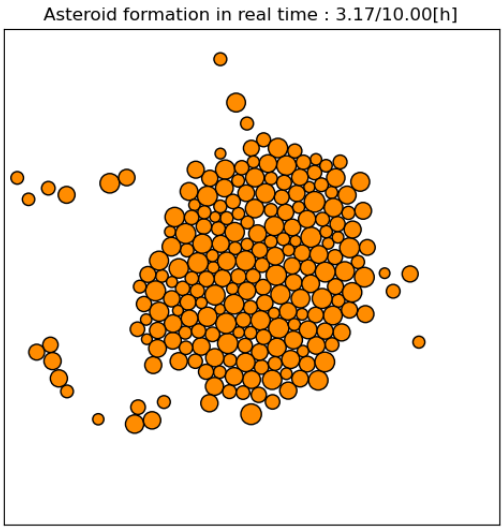
LMECA2300: Homework 2
The formation of an asteroid with DEM

In this second homework, we are going to simulate the formation of an asteroid in outer space with the Discrete Element Method. Micro and meso-scale asteroids and comets consist of rubble piles that aggregated over time because of the gravitational forces acting between them. This process can further lead to the formation of planets, and its modelisation is a key aspect in the understanding of the early formation of our solar system and other similar ones. The explicit and lagrangian nature of the Discrete Element Method allows to include any kind of interaction between the considered grains, which makes it particularly well suited to the study of gravitational aggregation. Since this process can be quite long, even for meter-scale objects, we will use tonnes and hours as units for mass and time for our simulation, instead of their SI counterparts kilograms and seconds.



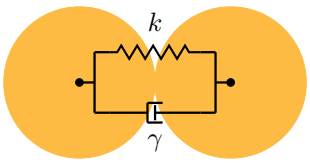
Gravitational forces

Since the theory of general relativity proposed by Einstein, gravity is understood as a curvature in space-time caused by objects with mass. However, for most purposes, Newton's law of universal gravitation is largely sufficient. Gravitational interactions are hence modeled as a force acting upon two bodies that is directly proportional to their masses and inversely proportional to the square of the distance between them. The force acting on a grain i because of a grain j is given as:

$$f_{ij} = -G\frac{m_i m_j}{x_{ij}^3}\mathbf{x}_{ij},$$

with $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ the distance vector between grain i and grain j and G the gravitational constant.

Nonlinear Kelvin-Voigt contact model



The resolution of a contact between particle is based on the explicit computation of a contact force, according to a chosen contact model. In this homework, the selected contact model corresponds to a Kelvin-Voigt representation of the contact, with a viscous dissipative term and a Hertz law for the elastic term:

$$f_n = \min\Big(0, \begin{cases} k|\delta|^{\frac{3}{2}} - \gamma u_n, & \delta < 0 \\ 0, & \delta > 0 \end{cases}\Big)$$

with k a spring constant with appropriate units, γ a viscosity and $u_n = (\mathbf{u}_i - \mathbf{u}_j)\cdot\mathbf{n}$ the normal relative velocity of the two grains.

What you have to do

You are asked to:

1. Write a function

   ```
   x = random_initial_conditions(n,R,xmin,xmax,ymin,ymax)
   ```

   that returns the positions `x` of `n` grains with maximum radius `R` uniformly randomly distributed in a domain delimited by `xmin`,`xmax`,`ymin` and `ymax`.

2. Write a function

   ```
   sRad, iM, mmG = compute_prefactors(rho,G,R)
   ```

   that computes the prefactors used in the other functions. The radii of the grains `R` is an array of shape `n`×`1`, while the grain density `rho` and the gravitational constant `G` are floats. `sRad` has a shape of `n`×`n` and contains the pair-wise sum of the radii of the grains (duplicates included). `iM` is of shape `n`×`1` and contains the inverse mass of each grain. `mmG` is of shape `n`×`n`×`1` and contains the pair-wise product of the masses of the grains (duplicates included) times `G`.

3. Write a function

   ```
   runge_kutta(x,v,sRad,iM,mmG,k,gamma,dt):
   ```

   that updates the positions `x` and velocities `v` of the grains using a fourth order explicit Runge-Kutta scheme.

4. Write a function

   ```
   velocity_verlet(x,v,sRad,iM,mmG,k,gamma,dt):
   ```

   that updates the positions `x` and velocities `v` of the grains using a velocity-verlet scheme.

5. Write a function

   ```
   f = compute_gravity(x,mmG)
   ```

   that computes the gravitational forces on the grains. `f` is an array of shape `n`×`2` and contains the resultant of the gravitational forces exerted by all the other grains on each grain.

6. Write a function

   ```
   f = compute_contacts(x,v,sRad,k,gamma):
   ```

   that computes the contact forces between the grains, given their positions `x` and velocities `v`, the sum of their radii `sRad`, the spring constant `k` and the viscosity `gamma` associated to the nonlinear Kelvin-Voigt contact model with a Hertz law. `f` is an array of shape `n`×`2` and contains the resultant of the contact forces on each grain.

7. Answer the following questions in a separate `README` file:

   - Are the values of the parameters given in the test file relevant compared to real-life cases? Justify.
   - You have to implement two different time schemes. Both are explicit, but what are the main differences between them? What are their advantages or inconvenients?

You have to implement the functions in the file `asteroid.py`. You are also given the script `asteroidTest.py` so that you can test your functions. The test script will produce an animation of your simulation, which can be written in a video file if you uncomment the corresponding line. You should pay attention to writing fast functions : this time, the efficiency of your program will be an important part of the grading. You have until Thursday March 14 23:59 to submit your program `asteroid.py` on the server in a `.zip` file. You may add comments in your code.