

Project programmeren: 4EE11

Project 6: The grain silo

Joris Remmers*

February 9, 2009

Discrete particle model

The motion of grains in a silo can be simulated with a discrete particle model. In this model, the grains are represented by particles with a certain mass. The forces that act on these particles are (i) gravity and (ii) the elastic and viscous contact forces due to the contact with neighboring particles. The new position and velocity of each particles is determined by means of the Verlet explicit time integration scheme.

Mechanical model

Consider the particles in Figure 1. The position and velocity of particle i with a radius R_i is denoted by the vectors \mathbf{x}_i and \mathbf{v}_i respectively. The mass of the particle is equal to m_i . The particle is in contact with a particle j when the particles overlap, or when:

$$|\mathbf{x}_{ij}| < R_i + R_j ,$$

where $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$. In that case, the overlap δ is defined as:

$$\delta = R_i + R_j - |\mathbf{x}_{ij}| .$$

The interaction force that particle i exerts on j can be decomposed into an elastic part \mathbf{F}_{ij}^e and a viscous part \mathbf{F}_{ij}^v . The elastic force is equal to:

$$\mathbf{F}_{ij}^e = k\delta\mathbf{e}_{ij} ,$$

*J.J.C.Remmers@tue.nl

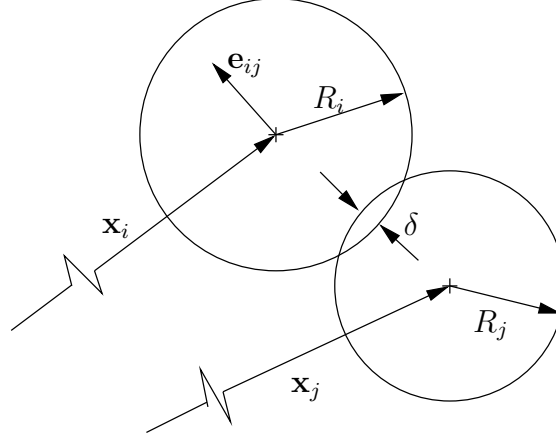


Figure 1: Two interacting particles

where k is a spring constant and \mathbf{e}_{ij} denotes the unit vector that is aligned with the relative position vector, see Figure 1:

$$\mathbf{e}_{ij} = \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|}.$$

The viscous force is a linear function of the relative velocities of the two particles:

$$\mathbf{F}_{ij}^v = -b(\mathbf{v}_i - \mathbf{v}_j),$$

where b is a drag constant. Finally, all particles are subjected to gravity:

$$\mathbf{F}_i^g = -m_i g \mathbf{j}$$

where g is the gravitation constant $g = 9.81 \text{ m/s}^2$ and \mathbf{j} is the unit vector in the positive y direction. Note that all forces act on the center of mass of the particle i , which is located at \mathbf{x}_i .

Solution algorithm

The total system consists of n particles, which have a position \mathbf{x}_i^t , a velocity \mathbf{v}_i^t and an acceleration \mathbf{a}_i^t at time t . The new state of each particle i at time $t + \Delta t$ can be determined by using an explicit Verlet algorithm [1]. For each particle i , the following procedure can be followed:

1. Calculate:

$$\mathbf{x}_i^{t+\Delta t} = \mathbf{x}_i^t + \Delta t \mathbf{v}_i^t + \frac{1}{2} \Delta t^2 \mathbf{a}_i^t;$$

2. Calculate:

$$\mathbf{v}_i^{t+\frac{1}{2}\Delta t} = \mathbf{v}_i^t + \frac{1}{2}\Delta t \mathbf{a}_i^t;$$

3. Determine the total force vector \mathbf{F}_i that is acting on particle i using the positions $\mathbf{x}_i^{t+\Delta t}$ and the velocities $\mathbf{v}_i^{t+\frac{1}{2}\Delta t}$ of all particles in the system.

4. Calculate:

$$\mathbf{a}_i^{t+\Delta t} = \frac{1}{m_i} \mathbf{F}_i$$

5. Calculate:

$$\mathbf{v}_i^{t+\Delta t} = \mathbf{v}_i^{t+\frac{1}{2}\Delta t} + \frac{1}{2}\Delta t \mathbf{a}_i^{t+\Delta t}.$$

The positions, velocities and accelerations are now known at time $t + \Delta t$ and one can proceed to the next time step.

