclusions

DEM is a useful tool in the simulation of granular materials. The two main branches of DEM, EDM and TDM, are combined into a hybrid algorithm, which is presented. In this new algorithm, the regions with low volume fraction are simulated with EDM and the others are simulated with TDM. Where the volume fraction is high or low is determined by the link cell method. In addition, a new EDM algorithm based on the link cell method can be used which has a constant lower limit of time step. Compared with the original EDM algorithm, this modified one is simpler and reduces the error. The advantage of the hybrid algorithm is for simulating cases with strongly non-uniform volume fractions distribution, which cannot be solved by EDM. Additionally, it requires less CPU time than TDM does for such cases.

## Acknowledgments

This work is partially supported by a grant from Department of Energy and Process Engineering, Norwegian University of Science and Technology, Institute for Energy Technology (IFE) and SINTEF through the FACE (Multiphase Flow Assurance Innovation Center) project. The authors thank Dr. Angela De Leebeeck from SINTEF, Materials and Chemistry for English polishing.

## Appendix

Consider two spherical particles in a Cartesian coordinate system  $[X, Y, Z]^T$  as shown in Fig. 10(a), each of the particles has 6 degrees of freedom (DOF), three for translation and three for rotation.

A Jacobian  $J$  can be found, which rotates the coordinate system as shown in Fig. 10(b)

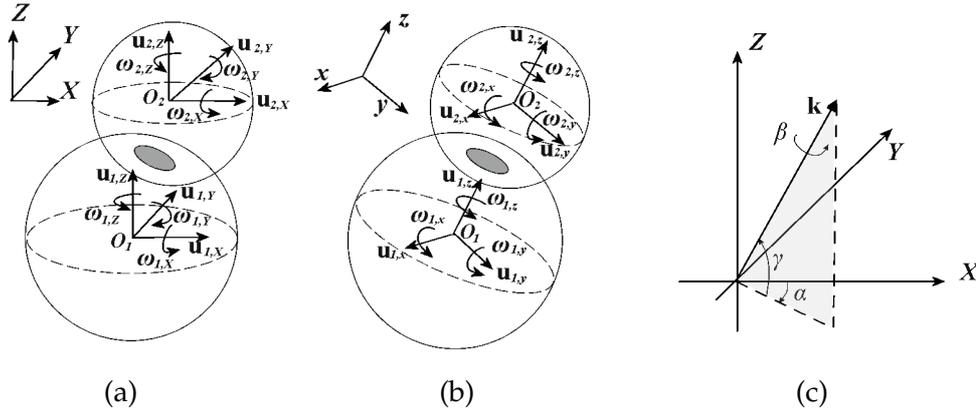


Figure 10: Rotation of coordinate system for binary collision. (a) the coordinates  $[X, Y, Z]^T$  and the velocity components before rotation, (b) the coordinates  $[x, y, z]^T$  and the velocity components after rotation, (c) The Euler angles in  $[X, Y, Z]^T$  coordinate system. The gray areas in (a) and (b) are the contact circles.

and makes the unit vector  $\mathbf{k}$  parallel to the  $z$ -axis [32],

$$\mathbf{J}[\mathbf{k}_X, \mathbf{k}_Y, \mathbf{k}_Z]^T = [0, 0, 1]^T, \quad (\text{A.1})$$

where

$$\mathbf{J} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta \\ 0 & \sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (\text{A.2})$$

$\alpha$ ,  $\beta$  and  $\gamma$  are the Euler angles of the vector  $\mathbf{k}$  in the original coordinate system, as shown in Fig. 10(c). If  $\beta = 0$ , the Jacobian becomes

$$\mathbf{J} = \begin{bmatrix} \cos\gamma\cos\alpha - \sin\gamma\sin\alpha & -\cos\gamma\sin\alpha - \sin\gamma\cos\alpha & 0 \\ \sin\gamma\cos\alpha + \cos\gamma\sin\alpha & -\sin\gamma\sin\alpha + \cos\gamma\cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (\text{A.3})$$

The pre-collisional velocities in the rotated coordinate system are given by

$$[\mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z]^T = \mathbf{J}[\mathbf{u}_X, \mathbf{u}_Y, \mathbf{u}_Z]^T, \quad (\text{A.4a})$$

$$[\boldsymbol{\omega}_x, \boldsymbol{\omega}_y, \boldsymbol{\omega}_z]^T = \mathbf{J}[\boldsymbol{\omega}_X, \boldsymbol{\omega}_Y, \boldsymbol{\omega}_Z]^T. \quad (\text{A.4b})$$

The relative velocity at the contact point is composed of three parts, one in the normal direction (along  $z$ -axis) and the other two in the tangential directions ( $x$ - and  $y$ -axis respectively). The law of momentum conservation in the normal direction ( $z$ -axis) is given by

$$m_1 \mathbf{u}_{1,z} + m_2 \mathbf{u}_{2,z} = m_1 \mathbf{u}'_{1,z} + m_2 \mathbf{u}'_{2,z}. \quad (\text{A.5})$$

