MIV Project
Session 1

INSA Fifth Year - 2015/2016
Modeling and Engineering for Biology and Health Applications
Maud Marchal, Ferran Argelaguet
December 8, 2015

1 Objectives

The goal of the first session of the project is to discover the physically-based modeling through a Mass-Spring-Damper system (MSD). It is a simple and computationally cheap approach for real-time simulation and interaction with deformable bodies. Different meshes can be simulated, such as fabric/cloth (surface meshes) or organs (volumetric meshes).

The objectives of this first session are:

- discovering the architecture and the behavior of an application allowing the simulation and the rendering of physical models;
- understanding and implementing the different simulation steps of a MSD model, such as the computation of forces and their integration;
- studying the behavior of the simulated objects according to the parameters involved in the simulation;
- observing the limits of the MSD model;
- having fun!

2 Files provided

A partial implementation of a physical simulation using the MSD model and allowing its visual rendering is available on Moodle. The Visual Studio solution is called `ProjectMIV.sln`. The classes and files provided are in the sources repository with the following organization:

- `main.cpp`: controls the flow of the application, using mainly the GLUT library for window and input/output management. Must not be modified.
- `lib_simu`: project that manages the 3D display and the mesh manipulation routines. This project must not be modified.
- `physics`: contains the data structures of the MSD model (i.e. the particle cloud) and, for each particle, its list of neighbors. Must not be modified.
- `Simulator`: deals with the the simulator. These files are the ones that you have to modify today.

The different meshes available are located in the `./media/meshes` folder. You will also find in this folder a Perl script that generates a rectangular mesh representing a piece of cloth. The resolution of this mesh is passed as a parameter to the script, allowing to generate more or less detailed meshes. Between two meshes of different resolutions, the one with a higher resolution will better behave during the simulation, but will be more computationally expensive to simulate.

A mathematics library is available also within the `Maths` namespace, allowing you to manipulate 3D vectors and perform common vector computations (including operator overloading).

3 Session tutorial

3.1 Project exploration

Question 1

Download the files from the Moodle and explore the source code of the `ProjectMIV` solution. Compile and run the solution. By default, the application loads a 10x10 mesh called cloth1.obj.
Which are the main steps of the application main loop in the `main` file. What are their purposes?

In the next questions, you will have to modify the `Simulator` class, starting with the main function `Update` composed of the different steps to be performed at each step of the simulation.

3.2 Physical simulation

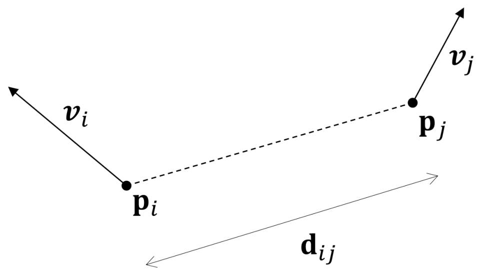

Figure 1: Notations used along the project. Two nodes i and j have a position $\mathbf{p}_i$ and $\mathbf{p}_j$, a velocity $\mathbf{v}_i$ and $\mathbf{v}_j$ and are separated by the distance d_{ij}. Each particle i also stores a force $\mathbf{f}_i$ that will modify its velocity over time.

Question 2

In the particle class, the `force_accumulator` member allows to stock (accumulate) the sum of the forces applied on a particle before the integration step (velocity and position update).
In the function `ComputeForces` of the `Simulator` class, add the gravity force as a new force applied on each particle. What do you observe? Why?

Question 3

A numerical integration needs to be added to proceed the physical simulation. The integration of a particle i computes its new velocity $\mathbf{v}_i^{t+1} = f(\mathbf{v}_i^t, \mathbf{f}_i, \Delta t)$ and its new position $\mathbf{p}_i^{t+1} = f(\mathbf{p}_i^t, \mathbf{v}_i^t, \Delta t)$ from the force applied on it, and from its current position $\mathbf{p}_i^t$ and velocity $\mathbf{v}_i^t$. Implement the integration of forces in the function `Integrate` of the `Simulator` class. What do you observe? Why?

