CIS 563 : Phyiscally Based Animation
Homework 3 : Fluid Simulation
Spring 2015

March 17, 2015

Due Date: March 30, 2015

Abstract

In this homework we will be writing a 3D Semi-Lagrangian smoke fluid simulator. The framework for drawing and running the code has been provided for you on Canvas. Much of the fluid simulator we will be writing is based off the SIGGRAPH 2007 Course Notes on Fluid Simulation (Bridson, Muller). If you have any questions, please refer to the notes before asking the TAs. There is a high chance the notes will answer your question faster than we can answer you on Piazza.

1 Grid : 15 points

Grid data structure that represents the Eulerian grid in our simulation.

1.1 grid_data.cpp

Much of the work will be done in grid_data.cpp. Most of this file contains stubs for the grid class that you will need to implement. You do not need to follow each of the stubs : do those and change the ones you see fit to match the architecture you are writing your fluid simulation. However, we strongly advise you to do design your grid with the following methods: 1. operator() for access / setting cells, 2. worldToSelf for changing from world indices into the local grid indices.

2 Set up simulation : 5 points

2.1 smoke_sim.cpp

Most of this class has been implemented for interaction purposes. The `step()` function is left for you to implement. The `step()` function is called on every step of the simulation to step the simulation forward by a single step.

2.2 mac_grid.cpp

The MAC Grid interface has been given to you. The functions to perform math on the MAC Grid are written as stubs. You will need to set up the actual information itself : the grids for the material quantities and fields on which we will be performing our math. In the basic version of the simulation, you will only consider density. As part of extra credit, you may add the temperature property.

In addition to initializing the MAC Grid, you will need to update the sources. The simulation framework we give you starts with an empty box. In order for smoke to enter this system, you will need to update the sources of the smoke coming into the system in `updateSources()`. You may choose whatever artistically pleasing sources make sense for you.

In the demo recording, we use the following settings : upward velocity of 2.0 at (0, 1, 0) and (0, 2, 0) on the Y grid, density of 1.0 at (0, 0, 0) and (0, 1, 0), and temperature of 1.0 at (0, 0, 0) and (0, 1, 0).

3 Advect Velocity : 20 points

In performing the simulation, you must update the velocities of the virtual particles in the simulation. Beware of the CFL condition when the changes are great. For more information, read the `Advection Algorithms` section of the Fluid Simulation course notes.

4 Apply forces : 20 points

In this section we will apply external forces on the system, specifically buoyancy and vorticity confinement. Stub functions have been provided for you to fill out. Please refer to `Smoke Simulation` section fo the Fluid simulation course notes to direct your implementation.

5 Project : 25 points

This is, arguably, one of the most important sections of fluid simulation. In this section we will be performing a pressure solve in order to maintain incompressibility of the fluid. You will first solve Ax = b for pressure, and then subtract the pressure from the velocity and save the difference in the target grid. We have given you a function to build A `calculateAMatrix` and a basic conjugate gradient solver `conjugateGradient`. Preconditioned conjugate gradient (PCG) solver is not required for the basic implementation and is left for extra credit. If you are planning on doing your final project on fluids, we strongly recommend you implement the preconditioning step.

6 Advect Other Material Properties : 5 points

In addition to advecting the velocity, we will need to advect the rest of the particle's properties. Refer to `Advection Algorithms` of the fluid simulation course notes.

7 Extra Credit

By the end of this homework, you will have implemented a basic fluid simulation with smoke support. This is just the beginning (albeit a very important beginning). Here is some extra credit you may do. You may earn up to a maximum of 50 points in extra credit. However, we will emphasize that incorrect implementation of the base fluid simulation will nullify any extra credit you may try to add on top of the fluid simulation, so PLEASE make sure that your base implementation is CORRECT.
If you are planning on doing anything that is not specified below or are planning on doing water simulation related extra credit, please be sure to post on Piazza or email Harmony (harmoli@seas.upenn.edu) to discuss what you're doing.
NOTE : If you are planning on doing anything fluids related (i.e. smoke, water, viscous fluids, multiple interacting liquids, fire, etc.), we strongly recommend you to implement preconditioning and to keep your code NEAT.

7.1 Temperature : 5 points

Like adding density, you will add temperature as a material property. However, you will need to take into account how temperature is transferred and how it affects the motion of the fluid itself. For more information, refer to the `Smoke` section of the fluids notes.

7.2 Preconditioned Conjugate Gradient : 20 points

In the conjugate gradient function, we have given stubs for the places where the preconditioner should be included. For information on how this helps the solver on a whole and how to implement it, refer to the `Maintaining incompressiblity` section of the fluids notes.

7.3 Optimization : 20 points

There are multiple ways to optimize in the fluid simulation, including multithreading and parallelizing portions of the fluid simulation. Naive parallelization will earn 5 - 10 points of extra credit. Substantial architectural changes for optimization beyond PCG will earn 20 (or more) points, depending on what is involved. For point evaluation, email Harmony.

7.4 Other Simulation : up to 50 points

This includes adding marker particles, surface tension, etc. and as far as expanding this simulation to other newer hybrid methods of fluid simulation. If you plan on doing this option, please make sure to email Harmony for point value evaluation. NOTE : Whatever you do in this section can be used in your final project; however, if you submit it for this assignment, you will not be allowed to use it as the core of your final project. For example, if you choose to extend this fluid simulation to a fire simulation, you will not be allowed to a final project that simply extends fluid simulation to a fire simulation; however, you would be allowed to do a final project that deals with fire simulation and burning objects, so long as there is a substantial feature on top of the fire simulation you build for this extra credit.

8 User Interaction Notes

The commands in the interface are as follows:

<	Start / Restart
>	Reset
=	Pause
(space)	Reset camera to default
R	Start / Stop recording
V	Toggle velocity field
0	Cube view mode
1	Plane view mode
Mouse left	Orbit camera
Mouse middle	Zoon camera