Simulating Fluid Motion using Smoothed Particle Hydrodynamics

Aayush Joshi

Abstract

Realistic simulation of fluids is an important tool with a wide variety of applications such as within the Aerospace industry to model fluid based phenomena of spacecraft parts and within the computer games industry for authentic graphics. In this paper, I explore a method for simulating fluids known as Smoothed Particle Hydrodynamics (SPH) in order to better understand the mathematical theory behind Computational Fluid Dynamic methods and their implementation in an appropriate programming language, C++.

Contents

1 Introduction **2**
- 1.1 Motivation 2
- 1.2 Definitions 2
- 1.3 Smoothed Particle Hydrodynamics 3
- 1.4 Outline and Structure 3
- 1.5 Success Criteria 3
- 1.6 Skills 4

2 Research Review **5**
- 2.1 History and Relevant Literature 5
- 2.2 Alternative Approaches 6

3 Theoretical model **7**
- 3.1 Smoothing Kernel 7
- 3.2 Interpolation Equation 7

4 Development and Testing **7**

5 Evaluation and Final Remarks **7**

1 Introduction

The field of simulation is one with many applications in all industries, with much overlap between Mathematics, Physics and Computer Science due to its predictable behaviour. One such application is Computational Fluid Dynamics (CFD), or in other words predicting the movement of fluids, which will be the focus for this project.

Simulating fluids involves observation of fluid phenomena such as wind, weather, ocean waves, waves induced by ships or simply pouring a glass of water. Such phenomena may seem extremely trivial at first glance, but in reality involve a deeper understanding of physical, mathematical and algorithmic methods.

1.1 Motivation

My motivation for this project stems from the work of Sebastian Lague [1], a games developer who shares his exemplar work on Github and through digital media on YouTube. Through his work, I was introduced to the concept of Smoothed Particle Hydrodynamics in the Computer Graphics community and was given great insight into the expectations from a project such as this. Further reading, especially into the sources of Lague, piqued my interest and only reinforced the idea of undertaking this concept because it provided the overlap between Mathematics, Physics and Computer Science, it was far beyond the scope of the A level curriculum but most importantly it provided a means to challenge, extend and implement new knowledge in a field which I plan to undertake in the future.

1.2 Definitions

In this section, I provide clear and concise definitions for the key terms essential to understanding the context and methods presented in this paper.

Computer Graphics. A technology that generates images and videos on a computer screen, also referred to as CG.

Simulation. Imitation of a situation or process.

Frame. A single image which makes up a collection of images for an animation.

Render.

Algorithm. A set of instructions used to solve a particular problem or perform a specific task.

Pseudocode. Writing an algorithm in plain English for design purposes.

Optimisation. Modifying an algorithm or software to reduce the usage of computer resources or compute time.

Lagrangian. A particle based approach to simulation.

Eulerian. A grid based approach to simulation.

Velocity. Speed of an entity associated with a direction.

Acceleration. The rate of change of velocity.

Force. An influence which causes an object to accelerate.

Friction. A force resisting the relative motion of an object.

Fluid. Any substance which flows due to applied forces, namely Liquids and Gasses.

Liquid. A type of fluid which takes the shape of any container or vessel it is stored within.

Advection. The horizontal movement of a mass of fluid.

Incompressible.

Density. The compactness of a substance, or the mass per unit volume.

Pressure. The physical force exerted on an object by something in contact with it, or the force per unit area.

Viscosity. A quantity defining the magnitude of the internal friction in a fluid, or the Pressure resisting uniform flow.

Surface Tension. The tension on the surface of a liquid caused by the attraction of particles in the surface layer, tends to minimise surface area.

1.3 Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics (SPH) stands out as a Lagrangian approach to fluid simulation, offering a dynamic method for modeling complex fluid behavior. Developed in 1977 from the work of Lucy [2] and Gingold and Monaghan [3] in astrophysics, it posed as a strong alternative to existing methods at the time. Its transformative potential was further realized in interactive liquid simulation, thanks to the efforts of Müller *et al.* [4] in 2003.

In SPH, the spatial domain is approximated into particles, each embodying various fluid properties like mass, density, and velocity. Throughout the simulation, these particles dynamically interact, forming a fluid-like continuum. Notably, the field quantities characterizing the fluid, such as pressure or velocity, can be precisely evaluated at any point in space by observing the overlapping influence spheres of individual particles. Adaptability and precision makes SPH a compelling choice for simulating fluid phenomena across a spectrum of scales and applications.

