Final presentation
Bachelor project

« Two-dimensional granular shear model »

Jean NAFTALSKI

Summary

I- Hertz-Mindlin contact model

II- Equations of motion

III- Velocity Verlet algorithm

IV- Choice of Δt and problem adjustment

V- 1D model

1- normal collision

2- energy monitoring

VI- 2D model

1- shear model

2- energy monitoring

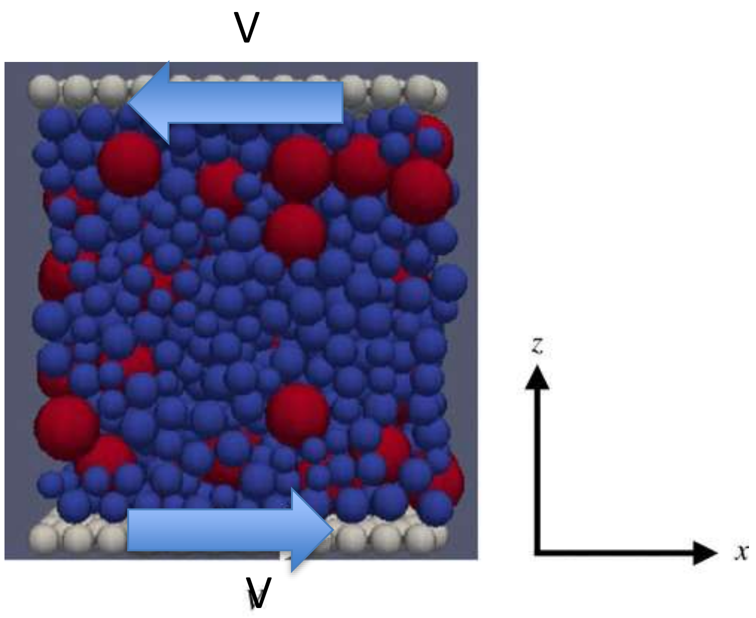
3- avoid instabilities

4- improve the model

Introduction

<u>The experiment</u>

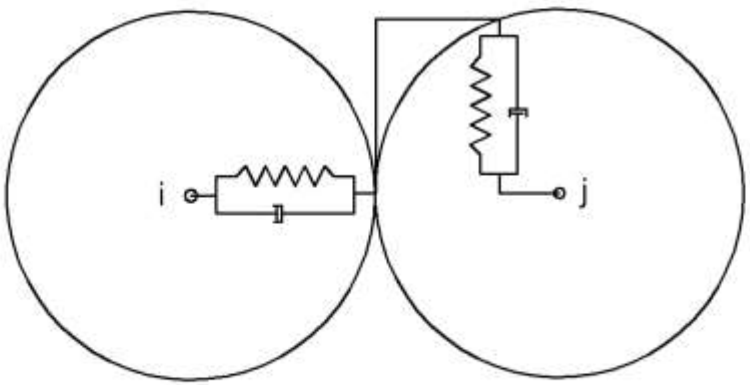
- Small grains confined between two walls
- Upper and lower walls displaced at constant velocity V
- Boundary conditions:
  - periodic in the x direction
  - compression and shear at the walls
- Non-conservative system



<u>The approach:</u> discrete element method (DEM)

- numerical methods for computing the motion of a large number of particles

Hertz-Mindlin contact model



- Normal force:

$$F_n = -k_n \delta_n - \gamma_n \frac{\partial \delta_n}{\partial t}$$

- Tangential force:

$$F_t = -k_t \delta_t - \gamma_t \frac{\partial \delta_t}{\partial t}$$

Coulomb's friction law:

$$|\hat{F}_t| < \mu F_n \longrightarrow F_t = -k_t \delta_t - \gamma_t \frac{\partial \delta_t}{\partial t}$$

$$|\hat{F}_t| \geqslant \mu F_n \longrightarrow F_t = sgn(\hat{F}_t) \mu F_n$$

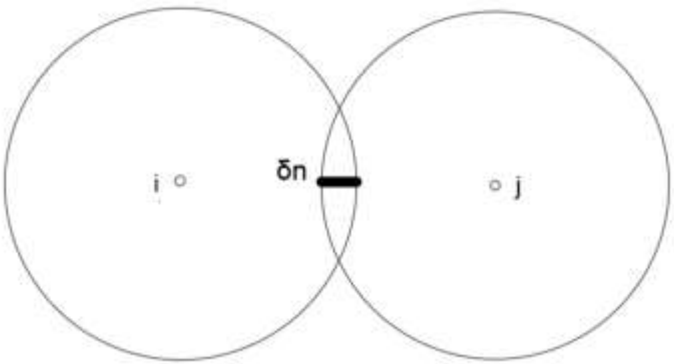
K_n := stiffness of the normal spring
K_t := stiffness of the tangential spring
γ_n := damping constant
γ_t := damping constant