Boundary conditions

The walls of the silo are represented by particles that have a fixed position, i.e. their velocity is always equal to zero. Note that although these particles do not move, they still interact with the particles inside the container.

The positions of the particles that describe the geometry of the silo are given in the file `silol.dat`. The first few lines of this file look as follows:

```
183 5
-1.300e+02 4.000e+02 3.000e+00
-1.300e+02 3.950e+02 3.000e+00
-1.300e+02 3.900e+02 3.000e+00
-1.300e+02 3.850e+02 3.000e+00
...
...
```

The integers in the first line denote the number of particles that represent the fixed wall and the number of particles that represent the door of the silo, respectively. The next lines contain three floats, representing the x and y coordinates of a particle and its radius, respectively. In this specific example, the first 183 lines describe the position and radius of the particles that represent the wall of the silo. The following 5 lines the particles that form the door.

Visualization

The data can be visualised using the Scalable Vector Graphics format [2]. SVG is a language for describing two-dimensional graphics and graphical applications in XML. By writing the data in a relatively simple ascii text file, the positions of the particles can be shown graphically in any viewing program or a web browser such as Windows Explorer. An example of a file that shows two circles is given here:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
    "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg width="600px" height="800px" version="1.1"
    xmlns="http://www.w3.org/2000/svg">
  <circle cx="400" cy="50" r="10" fill="red"/>
  <circle cx="210" cy="150" r="30" fill="green"/>
  <circle cx="400" cy="550" r="60" fill="blue"/>
  <circle cx="110" cy="650" r="30" fill="black"/>
</svg>
```

Simulation

In this case, we will simulate the flow of particles when they are added to a silo and the subsequent flow when the door in the bottom is opened. The dimensions of the silo, which is shown in Figure 2 are roughly 2.60 m wide and 4.0 m high (the exact dimensions follow from the input file with the silo walls). The radii of the particles that will be dropped into the silo may vary from 3.0 to 5.0 cm. The mass of the particles is equal to $m_i = \rho R_i^2$ where ρ is a density parameter. In this specific case, we take $\rho = 100 \text{ kg/m}^2$. Every 0.05 seconds, a new particle will be released from the point A, in Figure 2. The initial velocity of these particles is $\mathbf{v}_i^0 = (0, -2.0) \text{ m/s}$. The elastic constant for the particle interaction force k is taken to be $k = 100 \text{ N/m}$ and the drag constant $b = 0.05 \text{ Ns/m}$. The time step that may be used in the simulations is $\Delta t = 0.001 \text{ s}$.

Further assumptions are:

- The maximum number of particles in the simulation (including the particles that model the walls of the silo) will never exceed 2500.
- Particles that have left the silo through the opening at the bottom

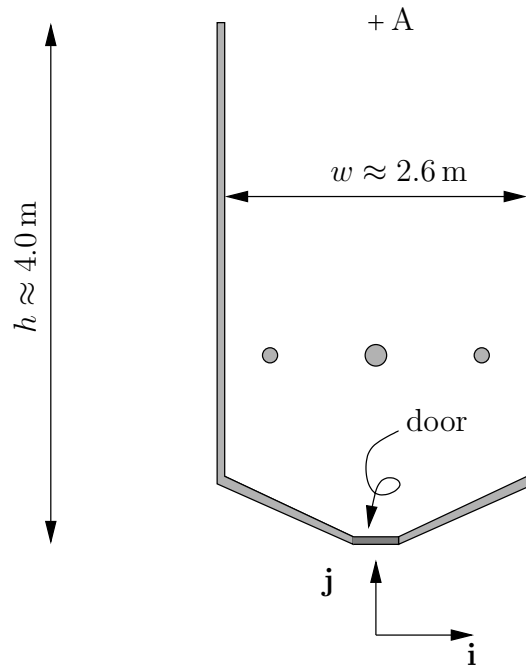


Figure 2: Geometry of the silo. The silo contains some internal obstacles to deviate the trajectories of the grains.

and which have crossed the line $y = -1.0$ m, may be removed from the simulation in order to speed-up the simulations.

Assignments

Week 1

1. Describe the data structure you will use to store the relevant data (such as positions etc.) of all particles. Give the C code for this data structure.

Week 2

2. Write a code that reads the positions of particles from the input file into the data structure as described in assignment 1. Add a routine to this code that writes these positions to an output file that plots the particles in SVG format. Plot the fixed wall particles as red circles

