Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

computer graphics laboratory

PHYSICALLY-BASED SIMULATION COURSE 2018

EXERCISE 1 - TIME INTEGRATION

Handout date: 26.09.2018
Submission deadline: 02.09.2018, 23:59 (optional)

General Rules

Setup. Please go to our gitlab repository (https://gitlab.vis.ethz.ch/cglphysics/PBS18-Exercises) and carefully follow the instructions to update and run your forked project.

What to hand in (optional). Implement your solution on your *own* repository including a READ file containing results, descriptions and so on. Then send me (kimby@inf.ethz.ch) your repository address.

Goal of this exercise

In this exercise, you will apply what you learned about the following time integration schemes:

- Explicit Euler
- Symplectic Euler
- Explicit Midpoint Euler
- Implicit Euler

Problem 1: Shooting a Cannon Ball

Statement. We will shoot a cannon ball having the mass m, the velocity $\mathbf{v}_t$, the position $\mathbf{p}_t$ at time t. In this problem, you are required to implement the following integrators in order to update the position $\mathbf{p}_{t+\Delta t}$ and the velocity $\mathbf{v}_{t+\Delta t}$ of the cannon ball with the time step Δt and the gravity $\mathbf{g}$:

- Analytic Solution

- Explicit Euler
- Symplectic Euler

Most of parts are implemented in the framework, except `advance()` method in `CannonBallSim.cpp`. Thus, you need to fill in `advance()` method with above time integrators to shoot a cannon ball.

Relevant member functions.

- `p_ball->setPosition(), getPosition()`
- `p_ball->setLinearVelocity(), getLinearVelocity()`

Goal. You can run by pressing *Run Simulation* button under *Simulation Control*, and pause with *Pause Simulation*. Then press *Reset Simulation* to reset, change to other integrator under *Simulation Parameters* and run it again. You will see color-coded trajectories (i.e., analytic, explicit, symplectic) for each integrator and compare them as seen in Fig. 1.

FIGURE 1. Screenshot of an example result of problem 1

Problem 2: Cube hanging from the ceiling

Statement. In this problem, a simple mass-spring system will be simulated as shown in Fig. 2. A cube has the mass m and the velocity $\mathbf{v}$ being connected to a spring at the position $\mathbf{p}$. One end-point of the spring is fixed at the position $\mathbf{p}_0$, and the other one falls due to gravity $\mathbf{g}$. It is characterized by its stiffness k, initial length L, and damping coefficient γ. These parameters are provided as function arguments. For damping, use a point-based damping force linear in the velocity so that the resulting force at the point $\mathbf{p}$ is

$$\mathbf{f} = -k(||\mathbf{p} - \mathbf{p}_0|| - L)\frac{\mathbf{p} - \mathbf{p}_0}{||\mathbf{p} - \mathbf{p}_0||} - \gamma \mathbf{v} + m\mathbf{g} \tag{1}$$

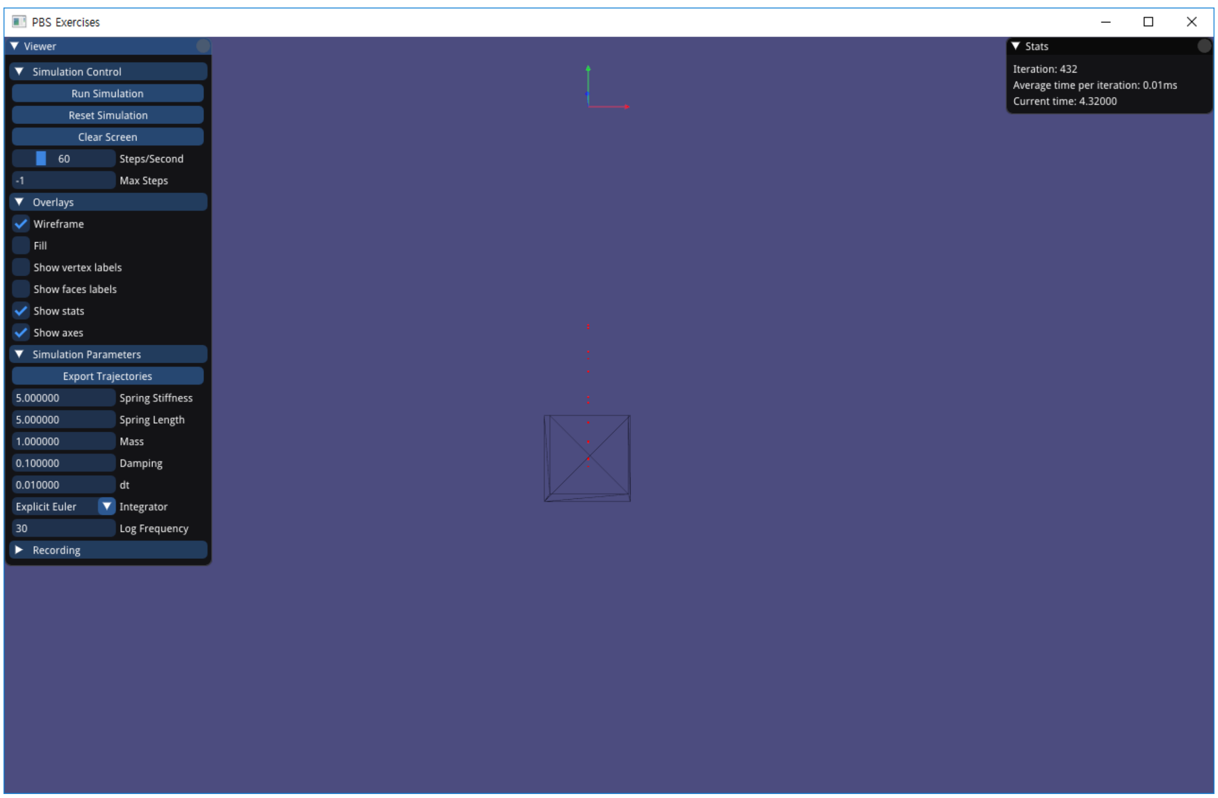

Figure 2. Screenshot of an example result of problem 2

Similar to problem 1, you are required to implement the following integrators:

- Analytic Solution
- Explicit Euler
- Symplectic Euler

- Explicit Midpoint
- Implitcit Euler

Again, you need to fill in `advance()` method in `SpringSim.cpp`. For analytic solution, find it with the following form and use its implementation as a baseline for comparison:

$$y(t) = c_1 e^{\alpha t} cos(\beta t) + c_2 e^{\alpha t} sin(\beta t) - L - m\frac{g}{k}, \quad \alpha = -\frac{\gamma}{2m}, \quad \beta = \frac{\sqrt{4km - \gamma^2}}{2m} \tag{2}$$

Note that it's a solution of 1-D behavior, and you are required to find only the constants c_1 and c_2. This should be done by using the rest state initial conditions (no energy in the system, except gravitational energy at $t = 0$).

Relevant functions and variables.

- `p_cube->setPosition(), getPosition()`
- `p_cube->setLinearVelocity(), getLinearVelocity()`
- `p_cube->getMass()`
- `m_spring`
- `m_time, m_gravity, m_dt`

Goal. You can test and compare the stability of integrators by setting a large time step (i.e., $\Delta t = 0.05$) or no damping (i.e., $\gamma = 0$). Which one is stable and which one is not? Is there any correlation between stability and accuracy (order of method)?