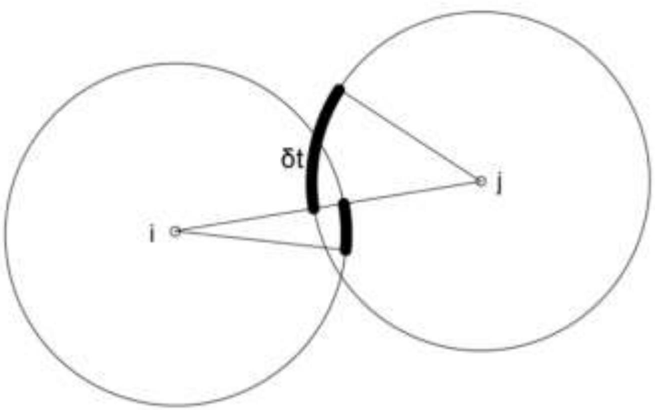
μ := friction coefficient

Hertz-Mindlin contact model

$$k_n = \frac{4}{3} E^* \sqrt{R^* |\delta_n|}$$

$$k_t = 8G^* \sqrt{R^* |\delta_n|}$$





$$\gamma_n = -2\sqrt{\frac{5}{6}} \beta \sqrt{S_n m^*} \geqslant 0$$

$$\gamma_t = -2\sqrt{\frac{5}{6}} \beta \sqrt{S_t m^*} \geqslant 0$$

$$S_n = 2E^* \sqrt{R^* |\delta_n|}$$

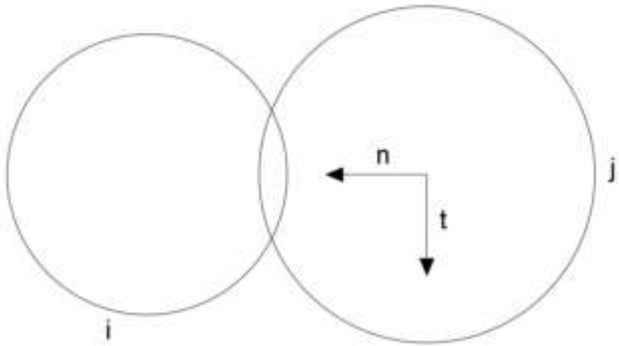
$$S_t = 8G^* \sqrt{R^* |\delta_n|}$$

$$\beta = \frac{\ln e}{\sqrt{\ln^2 e + \pi^2}}$$

<u>For my simulation</u>*:* use of the Y,G,u,e of glass beads

Force and equations of motion

- Total force on a particle *i* : $$F_i = \sum_{i \neq j} F_{ij}$$



with $$F_{ij} = F_n \mathbf{n} + F_t \mathbf{t}$$

- Newton's second law :

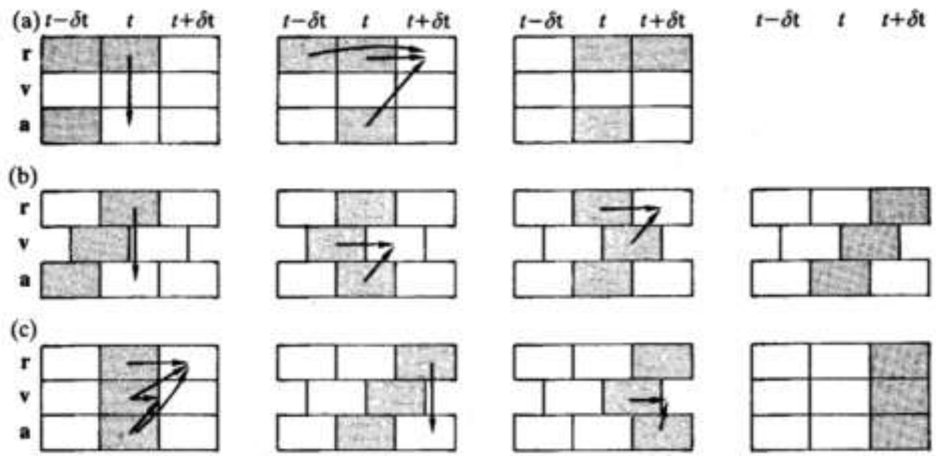
$$\begin{cases} m_i \dfrac{dv}{dt} = \sum_{i \neq j} F_{ij} \\ I_i \dfrac{d\omega}{dt} = \sum_{i \neq j} M_{ij} \end{cases}$$

where $$M_{ij} = -(a_i \mathbf{n}) \times (F_t \mathbf{t})$$

a_i := radius particle *i*

Velocity Verlet algorithm

1. Calculate $\vec{x}(t+\Delta t) = \vec{x}(t) + \vec{v}(t)\,\Delta t + \frac{1}{2}\,\vec{a}(t)\,\Delta t^2$.
2. Derive $\vec{a}(t+\Delta t)$ from the interaction potential using $\vec{x}(t+\Delta t)$.
3. Calculate $\vec{v}(t+\Delta t) = \vec{v}(t) + \frac{1}{2}\,\big(\vec{a}(t) + \vec{a}(t+\Delta t)\big)\Delta t$.



Velocity Verlet

Fig. 3.2 Various forms of the Verlet algorithm. (a) Verlet's original method. (b) The leap-frog form. (c) The velocity form. We show successive steps in the implementation of each algorithm. In each case, the stored variables are in grey boxes.

<u>2 uses of Velocity Verlet:</u>

- to update the translation of the particle
- to update the rotation of the particle

Choice of Δt and problem adjustment

$$\Delta t \sim 10\% * 2 * \sqrt{\frac{\min_i(m_i)}{\max_i(k_i)}}$$

with

$$\begin{cases} \min_i(m_i) = \textit{smallest mass} \\ \max_i(k_i) := \textit{maximum stiffness} \end{cases}$$

