A 2D Fluid Simulation using Smoothed Particle Hydrodynamics

Lalit Lal
University of Toronto
lalit.lal@mail.utoronto.com

***Abstract*— Fluid simulation can provide users intuition about how certain physical objects behave, as well as direction for improving certain applications. This work presents the implementation and results of a 2D fluid simulation against free surfaces using smoothed particle hydrodynamics. This work aims to implement a simplified 2D SPH fluid model, which applies solving Navier-Stokes equations on a particle-based system using smoothing kernels.**

CCS Concepts: Computing Methodologies → Animation
KEYWORDS: Fluid Simulation

1. Introduction

This work attempts to outline the rudimentary implementation of a particle-based, simplified 2D fluid simulation for two scenarios (single/multi stream freefall, and a dam-breaking scenario).

2. Related Work

This work provides no novelty in the area of fluids modelling; it is merely a summarized implementation of the work presented by [Bridson and Müller 2007].

3. Method

This section provides the algorithmic details of the SPH implementation. It takes the form of describing the application of fluid advection, external body forces, pressure projection and viscosity forces using smoothing kernels, and finally applying boundary conditions to constrain the fluid to its environment and physical limitations. In the following pieces, the following notations are used X^t_{xy}, V^t_{xy},F^t_{xy}, ρ^t, P^t representing position, velocity, force, density, and pressure respectively, in their various components. Each particle in the system has these attributes.

A. Particle Advection

Particle advection refers to the time dependent displacement of the individual particle based on its existing velocity. As such, per particle, advection would be defined using Δt as the time step.

$$X^t_x \mathrel{+}= \Delta t V^t_x$$

$$X^t_y \mathrel{+}= \Delta t V^t_y$$

B. External Body Forces

In this case, the only external force that is accounted for in this section is gravity. As such, given a constant gravity acceleration, a particle's position would also displace by

$$X^t_y \mathrel{+}= \frac{\Delta t^2}{2} a^t_y = \frac{\Delta t^2}{2} g$$

where g is the acceleration due to gravity, a simulation parameter.

C. Density Impact

In this section, all particles within a certain distance, r of a particle, p_i are noted to be its neighbors in a separate list, and then the neighboring particle's density impact on this particle is accumulated using the W_{poly6} smoothing kernel, using the distance between the two particles i and j, r_{ij}

$$\left\| r^t_{ij} \right\| = \sqrt{(X^t_{jx} - X^t_{ix})^2 + (X^t_{jy} - X^t_{iy})^2}$$

$$\rho^t_i = \sum_{j \neq i}^{N} m_i * W_{poly6}(r^t_{ij}, h)$$

Where W_{poly6} is (for neighboring particles only within r, otherwise 0)

$$W_{poly6}(r^t_{ij}, h) = \frac{315}{64\pi h^9}(h^2 - \left\| r^t_{ij} \right\|^2)^3$$

and h is a parameter. The accumulated densities will be used for pressure projection on each particle.

D. Pressure Projection

This is a two stage procedure. Firstly, iterate through each particle, p_i and calculate its pressure using the ideal gas law:

$$P^t_i = k(\rho^t_i - \rho_{rest})$$

where k ρ_{rest} are the gas constant and rest density of the fluid particle, and are simulation parameters.

Secondly, loop through p_i again, and for each of its earlier noted neighbors, accumulate the global force for p_i using the following formula (similarly for the y-component):

$$F_i^t x \mathrel{-}= \sum_{j \neq i}^{N} \left\| r_{ij,x}^t \right\|^2 m_i \frac{P_{ix}^t + P_{jx}^t}{2\rho_j^t} \nabla W_{spiky}(r_{ij}^t, h)$$

where N is the set of of neighbors for p_i, $\nabla W_{spiky}(r_{ij}, h)$ is defined as the gradient of the spiky smoothing kernel.

$$\nabla W_{spiky}(r_{ij}^t, h) = \frac{-45}{\pi h^6}(h - \left\| r_{ij}^t \right\|)^2$$

E. Applying Viscosity Forces

Finally, applying the viscosity forces involve a similar procedure of looping through each particle and its respective neighbors. Given a particle, p_i. and its set of neighbors, the force due to viscosity is accounted for in each particle by applying the following formula to p_i's global force vector (similar for y):

$$F_i^t x \mathrel{+}= \mu \sum_{j \neq i}^{N} m_i \frac{V_{jx}^t - V_{ix}^t}{\rho_j^t} \nabla^2 W_{visc}(r_{ij}^t, h)$$

where N is the set of neighbors for p_i , μ is a viscosity parameter controlled in simulation, and $\nabla^2 W_{visc}(r_{ij}, h)$ is the Laplacian of the viscosity kernel, and is defined as

$$\nabla^2 W_{visc}(r_{ij}^t, h) = \frac{45}{\pi h^6}(h - \left\| r_{ij}^t \right\|)$$

It is important to note here that velocity is simply the change in position over time, $\frac{X^t - X^{t-1}}{\Delta t}$.

