CS838 Topics in Computing:
Advanced Modeling and Simulation
Fall 2011

Programming Assignment #2

Due Monday 21 Nov 2011, 1:00pm

For this assignment, you are given the source code for two simulation tasks, involving volumetric tetrahedralized bodies. Your task will be to examine the code, comprehend how it correlates with the theoretical formulations discussed in lectures, and implement some minor modifications/upgrades.

`http://pages.cs.wisc.edu/~cs838-2/media/cloth.avi`

1. The first simulation involves a volumetric tetrahedral model in the shape of a sphere. This sphere is tessellated into approximately $21K$ vertices and $5K$ tetrahedra (i.e. this is a tessellation of the entire *volume* of the sphere, not just a triangulation of its surface). We specify that the "cap" of the sphere will behave as a kinematic handle, and we animate a periodic motion of this constrained part, which forces the sphere to swing back and forth. The code for this simulation is given in:

   `http://pages.cs.wisc.edu/~cs838-2/software/ball_semi_implicit.zip`

   A video of the simulation is shown in:

   `http://pages.cs.wisc.edu/~cs838-2/media/ball.mov`

   - In the code given to you, the *linear elasticity model* has been used to describe the material behavior of the elastic sphere. However, this model suffers from known artifacts, especially an apparent reduction in stiffness when it rotates, which is a clearly visible artifact in the video. Change the material model to *St.Venant-Kirchhoff* in order to annihilate (or limit) this artifact. Feel free to modify any additional parameters that will improve the visual appeal of the simulation.
   - [OPTIONAL, EXTRA CREDIT] Implement the *Co-rotated linear elasticity* or *Neohookean elasticity* material models, instead.

   Keep in mind the following:

   - The damping model implemented in the code is what we described as a *linear damping* model. You do not need to change this model, even if you change the definition of the elastic forces (i.e. linear damping is OK with St.Venant-Kirchhoff or Neohookean elastic forces, too).

- The maximum Δt for in this code is set manually, not via a CFL number. This is for the purposes of simplicity, as the exact formula for the maximum stable Δt that can be used would be somewhat complicated. If you encounter instabilities, you may need to reduce that time step accordingly.

2. The second simulation uses the same sphere, but with no kinematic constraints; instead, the sphere is set up such that it will collide and bounce on the rigid ground. A *penalty force* formulation is used for collision response. Any particle that penetrates into the ground feels a proportional spring-like force, that causes it to move upwards. Since penalty forces do not guarantee a collision-free configuration, we artificially "thicken" the ground such that collision forces start being applied at a small distance away from it. The code/video for this simulation are given in:

   `http://pages.cs.wisc.edu/~cs838-2/software/collision_semi_implicit.zip`

   `http://pages.cs.wisc.edu/~cs838-2/media/collision.mov`

   - The strength of the collision penalty force is dictated by the parameter `collision_coefficient`, which is the spring constant of the virtual zero-restlength spring that generates the collision response. Set the ground thickness to zero and try increasing this coefficient to the order of $10^9 - 10^{10}$ (in an effort to limit the interpenetration, and make it unnecessary to "thicken" the ground). What do you observe? What is the reason for this behavior?
   - Change the simulation so that the plane is inclined, rather than horizontal. You may observe artifacts as the sphere bounces against this inclined plane, due to the fact that the default material model that was implemented was *linear elasticity*; switch to an appropriate, different material model to limit these rotational artifacts.

Deliverable: Source code and (optionally) a compressed archive of the simulation output directories. Include a short report (1-1.5 page should suffice) with your findings. You can email the code to the instructor directly. If output files exceed 2-3MB, it is best that you send a download link for them.