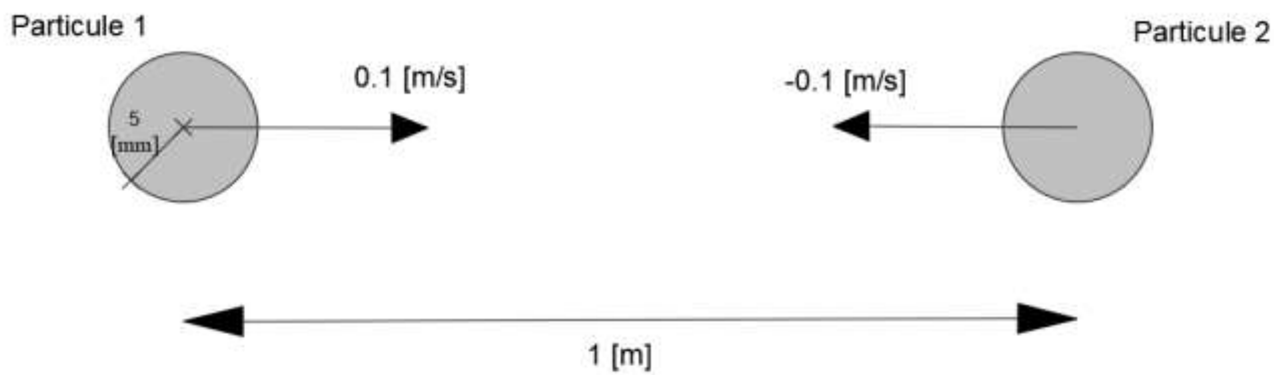
↓

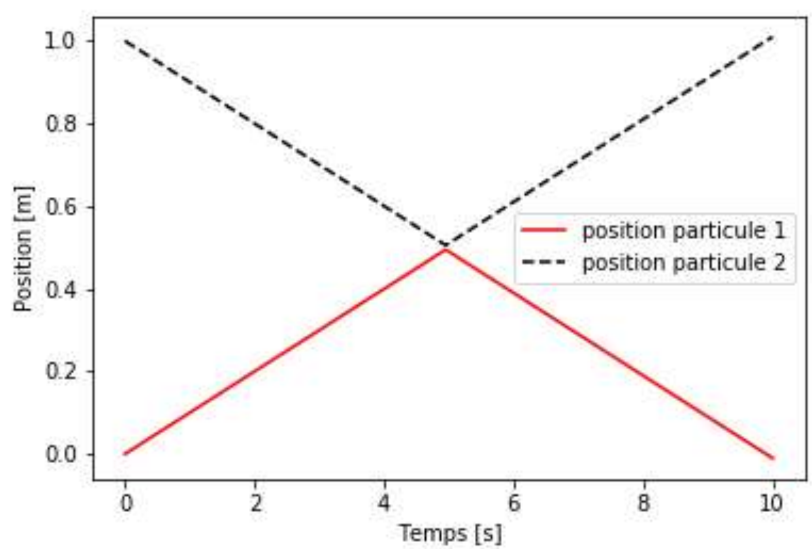
High CPU time with Hertzian contact

↓

Set k_n and k_t so that the
simulation can run on reasonable time

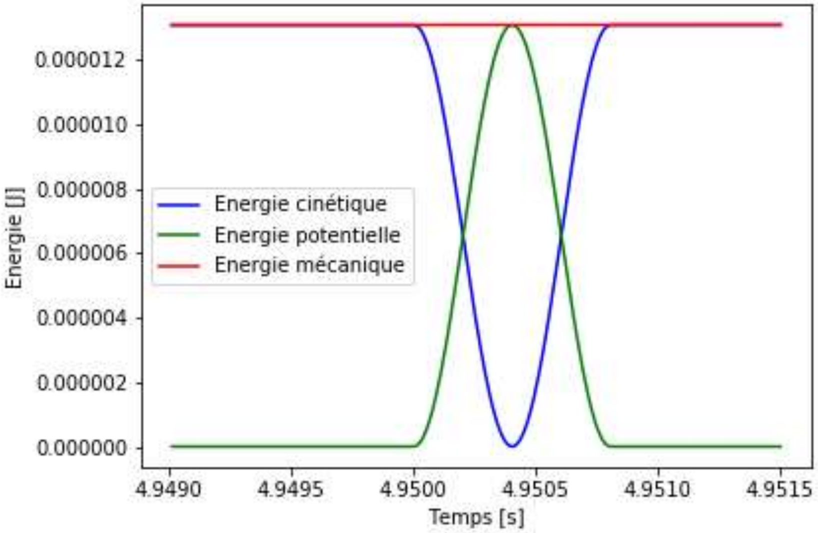
1D model: normal collision between two particles



