Assignment 3: Mass-Spring Systems and Dynamics

Due: Octobe 12, 11:59pm

Introduction

In this assignment you will implement a simple mass-spring system which will be time integrated using a linearly implicit Euler integration scheme. You will also implement a simple collision detection and resolution scheme to allow your mass-spring object to make contact with a flat ground plane.

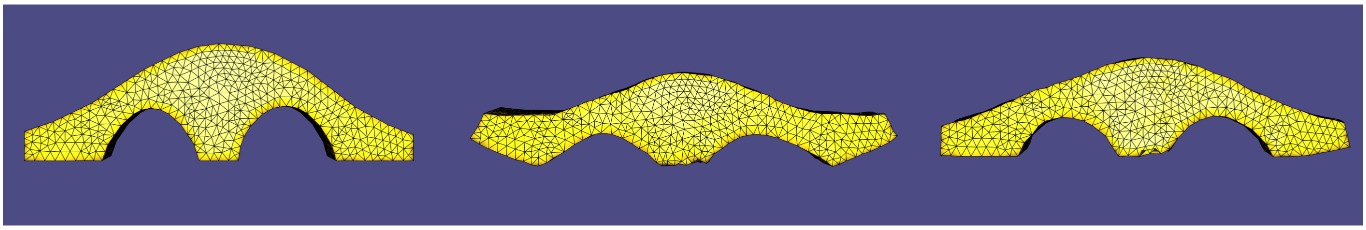

Fig. 1:Screenshots from final mass-spring simulation.

Setup Assignment Code Base

This should be done in a similar manner to all previous assignments

1. Run 'cd {SRC_DIRECTORY}/A3'
2. Run 'mkdir ./build'
3. Run 'cd ./build'
4. Run 'cmake .. -DCMAKE_BUILD_TYPE=RELEASE' to build the project Makefile.
5. Run 'make -j8'
6. This should build an execubtable named **./bin/CompFabA3**.

Part A: Implement Linearly Implicit Euler [4 marks]

1. The starter code will load a mesh of a bridge, assign each vertex to be a particle with mass and add a gravity force to each particle. The first part of this assignment will involve implementing a time integrator to advance this system in time. Start by opening the file '{SRC_DIRECTORY}/A3/include.TimeStepperEulerImplicitLinerA3.h'.
2. The method **step** is responsible for updating the displacements and velocities of the mass-spring object, over a single time step of duration dt.
3. Implement the linearly implicit Euler integration scheme described in class. You only need to alter the code in between the `//---- YOUR CODE HERE ----//` and `//---- END YOUR CODE HERE ----//` lines [4 marks]
4. Run the starter code using your new integrator. Verify that the default object falls straight down. Press 'a' to start the animation.

Part B: Implement Spring Force and Stiffness Matrix [12 marks]

1. Open '{SRC_DIRECTORY}/A3/include/ForceSpring.h'. You will be responsible for implementing the following functions: **getEnergy**, **getForce** and **getStiffnessMatrix**.
2. Recall that the energy for a single spring between two particles with positions in space $\mathbf{x}_0$ and $\mathbf{x}1$ is $E{spring}=\frac{1}{2} k \cdot\left(1.0 - \frac{|\mathbf{x}_1-\mathbf{x}_0|}{l_0}\right)^2$, where k is a positive real number that determines the stiffness of the sping and l_0 is the length of the spring in its undeformed state.
3. The force acting on the i^{th} node attached to the spring is $\mathbf{f}_i = -\frac{\partial E}{\partial \mathbf{x}_i}$, where $i\in [0,1]$. Derive the formula for the force on both spring nodes and include the derivation in your write up. [2 marks]
4. The stiffness matrix is matrix formed by $K_{ij} = \frac{\partial \mathbf{f}_i}{\partial \mathbf{x_j}}$. In this formaula each each ij is a block 3x3 matrix. The full stiffness matrix is a 6x6 matrix. Derive the stiffness matrix for your spring. [4 marks]
5. Implement both formulas in the **getForce** and **getStiffnessMatrix** methods respectively. You only need to alter the code in between the `//---- YOUR CODE HERE ----//` and `//---- END YOUR CODE HERE ----//` lines. [6 marks]

Part C: Implement Simple Collision Detection

1. After implementing the spring force, you will notice that your object still just falls straight down. This is because springs only exert force when deformed. To help deform the object you will implement a simple collision detection mechanism to allow your object to hit a horizontal ground plane.
2. Open '{SRC_DIRECTORY}/A3/src/main.cpp'
3. In the method **postDrawCallback** implement a collision detector which does the following:
   - checks whether any particle is below the floor
   - if so, sets the position of the particle to be at the floor
   - sets the velocity of the particle to be 0 (inelastic impact)

Part D: Take Some Screenshots

1. Run the code and take 3 screen shots of it. Include these in your report. You should see something like what is shown above.

Hand In [1 mark]

Collect all required images into a PDF report which must include your full name and student number. Submit this PDF and a zip file containt your A3 source code via **email** to **diwlevin@cs.toronto.edu**. The subject of the email must be **CompFabA3_LASTNAME_STUDENTNUMBER**.