Eq. (1.1a) can be rewritten as

$$e = -\frac{\mathbf{u}'_{1,z} - \mathbf{u}'_{2,z}}{\mathbf{u}_{1,z} - \mathbf{u}_{2,z}}. \quad (\text{A.6})$$

Combining Eqs. (A.5) and (A.6) leads to

$$\mathbf{u}'_{1,z} = \frac{m_1 \mathbf{u}_{1,z} + m_2 \mathbf{u}_{2,z} - e m_2 (\mathbf{u}_{1,z} - \mathbf{u}_{2,z})}{m_1 + m_2}, \quad (\text{A.7a})$$

$$\mathbf{u}'_{2,z} = \mathbf{u}'_{1,z} + e (\mathbf{u}_{1,z} - \mathbf{u}_{2,z}). \quad (\text{A.7b})$$

The two components in the tangential directions are independent of each other. Here, only the component in  $x$ -direction is considered. The  $y$ -component can be obtained using a similar method. The shear coefficient of restitution given in Eq. (1.1b) can be rewritten as

$$\beta = \frac{(\mathbf{u}_{1,x} + \Delta \mathbf{u}_{1,x} + \boldsymbol{\omega}_{1,y} \cdot r_1 + \Delta \boldsymbol{\omega}_{1,y} \cdot r_1) - (\mathbf{u}_{2,x} - \Delta \mathbf{u}_{2,x} + \boldsymbol{\omega}_{2,y} \cdot r_2 + \Delta \boldsymbol{\omega}_{2,y} \cdot r_2)}{(\mathbf{u}_{1,x} + \boldsymbol{\omega}_{1,y} \cdot r_1) - (\mathbf{u}_{2,x} - \boldsymbol{\omega}_{2,y} \cdot r_2)}. \quad (\text{A.8})$$

In addition, the change in angular velocity for any sphere  $i$  in the binary system is

$$I_i \Delta \boldsymbol{\omega}_{i,y} = \int_{t_c} (\mathbf{F}_{x,i} r_i) \cdot dt, \quad (\text{A.9})$$

where the moment of inertia is  $I = 0.4 m_i r_i^2$  and  $t_c$  is the collision time. The change in tangential velocity is given by

$$\Delta \mathbf{u}_{x,i} = \frac{\int_{t_c} \mathbf{F}_{x,i} \cdot dt}{m_i}. \quad (\text{A.10})$$

These two equations not only lead to the relationship between the change of angular velocity and the tangential velocity as given by Eq. (3.3), but also the relationship between the two spheres, as follows

$$\frac{\Delta \mathbf{u}_{1,y}}{\Delta \mathbf{u}_{2,y}} = -\frac{m_2}{m_1}, \quad (\text{A.11a})$$

$$\frac{\Delta \boldsymbol{\omega}_{1,x}}{\Delta \boldsymbol{\omega}_{2,x}} = \frac{I_2 r_1}{I_1 r_2}. \quad (\text{A.11b})$$

Thus,  $\Delta \boldsymbol{\omega}_{1,y}$ ,  $\Delta \boldsymbol{\omega}_{2,y}$ ,  $\Delta \mathbf{u}_{1,x}$  can be written as functions of  $\Delta \mathbf{u}_{2,x}$ . Substituting all of these four terms into Eq. (A.8) gives  $\Delta \mathbf{u}_{2,x}$ , as

$$\Delta \mathbf{u}_{2,x} = -(1 + \beta) \frac{\mathbf{u}_{2,x} + \boldsymbol{\omega}_{2,y} \cdot r_2 - \mathbf{u}_{1,x} - \boldsymbol{\omega}_{1,y} \cdot r_1}{1.5 + 3.5 m_2 / m_1}. \quad (\text{A.12})$$

The post-collisional velocity components are given by

$$\mathbf{u}'_{1,x} = \mathbf{u}_{1,x} - \Delta\mathbf{u}_{1,x}, \quad (\text{A.13a})$$

$$\mathbf{u}'_{2,x} = \mathbf{u}_{2,x} + \Delta\mathbf{u}_{2,x}, \quad (\text{A.13b})$$

$$\boldsymbol{\omega}'_{1,y} = \boldsymbol{\omega}_{1,y} - \Delta\boldsymbol{\omega}_{1,y}, \quad (\text{A.13c})$$

$$\boldsymbol{\omega}'_{2,y} = \boldsymbol{\omega}_{2,y} - \Delta\boldsymbol{\omega}_{2,y}, \quad (\text{A.13d})$$

where  $\Delta\boldsymbol{\omega}_{1,y}$ ,  $\Delta\boldsymbol{\omega}_{2,y}$  and  $\Delta\mathbf{u}_{1,x}$  can be obtained from Eqs. (A.11a) and (3.3). Substituting these post-collisional velocities and negative the Euler angles shown in Fig. 10(c). into the coordinate rotating equation (Eq. (A.4b)), the corresponding post-collisional velocities in the original Cartesian coordinate system  $[X, Y, Z]^T$  can be obtained.

