1 Programming

In this assignment you will write a program (with some C++ code already provided to you) to animate a particle system that mimics fluid behaviour (think mud).

You should find the software, and associated README files, in the directories `mud_*`.

The project `mud_simulation` does the simulation of the particle system. As indicated in the file `mud.cpp` you will need to write the code for calculating the maximum time step, for running Symplectic Euler, and for computing the gravity force and the spring forces between particles. In the file `collisions.cpp` you will need to write the code which handles collisions between the particles and the ground, which in this case is not simply the flat plane $y = 0$ but instead the heightfield defined analytically by $y = 1.5(x^2 + z^2)$.

The project `mud_opengl` allows you to interactively view the output from the simulation, and save a snippet of RIB to define the camera you will use for the final render. It also allows you to take a PPM screenshot; if you don't have time to produce a high quality animation, then at least take a screenshot of one of the more interesting frames of your simulation (not frame 0!) to submit in lieu of an animation, to get part marks.

The project `mud_renderman` takes the output from the simulation along with a camera description and a RIB file, as in assignment 1.

As in assignment 1, you will need to produce an animation demonstrating your completed system. While the defaults given in the code produce something reasonable, feel free to tweak parameters, change the initial conditions completely, or make the collision geometry more complex. Submit the code you modify and a URL to the animation. **As with assignment 1, send the code you modified/added in an email to the course account, along with a URL to the animation.**

2 Analysis of mud

1) Derive the normal for the collision geometry you use in the programming part.

2) As seen in class, the time step you should select for an explicit solver like Symplectic Euler depends on what the particle forces are. If the step is too large, the system will be unstable and could explode; if it's too small, you are wasting time. Derive roughly the correct time step for your code to simulate mud. It should be based on the spring coefficients and the number of nearby particles to any given particle (you should come up with an approximate "average" figure for this number, and justify it!). Note that in practice, you will need to run some experiments to come up with the right fudge factor to multiply your analytic answer.

As before, give me the written part in class, in office hours, or under my office door before I get in the next day.