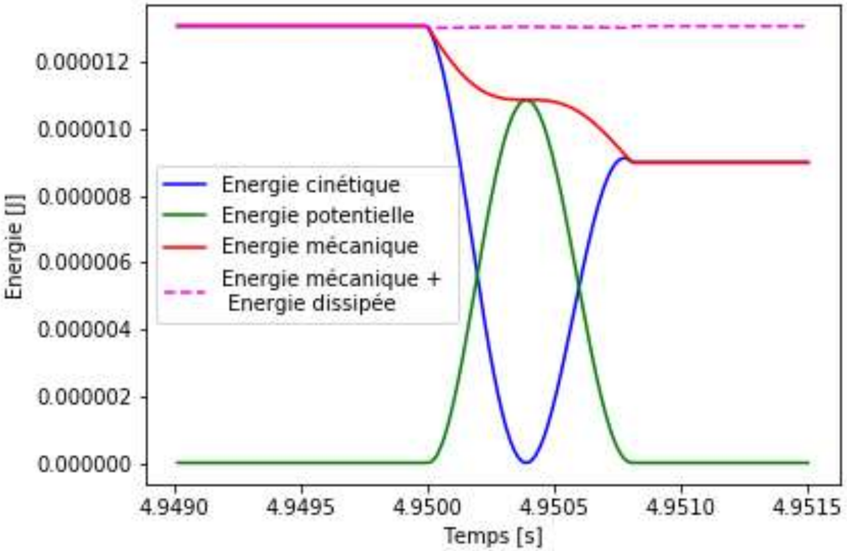


- Conservation of the total kinetic energy
- Conservation of the total momentum

Energy monitoring during the collision

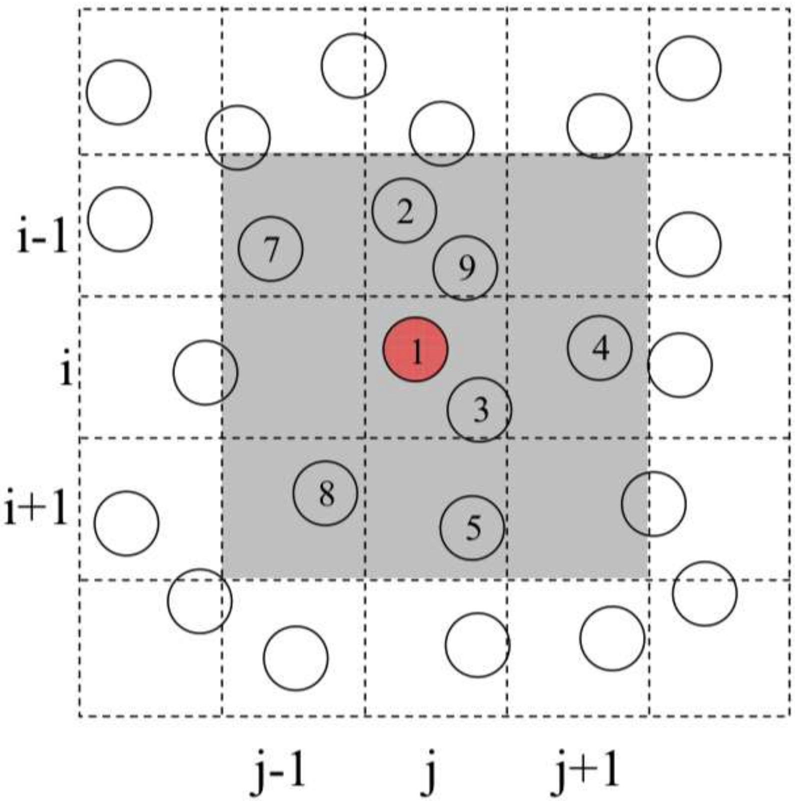


Energy during elastic collision



Energy during inelastic collision

2D model: Neighboring Cell

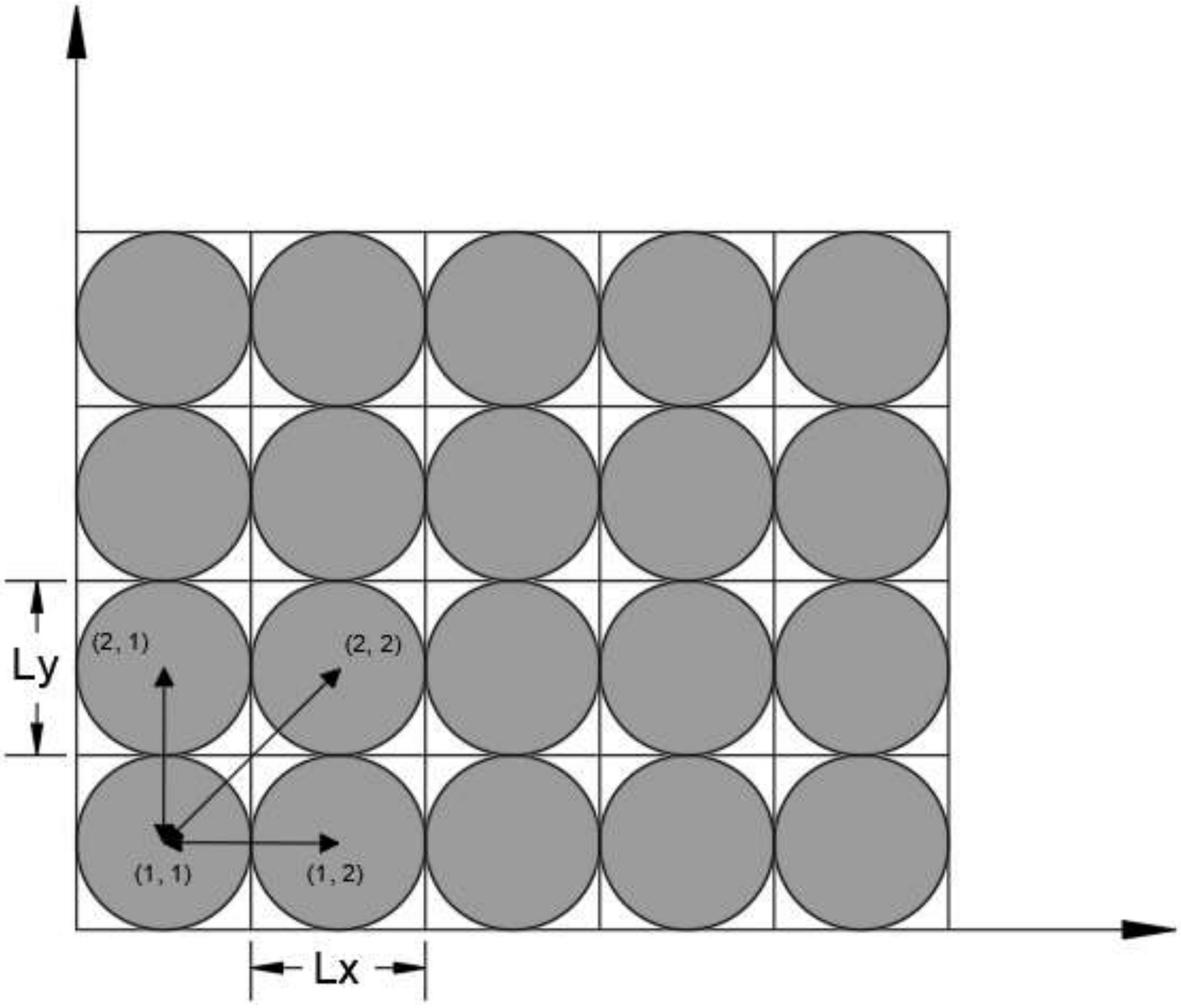