F. Boundary Conditions

After displacing each particle and updating its forces, velocities are calculated and the boundary conditions are simply used to enforce that the position of each particle is not beyond the simulation visibility space, nor is its velocity beyond a certain value (else it is dampened).

G. 2D SPH Algorithm Overview

Summarizing each of the stages above into a unified algorithm, the 2D simplified SPH is in algorithm 1. The limitations of this work is that it does not account for surface tension, and it clearly suffers in performance due to a lack of grid structure. Additionally, on a technical note, applying the gas equations neglects the incompressibility of liquids.

4. Experimental Results

A. Results

This section presents the visualized results of the simplified 2D SPH implementation for specific cases. Specifically, two different viscosity values for 3 scenarios are shown in figure 1 - dam (a, b) single stream (c, d), multi-stream (e, f). The left column (a, c, e) are for a lower viscosity (0.25), and the right column (b, d, f) are a higher viscosity, (2.5).

Result: SPH Update
for p_i *in particles* **do**
    Save a copy of $X^t \rightarrow X^{t-1}$;
    advectParticle();
    applyBodyForce();
    applyBoundaryConditions();
end
clearAllNeighborsandDensities();
for p_i *in particles* **do**
    buildNeighborsAndDensity();
    projectPressure();
    applyViscosityForce();
end

Algorithm 1: SPH Algorithm

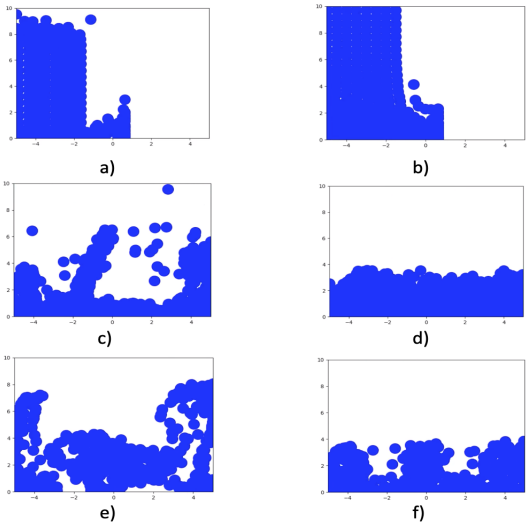

Fig. 1: Results for multiple scenarios of 2D SPH

5. Summary and Future Works

Overall, this work is the simplified implementation of a 2D fluid SPH algorithm. By not accounting for surface tension and neglecting incompressibility by using the gas law, further improvements can still be made. Other improvements include providing interactive forces and structures to impede fluid flow. Furthermore, on a performance note, GPU acceleration and adopting a grid structure can help speed the performance of the simulation.

6. Acknowledgements

This work is inspired by the summary of SPH fluid simulation by [Bridson and Müller 2007], and is complemented by David Levin's course lecture on fluids, for which this report is written.

REFERENCES

Bridson, R., and Müller-Fischer, M. 2007. Fluid Simulation. SIGGRAPH 2007 Course Notes

APPENDIX

A. *Simulation Setup*

Overall, one of the tougher parts of this simulation involved empirically, through intuitive trial and error, choosing the set of simulation parameters in order to make the model behave visually realistically. The simulation parameters are summarized in table 1 below.

Name	**Value**
number particles	500
gravity	$\frac{-9.81}{1000}$
(simulation) width	5
(simulation) floor	0
initial particle spacing	$\frac{width}{10}$
gas constant, k	$\frac{width}{1000}$
rest density, ρ_{rest}	1.0
max velocity, V_{max}	2.0
wall damp, $wall_{damp}$	0.09
floor damp, $floor_{damp}$	0.05
velocity damp, V_{damp}	0.5
viscosity factor, μ	0.25, 2.5
kernel distance, h	20
mass (same for all)	1000
neighbor radius, r	spacing * 1.75

TABLE 1: Simulation Parameters