## References

- [1] H. M. Jaeger, S. R. Nagar and R. P. Behringer, Granular solids, liquids, and gases, *Rev. Mod. Phys.*, 68 (1996), 1259–1273.
- [2] Y. Forterre and O. Pouliquen, Flows of dense granular media, *Annu. Rev. Fluid Mech.*, 40 (2008), 1–24.
- [3] C. Hogue and D. Newland, Efficient computer simulation of moving granular particles, *Power Tech.*, 78 (1994), 51–66.
- [4] A. Dziugys and B. Peters, An approach to simulate the motion of spherical and non-spherical fuel particles in combustion chambers, *Granular Matter*, 3 (2001), 231–265.
- [5] P. C. Johnson and R. Jackson, Frictional-collision constitutive relations for granular materials with application relations to plane shearing, *J. Fluid Mech.*, 176 (1987), 67–93.
- [6] C. S. Campbell, Rapid granular flow, *Annu. Rev. Fluid Mech.*, 22 (1990), 57–92.
- [7] J. W. Parram, M. S. Wertheim, J. L. Lebowitz and G. O. Williams, Monte Carlo simulations of hard spheroids, *Chem. Phys. Lett.*, 105 (1984), 277–280.
- [8] P. A. Cundall and O. D. L. Strack, A discrete numerical model for granular assemblies, *Geotechnique*, 29 (1979), 47–65.
- [9] R. Ramirez, T. Pöschel, N. V. Brilliantov and T. Schwager, Coefficient of restitution of colliding viscoelastic spheres, *Phys. Rev. E*, 60 (1999), 4465–4472.
- [10] H. Hertz, Über die Berührung fester elastischer Körper, *J. Reine Angew. Math.*, 92 (1882), 156–171.
- [11] K. L. Johnson, *Contact Mechanics*, Cambridge University, 1982.
- [12] G. Kuwabara and K. Kono, Resitution coefficient in a collision between two spheres, *Jpn. J. Appl. Phys.*, 26 (1987), 1230–1233.
- [13] R. D. Mindlin, Compliance of elastic bodies in contact, *J. Appl. Mech.*, 16 (1949), 259–268.
- [14] Y. J. Huang, C. K. Chan and P. Zamankhan, Granular jet impingement on a fixed target, *Phys. Rev. E*, 82(3) (2010), 031307.
- [15] S. Ji and H. H. Shen, Effect of contact force models on granular flow dynamics, *J. Eng. Mech.*, 11 (2006), 1252–1259.
- [16] H. H. Shen and B. Sankaran, Internal length and time scales in a simple shear granular flow, *Phys. Rev. E*, 70 (2004), 051308.
- [17] M. Babic, H. H. Shen and H. T. Shen, The stress tensor in granular shear flows of uniform, deformable disks at high solids concentrations, *J. Fluid Mech.*, 219 (1990), 81–118.

- [18] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Oxford University, 1987.
- [19] P. Zamankhan and J. Huang, Complex flow dynamics in dense granular flow, part II: simulation, *J. Appl. Mech.*, 74 (2007), 691–702.
- [20] C. K. K. Lun and A. A. Bent, Numerical simulation of inelastic frictional spheres in simple shear flow, *J. Fluid Mech.*, 258 (1994), 335–353.
- [21] N. Maw, *A Theoretical and Experimental Investigation into the Impact and Rebound of Elastic Bodies*, PhD thesis, Sunderland Polytechnic U.K, 1976.
- [22] P. Zamankhan and M. H. Bordbar, Complex flow dynamics in dense granular flow, part I: experiment, *J. Appl. Mech.*, 73 (2006), 648–657.
- [23] W. Goldsmith, *Impact: The Theory and Physical Behavior of Colliding Solid*, Edward Arnold, 1960.
- [24] L. Verlet, Computer experiments on classical fluids I: thermodynamical properties of Lennard-Jones molecules, *Phys. Rev.*, 159 (1967), 98–103.
- [25] B. Quentrec and C. Brot, The potential calculation and some application, *J. Comput. Phys.*, 13 (1975), 430–432.
- [26] Y. Amarouchene, J. F. Boudet and H. Kellay, Dynamics and sand dunes, *Phys. Rev. Lett.*, 86 (2001), 4286–4289.
- [27] V. Buchholtz and T. Poschel, Interaction of a granular stream with an obstacle, *Granular Matter*, 1 (1998), 33–41.
- [28] C. R. Wassgren, J. A. Cordova, R. Zenit and A. Karion, Dilute granular flow around an immersed cylinder, *Phys. Fluids*, 15 (2003), 3318–3330.
- [29] R. M. Nedderman, *Statics and Kinematics of Granular Materials*, Cambridge University Press, London, 1992.
- [30] M. Hou, W. Chen, T. Zhang, K. Lu and C. K. Chan, Global nature of dilute-to-dense transition of granular flow in a 2-D Channel, *Phys. Rev. Lett.*, 91 (2003), 204301.
- [31] G. D. R. Midi, On dense granular flows, *Eur. Phys. J. E*, 14 (2004), 341–365.
- [32] G. B. Arfken and H. J. Weber, *Mathematical Methods for Physicists (International Edition)*, 6 Ed., Elsevier, 2005.