Question 4

Freeze the first row of particles in order to make the mesh hang like a piece of cloth.
How did you proceed?

Question 5

In a MSD model, we consider the attraction between particles as a spring system. The force f_{ij} applied on a particle i by its neighbor j is: $f_{ij} = (\mathbf{p}_i - \mathbf{p}_j) * K * (d_{ij} - d_{ij\_init})$, where K is a spring constant that represents the stiffness of the link.
Add the forces due to the springs between particles in the function `ComputeForces` of the `Simulator` class. What do you observe?

Question 6

K will define the stiffness of the spring. The integration timestep Δt defines the time between each computation.
Try different values for the integration timestep Δt and the spring stiffness K. How does the simulation behave according to the different values? Explain.
Are you able to obtain a stable simulation? Why?

Question 7

The `app_loop` function (it calls 'update' and 'draw') is the main loop function, called automatically by GLUT at given time.
Modify the function `Update` of the Simulator class so that it executes n calls to update with a timestep of $\Delta t/n$, n being a new parameter (see figure 3.2). What is the purpose of this modification?
Hint : Compute and output the update frequency of the physical simulation and the visual rendering, watch also the processor use during the simulation. Compare those values before and after the modification.

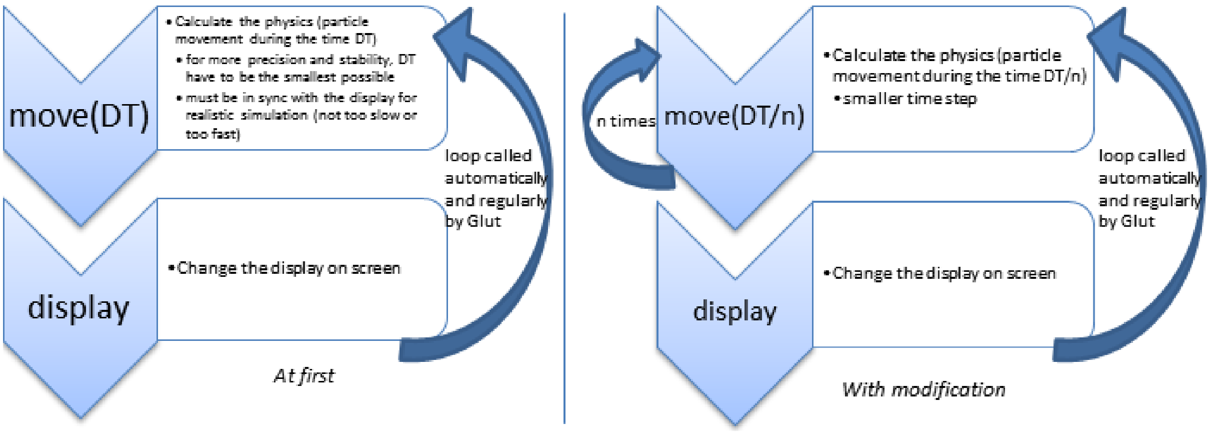

Figure 2: The app_loop function.

Question 8

For a more realistic simulation, we need to add damping forces. Damping forces are forces that counteracts the velocity of the particles.The damping force f_i^d acting on a particle i is defined as: $f_i^d = -D_v * \mathbf{v}_i$ where D_v represents the damping coefficient.
In the function `ApplyVelocityDamping`, add damping forces on the velocity of the particles with a damping coefficient. Do not forget to call this function in the `Update` function. What is the purpose of these additional damping forces?

Question 9

Try different values for the simulation parameters $(\Delta t, K, Dv, n)$.
Give a set of values for which you think the simulation is realistic and stable.

3.3 Conclusion

Question 10

Briefly summarize what are the pros and cons of a Mass-Spring-Damper model when simulating a deformable body. Use what you have found during this session to support your statements.
Keep screenshots from the main steps of your project.