Particle-Based Simulations — Assignment 4 (2025)

Discrete Element Method (DEM) — Collapse of a Granular Column

Context and goals

In this final assignment, you will study the collapse of a granular column using the Discrete Element Method (DEM). You will extend the provided DEM code to simulate the dynamics of granular materials and compare your results to experimental and literature data. The goal of this assignment is to understand how microscopic particle properties influence macroscopic flow and pile formation.

Particle and wall properties

The simulation involves spherical particles that interact via soft-sphere contact forces including tangential friction, restitution, and damping. Use the following parameters:

	Normal restitution	**Tangential restitution**	**Friction coefficient**
Particle-particle	0.96	0.33	0.40
Particle-wall (bottom)	0.86	0.33	0.90
Particle-wall (cylinder)	0.86	0.33	0.15

- Particle diameter: $d_p = 2.3$ mm
- Material density: $\rho_p = 2500$ kg/m^3

Code and visualization

The provided DEM code has a similar structure to the MD code used in earlier assignments. The key differences are:

- A *collision list* is constructed from the neighbor list to store the tangential displacement of collision pairs (Cundall–Strack model).
- Angular velocities of particles are computed and updated.
- A restart functionality allows saving and continuing simulations from a stored state.

The output is written in the `.xyz` format. The fifth column represents particle radius, and the sixth column represents the particle velocity magnitude. Use **OVITO** for visualization and color-code particles by velocity magnitude (*Add modification → Color Coding → Input property: Velocity Magnitude*).

Tasks

A. Verification

A1) **Single particle collision:** Simulate a single particle impacting a wall at various velocities. Determine the effective normal restitution coefficient and compare it with the tabulated particle–wall value.

A2) **Two-particle collision:** Perform simulations of two colliding particles. Determine the normal restitution coefficient from the simulations and verify agreement with the particle–particle interaction value.

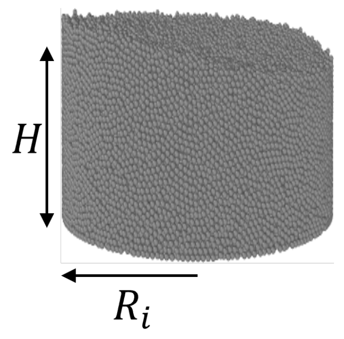
B. Validation: Granular slope stability

B1) Create an initial configuration with a stationary bed of particles several layers thick resting on a bottom wall.

B2) Implement a sloped wall by rotating the gravitational acceleration vector (rather than the wall).

B3) Observe and discuss how the slope angle influences particle motion and stability. Are the results consistent with expectations?

The main task of this assignment is to simulate the collapse of a granular column as shown in Figure 1. The column is initially confined by a cylindrical wall, which is removed at the start of the simulation. The setup is inspired by the study of Lajeunesse et al. (Phys. Fluids 16, 2371–2381, 2004), which analyzes the spreading of a granular mass on a horizontal plane.



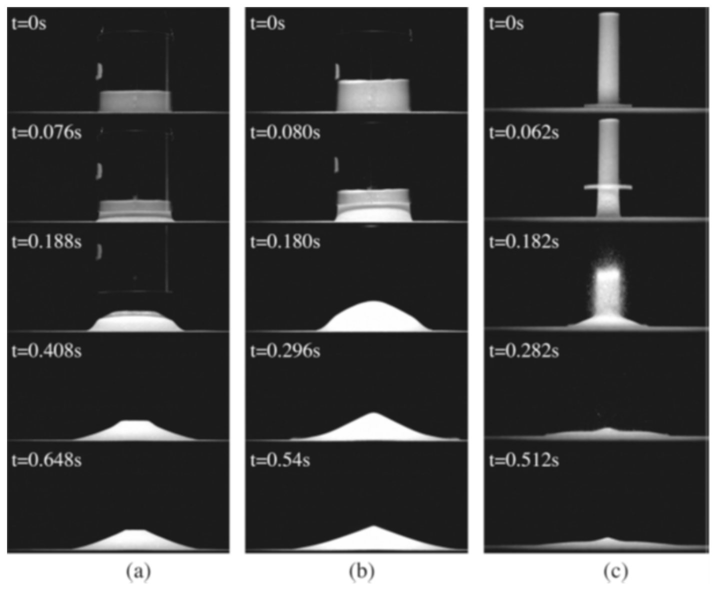


Figure 1: (a) Schematic setup of the granular column used in the DEM simulations. (b) Experimental image sequences of granular column collapse for glass beads (d_p = 350 μm) on a sandpaper substrate: (a) $H/R_i = 0.56$, $R_i = 70.5$ mm; (b) $H/R_i = 0.8$, $R_i = 70.5$ mm; (c) $H/R_i = 5.4$, $R_i = 28.0$ cm (adapted from Lajeunesse et al., 2004)

Change the code to implement a cylindrical wall. In the initial version of the code an upper and lower solid wall are implemented. See the implementation of these walls in the file `walls.c`. A `wall` function takes a particle position. It returns `true` if the particle has overlap with the wall and `false` otherwise. In case of overlap two additional vectors should be returned (using Vec3D pointers) namely r_{iw} and v_w. Here $r_{iw} = r - r_w$ where r is the center-of-mass position of the particle and r_w the point on the wall closest to the particle. v_w is the local velocity of the wall at r_w, which for a stationary wall is zero. You are asked build such a wall function to model the cylinder and provide these function by means as a function pointer to the `parameters` variable. Have a look in `set_parameters` how this is done.

C. Implementation and setup of a granular column

C1) **Cylindrical wall implementation:** Implement a cylindrical wall function in `walls.c`. The function should:

- Return `true` if a particle overlaps with the wall and `false` otherwise.
- Provide two additional vectors via pointers:
  - $\mathbf{r}_{iw}$: the vector from the particle center to the nearest point on the wall.
  - $\mathbf{v}_w$: the local velocity of the wall (zero for stationary walls).

Register this wall via a function pointer in `set_parameters()`.

C2) **Initialization:** Write a routine to initialize particle positions inside the cylinder. Generate a randomly packed cylinder with an aspect ratio of about 0.8 by:

(i) Initializing particles on a lattice inside a long cylinder.

(ii) Randomizing the configuration with collisions at zero gravity.

(iii) Settling the particles under gravity.

C3) Discuss your choices for cylinder dimensions, particle number, and initialization method.

C4) Provide snapshots showing key stages of the initialization sequence.

C5) Compute and plot the radial and axial solids-volume fraction profiles of the final packing.

D. Collapse of the granular column

D1) Remove the cylindrical wall and simulate the collapse of the granular column. Provide snapshots showing the progression of the collapse.

D2) Characterize the final pile: measure its maximum height, base radius, and slope angle.

D3) Compare your results qualitatively with the findings of Lajeunesse et al. (2004).

Deliverables

1. Report (PDF, max 8 pages + appendix) including:
   - Verification and validation results with figures.
   - Implementation description and plots of packing and collapse behavior.
   - Comparison with literature.
   - A concise *lessons learned* box reflecting on DEM implementation and key insights.
2. Code archive (zip) containing modified `.c` and `.h` files and a `README.md`. No binaries.