- divide the workspace into a grid of cells
- for each cell, maintain a list of the particles contained within that cell
- for a given particle, only check for contact with other particles in its own cell and neighboring cells

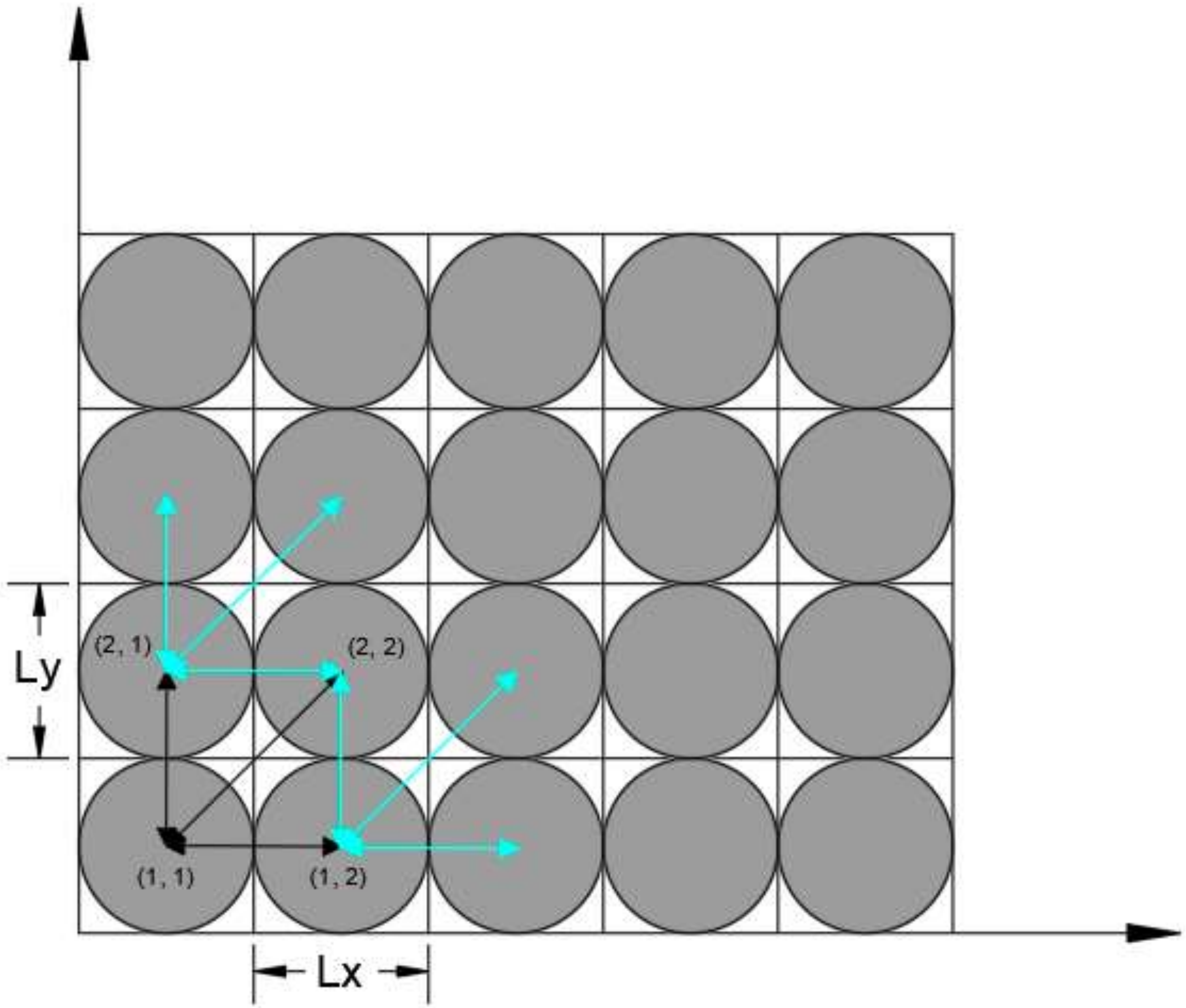
[1]

[1] DEM Modeling: Lecture 11, Coarse Contact Detection
C. Wassgren, Purdue University

