Physikbasierte Modellierung und Simulation

https://graphics.tu-bs.de/teaching/ws1819/PBM

Assignment 2

Present your solutions for this sheet in the exercise on Thursday, November 15, 2018. This exercise extends the previous exercise about the simulation of particle systems.

2.1 Runge-Kutta method (50 points)

Implement a 4th order Runge-Kutta solver (e.g. http://en.wikipedia.org/wiki/Runge-Kutta_methods) in the function step(const Time timestep) in RungeKuttaSolver.cpp. Your solver has to evaluate the accelerations in the system at intermediate time steps; remember you can do this by setting the position and velocity of the particles, calling computeAccelerations() on the system, and reading the accelerations from the particles.

2.2 Adaptive explicit Euler method (30 points)

You may notice that for some situations the Euler solver may become unstable when the step size is too large. A clear sign of an unstable situation is that the result of the simulation changes when you change the step size. We want to automatically find a step size for which the simulation is stable. Denote the position of a particle \( p \) after a step of size \( h \) as \( \vec{x}_{p,h} \), and the position after two successive steps of size \( h/2 \) as \( \vec{x}_{p,h/2,h/2} \). Halving the step size of the simulation should not change the position of any particle by more than some threshold distance \( \epsilon \), that is,

\[
\max_{p} \left| \vec{x}_{p,h} - \vec{x}_{p,h/2,h/2} \right| \leq \epsilon .
\]

Implement the function step(const Time timestep) in AdaptiveEulerSolver.cpp. The function should find a step size that satisfies the above condition, compute an explicit Euler step with this step size, and repeat these two steps until a cumulated step size equal to the function parameter stepsize is reached. You may find the step size either analytically (recommended) or by successively reducing the step size until the condition is fulfilled.

2.3 Evaluation (20 points)

In order to get an intuition about how the different solvers behave under different conditions, perform at least one of the following tasks:

- Create a simulation that shows significantly different behavior for all three solvers. Try to explain why this happens, and how the simulations change when you vary the step size.

- Compare the run time and accuracy of all three solvers. In order to evaluate the accuracy, you may create “ground-truth” data by choosing a very small step size, or you may choose a very simple system for which an analytical solution is known (for example, a solar system consisting only of the Sun and the Earth).