and the particles that represent the door as green circles. Hint: you may need to scale and translate the particle positions to get a nice svg-picture. You can find these instructions at the SVG specification website [2].

3. Implement the Verlet algorithm to calculate the trajectory of a single particle under gravity. Do not consider the interaction forces for the time being. Design a test to check if the implementation is correct. Describe this test and its results in the report.

Week 3

4. Finish the code by adding the following items:
 - A routine that releases new particles from point A with the correct initial velocity.
 - A routine that calculates the interaction forces.
 - A routine that plots the state of the system in an SVG file for every second (i.e. every 1000 steps).

One can imagine that the routine in which the particle interaction forces are calculated is the most time consuming part of the code. Try to reduce the number of calculations in this part of much as possible, without losing accuracy. Indicate what 'programming-tricks' you have been using to do this.

5. Simulate the flow of particles in the silo when 1000 particles are added with a time interval of 0.05 seconds. Consider the following two cases:
 - The silo is filled with particles that all have a uniform radius $R_i = 0.04$ m.
 - The silo is filled with a 50%-50% mixture of particles with radii $R_i = 0.04$ m and $R_i = 0.03$ m.

What is the difference between the two results?

6. Implement a routine to determine whether all particles that have been added to the silo have come to rest. At this instance, the door can be opened to let the particles flow out. What kind of routine can be used to open the door?

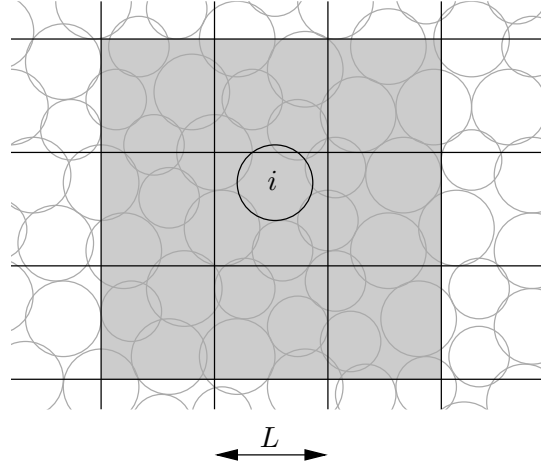


Figure 3: Division of the domains in cells with length L . Particle i can only interact with particles in the shaded area.

Week 4

You may have noticed that when the number of particles increases, the simulation slows down drastically. This is because of the fact the the algorithm that calculates the interaction forces is a double loop over the total number of particles, which scales with $O(N^2)$ (where N is the number of particles. This means that when the system contains 1000 particles the interaction force algorithm is invoked approximately 10^6 times per time step.

The number of evaluation in particle interaction routine can be reduced significantly when the evaluation of particles that are far away is skipped in this process. This can be done by dividing the domain in a number of square cells with length L , see Figure 3. When the dimension of the cell is sufficiently large, it may be assumed that a particle can only interact with the other particles in the cell and the particles in the neighboring cells (this is done to cover the particles that are on a cell boundary). In this case, the number of evaluations needed to calculate the interaction forces is much smaller.

7. Implement this cell algorithm as a new function in your code. Make sure that you can switch between the old implementation and this new one. Describe the data structure that you use to store the mapping of

particles in the cells¹. Test the algorithm with a cell size $L = 0.25$ m. What is the speed-up of your simulation?

8. What is the minimum cell size for which this technique will still work.

Week 5

Time allotted to finish the assignments from the weeks before and to finish the report.

References

- [1] Verlet-Integration, http://en.wikipedia.org/wiki/Verlet_integration.
- [2] Scalable Vector Graphics, <http://www.w3.org/Graphics/SVG/>.

¹If there is time left, one may consider to store the mapping of the particles on the cells in a so-called chained-hash table.