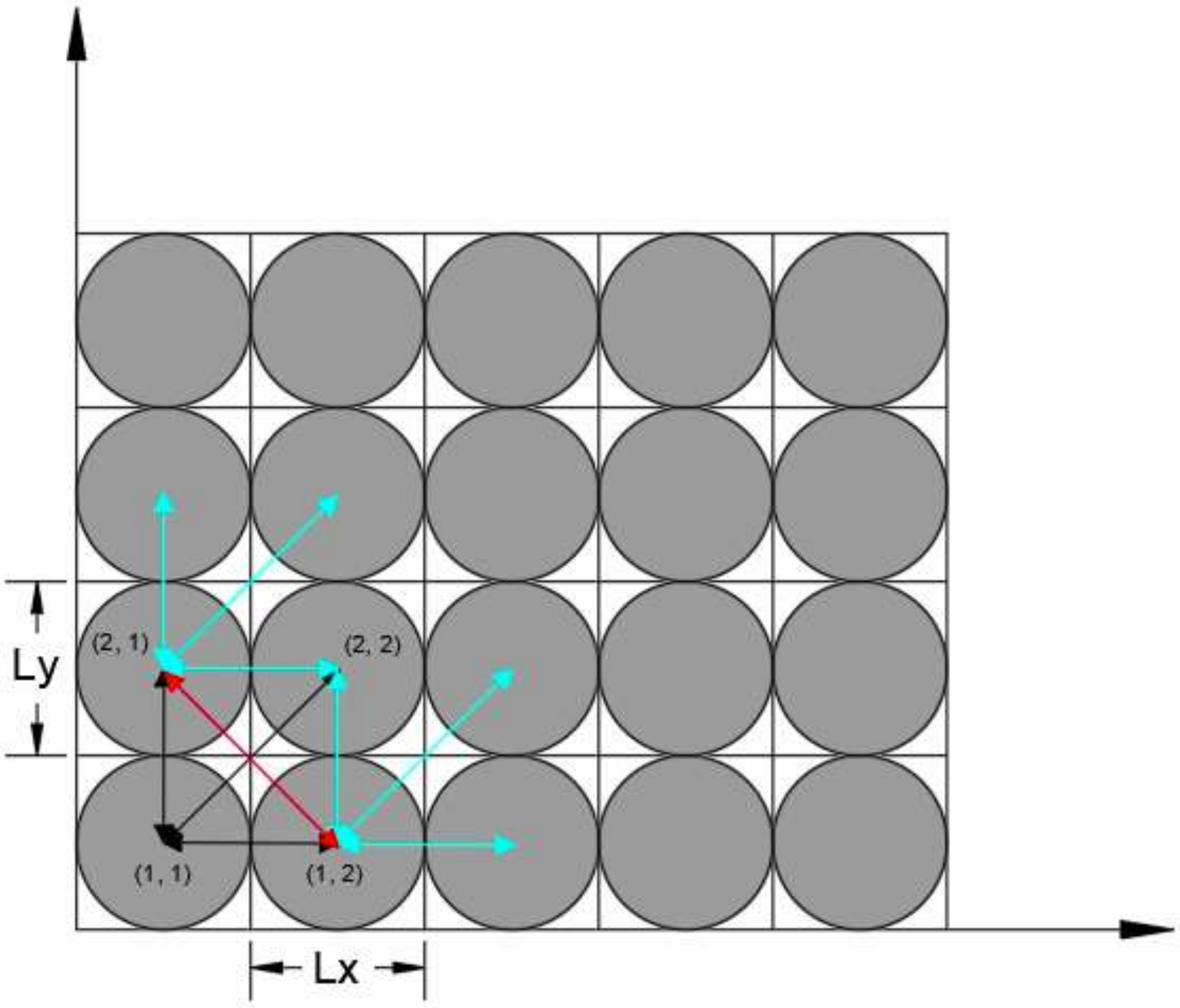
2D model: Neighboring Cell implementation



