Project 2: Fluid simulation

Due date: **23.06.2023**

Overview

In this project you will experiment with coupled *fluid-rigid simulations*. You must implement at least the required features given below; you will also have to make a demo video showing your simulator in action. Additionally, you may implement some of the listed extensions (or invent your own!) for extra credit.

Stam's fluid simulator

The starting point of this project is provided by Stam's implementation ref. [6] (see also his paper ref. [5]) of a basic 2D fluid simulator. The code performs the basic fluid steps, as follows:

- **Density Advection** Uses the Semi-Lagrangian advection algorithm described in the Stable-Fluids paper (ref. [1]); see also slides.
- **Velocity Advection.** Similar to density advection; applies twice scalar advection for each component of the velocity field.
- **Density and velocity diffusion.** The implementation is described in the paper by Stam [5]; see also slides.
- **Pressure Projection.** The implementation is described by Stam [5]; see also slides.
- **Boundaries.** The boundaries of the grid form a solid boundary. This implies path clipping at the boundary for the advection step, and correctly defining the pressure at the boundary.

Both diffusion and projection steps above rely on implicit ODE integration, thus leading to solving a linear system. As linear system solver, Stam's code uses the Gauss-Seidel method, see ref. [7]. The conjugate gradient method could also be used, but the implementation is more involved.

Make sure that you understand Stam's method and its implementation, since you will need to heavily modify the code base. However, before studying the code, carefully read refs. [1] and [5].

Required Features

- **Vorticity Confinement: 1 points.** Implement Vorticity Confinement. The implementation is described in the ref. [2].
- **Fixed Objects: 2 points.** Any edge between neighbouring cells can be marked as boundary edge, allowing for arbitrary, fixed solid boundaries within the fluid simulation domain.
- **Moving Solid objects: 5 points.** Implement fluid-solid interaction with moving (solid) objects. Allow the user to drag solid objects within the flow. The objects come to rest at exact grid offsets. The movement of the object must exert a force on the fluid. Proper boundary conditions should be applied.
- **Moving Rigid bodies: 4 points.** Similar to the above, but allow object(s) to rotate.
- **Colliding contact between rigid bodies: 4 points.** Handle colliding contact among simple rigid objects (boxes, disks, etc.); see course notes and slides. You may simplify here and, get away with just applying impulses so as to separate the objects. Resting contact is not required.
- **Two-way Coupling (extension to moving objects): 2 points.** Not only do the objects affect the flow, but the flow also exerts forces on the objects.
- **Particles and Fluids: 2 points.** Put your particle system from the previous assignment in your fluid-simulation system. Allow the velocity field to exert forces on the particles. Simulate a piece of cloth interacting with the fluid.

Your demo must be able to turn on and off some of the features above.

Optional Features

- **Temperature**. Causes hot air to rise; see ref. [2].
- **Water**. Implement a water solver with zero air pressure. Could use some signed-distance function to track the air-water interface, but no need to use the full particle-level-set method, see ref. [4].
- **Free-surface**. Trace the air-water interface using a simple particle-based method.

Deliverables

- **Code.** On the due date, you must deliver the code of your project via Canvas. Instructions about how to compile and run your project should also be included.
- **Demo video.** Shows-off what and how you implemented; delivered through Canvas.

- **Paper.** The paper (4 – 8 double-column pages, in e.g. IEEE format) describes the results you obtained, and the methods you used (implemented). Snapshots, pseudo-code, equations, timings, etc. should be included; deliver through Canvas. Additionally, make sure that you include an appendix in which you explain what and how each group member contributed to the project. Deliver the paper through Canvas.

References

[1] STAM, J. 1999. Stable Fluids. In Computer Graphics (SIGGRAPH 1999), ACM, 121-128.

[2] FEDKIW, R., STAM, J., AND JENSEN, H. 2001. Visual Simulation of Smoke. In Computer Graphics (SIGGRAPH 2001), ACM, 15-22.

[3] Rendering smoke and fire in real-time. `http://graphics.ethz.ch/teaching/former/imagesynthesis_06/miniprojects/p3/index.html`

[4] ENRIGHT, D., MARSCHNER, S., AND FEDKIW, R. 2002. Animation and Rendering of Complex Water Surfaces. In Computer Graphics (SIGGRAPH 2002), ACM, 736–744.

[5] Jos Stam. Real-Time Fluid Dynamics for Games. Proceedings of the Game Developer Conference, March 2003.

[6] Jos Stam, solver code, `http://www.dgp.toronto.edu/people/stam/reality/Research/zip/CDROM_GDC03.zip`.

[7] Wikipedia, Gauss-Seidel method, `https://en.wikipedia.org/wiki/Gauss%E2%80%93Seidel_method`.