Project 5: Fun with fluid

Due: Apr 2, 2018, 11:55PM

Let's build a fluid simulator! Your job is to create 2D smoke fluid simulation using a grid-based, Eularian approach. Your velocity field and density field must obey Navior-Stoke equations, incompressibility, and boundary conditions. The simulator must be interactive and run in real-time; the user can use a mouse to manipulate velocity and density fields and visualize the smoke evolving in the 2D grid in real-time. The skeleton code provides a pretty good starting point for you. It includes basic UI and mouse interaction code, as well as the data structures and major procedures required to simulate 2D fluids. In particular, the code for evolution of density field includes all three steps: adding density, diffusion, and advection. However, the code for evolution of velocity field is incomplete, as you can see that the smoke stays stationary in the current skeleton code. You will develop code for velocity advection and projection to move the smoke around accurately. Specifically, your task is to

1) develop a procedure to advect the velocity field using semi-Lagrange method
2) develop a procedure to project the velocity field such that incompressibility and boundary conditions are satisfied.
3) Create a colorful smoke simulator by using three density fields, each of which stores the density of a color channel (e.g. Red, Green, Blue).
4) After you are done with your fluid simulator, use your simulator to create a 10 second of animation for a screen saver program. 5% of your grade for this project will be based on your "effort" on the animation you create.

Skeleton code

Download the following skeleton code. You don't need DART for this project. The two functions you need to complete are MyWorld::advectVelocity and MyWorld::project.

To create the screen saver animation, run your program with an argument "-p". On a Mac or a Linux, simply modify you command line to: ./main.out -p. On a Windows, right click your project in Solution Explorer and select Properties from the menu. Go to Configuration Properties -> Debugging. Set the Command Argument "-p" in the property list.

The argument "-p" will set the boolean variable "screenSaverOn" to 1. The function initializeFields() will be called from the main function. You can program the initial density and velocity fields in this function. Run the animation by hitting the space bar.

UI control provided by the skeleton code:

1. Left click and drag: add density
2. Right click and drag: add velocity
3. 'v': Visualizing velocity/density
4. Space bar: simulation on/off

Extra points:

1. Implement code to load in an initial density field described as an image (grayscale of RGB). e.g. You can load in a picture of Mona Lisa and blow it away (2 points).
2. Add arbitrary boundary in the grid. e.g. Create a "C" shape in the middle of the grid to see how it affects the smoke (3 points) .
3. Add external forces to the velocity field. e.g. Mimic the effect of a fan in the middle of the grid (2 points).