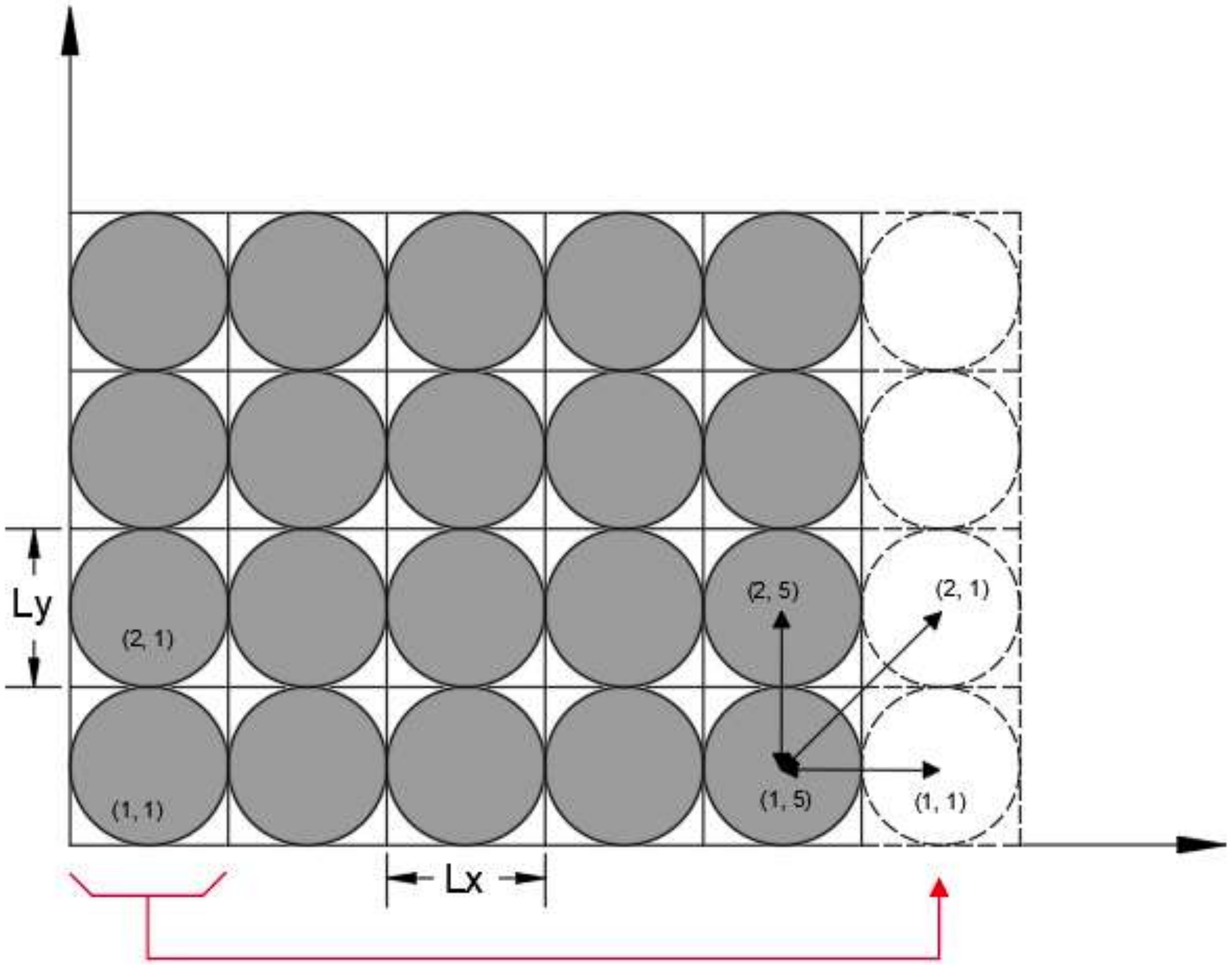
2D model: Neighboring Cell implementation



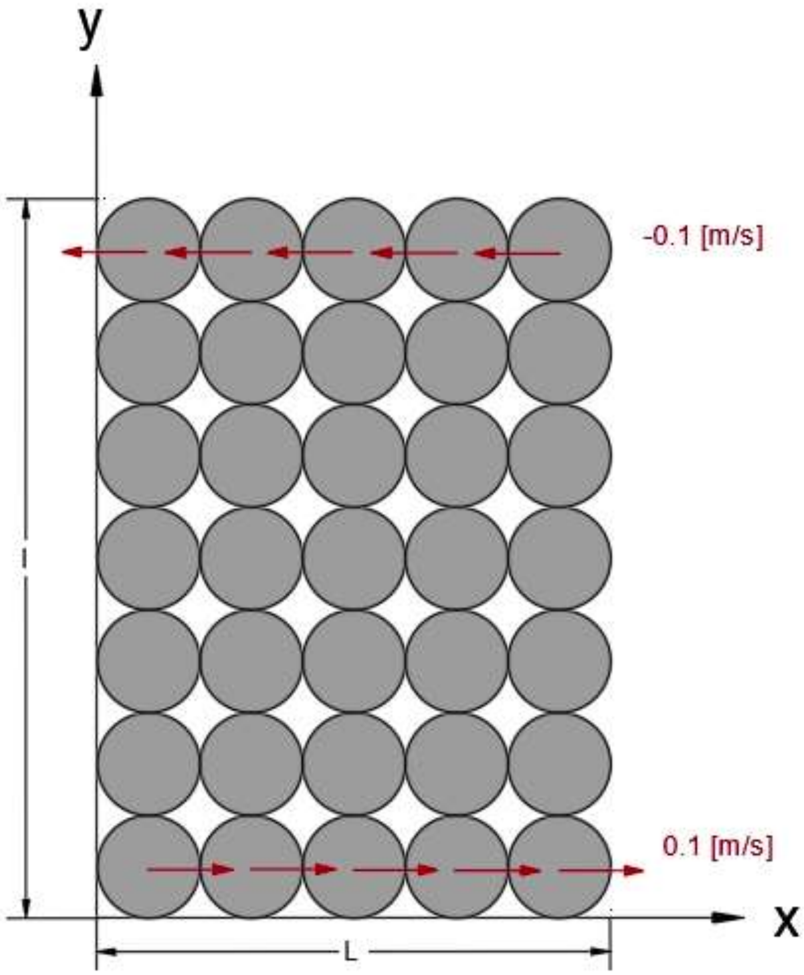
2D model: Neighboring Cell implementation