1.4 Outline and Structure

I plan to code a semi-realistic 2-D animation of an incompressible liquid in the programming language C++. This will involve describing liquid phenomena mathematically to come up with a theoretical model. I will then implement each section of the theoretical model, test its efficacy and possibly look into optimisation techniques as required. Finally to evaluate the success of my simulation I will check against the success criteria, reverting to previous methods of development if necessary.

1.5 Success Criteria

The success criteria is as follows:

- Implement all aspects of the Theoretical model within the animation where every section behaves as intended in C++.
- Have an animation of a semi-realistic 2-D incompressible fluid where the

window can be resized to interact with the simulation.

- Have an animation that runs at a satisfactory speed with minimal time lag and resource wastage.

1.6 Skills

2 Research Review

Much of my research comes from GitHub, a Microsoft owned cloud-based platform for developers to store their personal or professional projects and publish them for wider use by the community. From the GitHub page of Sebastian Lague, I was introduced to articles on SPH written by many reputable institutes such as by *ETH Zurich*, *Université de Montréal* and *University College London, UK*. The majority of these papers had affiliation with *SIGGRAPH*, the international Association for Computing Machinery's Special Interest Group on Computer Graphics and Interactive Techniques. SPH related techniques are researched and published most for showcase at *SIGGRAPH* events. Through the use of google scholar and *SIGGRAPH*, I have been able to narrow my search for related documents for simulating liquids.

2.1 History and Relevant Literature

Lucy [2] introduced Smoothed Particle Hydrodynamics as a numerical testing tool for astrophysical calculations involving fission[1] within stars. This idea of quantity interpolation or "approximation" of fluid quantities was furthered by Gingold and Monaghan [3] and applied to non-spherical stars. Although both sources provide appropriate applications of this technique, the obvious limitation is that the majority is within the context of Astrophysics and not CFD. Additionally, both sources were released in 1977 with major development in the simulation field, such as the use of more modern optimisation techniques which utilise the powerful hardware now widely available, leading to the source being obsolete for present-day applicational use.

The work of Müller *et al.* [4] adapted SPH for interactive fluid applications, the first of its kind, putting forward an alternative Lagrangian method than the more common Eulerian method used for CG and modelling purposes. The paper provides a gentle introduction to SPH with a mathematical brief to the most important phenomena observed within fluids for simulation, including Pressure, Viscosity and Surface Tension. There is a distinct lack of algorithms, which leaves implementation up to the reader but the paper fulfills its purpose as an excellent introduction to SPH.

After the foundational work in 2003, Clavet *et al.* [5] release their work two years after with the primary focus on implementation, introducing key algorithms such as the Simulation step which covers the pseudocode for every frame of the animation and how the quantities of individual particles change frame by frame. Problems specific to implementation are also acknowledged, for example the near-density and near-pressure tricks are also introduced which prevent an issue that causes liquid particles to cluster.

An example of a more recent publication is Koschier *et al.* [6] in 2019. This tutorial summarises the state of SPH in its entirety by covering the theory and implementation rigorously, but also with a focus on optimisation methods to lessen compute time utilising modern hardware. The tutorial is diagrammatic and visual helping reinforce the ideas being expressed. Compared to earlier iterations covering SPH, this paper acts as the ultimate guide by placing all the information needed in one

[1] Splitting of atomic nuclei causing a release of energy

document. The paper dives much deeper into the niche complexities involved with simulating any fluids or even soft-bodied solids, but are beyond the scope of this project.

2.2 Alternative Approaches

An alternative approach for simulating fluids I have mentioned across this write-up is the Eulerian approach. Robert Bridson, in his book *Fluid Simulation for Computer Graphics* [7], perfectly encapsulates the Eulerian Viewpoint. In his words, *"The Eulerian approach, named after the Swiss mathematician Euler, takes a different tactic that's usually used for fluids. Instead of tracking each particle, we instead look at fixed points in space and see how measurements of fluid quantities, such as density, velocity, temperature, etc., at those points change in time. "*. The non-particle centric approach means the fluid is treated like a continuous medium and the simulation solves Partial Differential Equations (PDEs) to model its behaviour. Spatial domain is split up into equal sized grids and the fluid is modelled as being incompressible which would mean the total inflow within a grid must equal the total outflow. This process leads to advection within the simulation and then is rendered on screen.