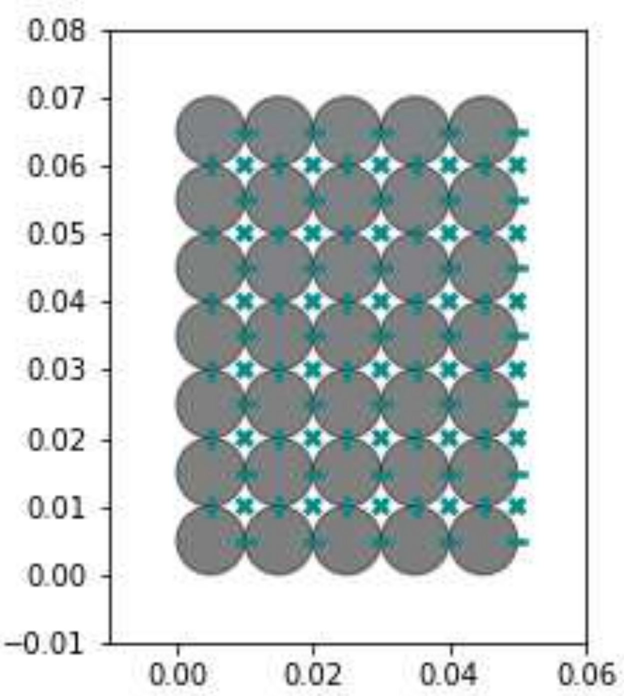


2D model: Periodic boundary conditions

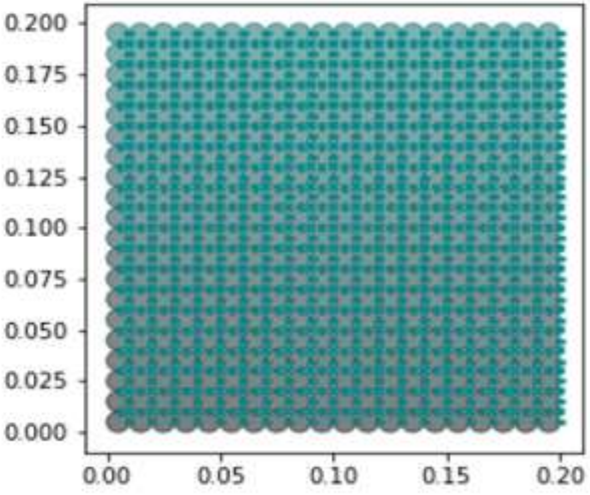


2D experiment: granular shear model

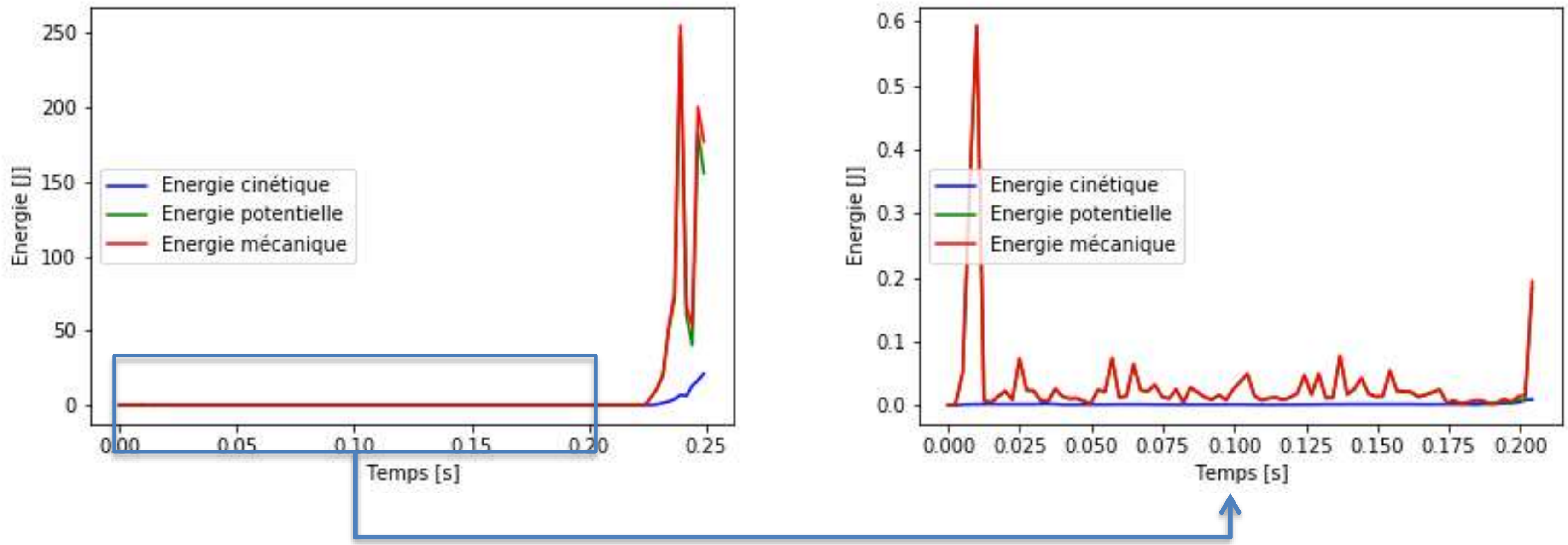




2D experiment: granular shear model



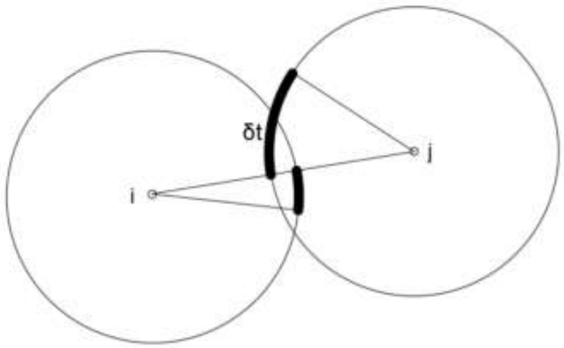
2D model: Energy monitoring



zoom before instability

2D model: how to avoid instability

- Decrease Δt ? Not necessarily the smartest solution
- Different implementation of the tangential overlap



- General damping of the system

2D model: how to improve the model

- Add more randomness to the system

- Implement a 3D model

  → 3 additional equations (translation along z and rotation along x and y)

- Interesting to test other contact models

Thank you for your attention !