Grid-based approaches have the advantage of having higher numerical accuracy and efficiency because solving PDEs can be optimised using techniques like finite difference or finite volume methods. Exactly enforcing incompressibility is important for accurate production of turbulence, and SPH methods have a hard time enforcing incompressibility efficiently. They also can have difficulty allocating computational elements throughout space efficiently. For these reasons, they have not been demonstrated to be effective for calculating flows such as air around a car, as stated per this paper by NVIDIA employees in proceedings of the 2010 ACM SIGGRAPH symposium [8].

For my artefact, I aim to create a semi-realistic animation of a liquid suggesting SPH is a viable technique as I do not aim for accuracy like some applications in industry require, for example modelling airflow around rocket fuselages. Furthermore, I aim to create an element of interaction by resizing windows to show some kind of advection. "Particle-based methods like Smoothed Particle Hydrodynamics (SPH) are attractive because they do not suffer from the limitation to be inside a box" [8], also implying that resizing is impractical to implement with an Eulerian approach as resizing the window would restructure the grids that an Eulerian approach relies upon.

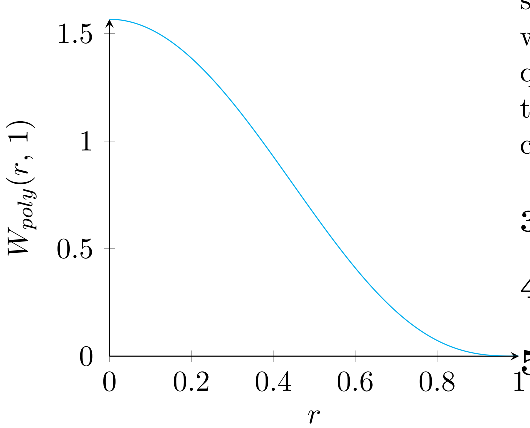
3 Theoretical model

3.1 Smoothing Kernel

The fundamental property every SPH particle has is a circle of influence which is dictated by the Smoothing Kernel $W(r, h)$, where r is the distance between the sample point and the particle centre and h is the radius of the circle of influence, also known as the smoothing radius. The closer a sample point is to the centre of the particle, the larger the influence the particle has on the sample point. This influence value can be described and quantified by picking an appropriate smoothing kernel, which is vital as it is used by the **Interpolation Equation** to calculate scalar quantities, most notably density.

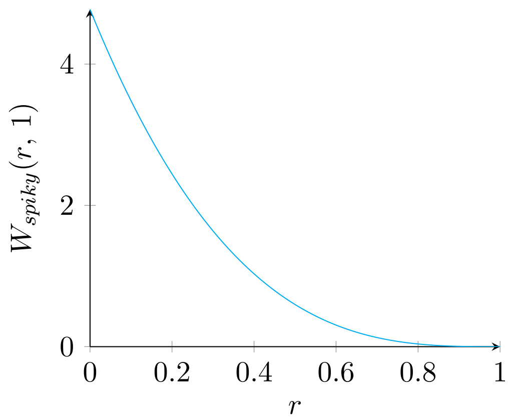
Müller *et al.*[4] describe two popular smoothing kernels, both with different properties.

$$W_{\text{poly}}(r, h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2)^3 & 0 \leq r \leq h \\ 0 & \text{otherwise,} \end{cases}$$



$W_{\text{poly}}(r, h)$ was created as an all purpose kernel with its major highlight being that r appears squared, so the square root does not need to be evaluated when using the Pythagorean Theorem to calculate distance, easing the distance computations. Müller *et al.*[4] mention that if this kernel is used for pressure computations, which this project will entail to enforce incompressibility, a problem arises where the particles will tend to cluster because the gradient is used to compute pressure. The gradient of $W_{\text{poly}}(r, h)$ drops close to 0 for small r leading to a diminishing repulsion force.

$$W_{\text{spiky}}(r, h) = \frac{15}{\pi h^6} \begin{cases} (h - r)^3 & 0 \leq r \leq h \\ 0 & \text{otherwise,} \end{cases}$$



$W_{\text{spiky}}(r, h)$ is used specifically for pressure computations. Its gradient is high when r is close to 0, generating the required repulsion forces for pressure calculations and therefore making it the superior choice for my simulation.

3.2 Interpolation Equation

4 Development and Testing

5 Evaluation and Final Remarks

References

[1] S. Lague, "Coding adventure: Simulating fluids," 2023. [Online]. Available: https://github.com/SebLague/Fluid-Sim

[2] L. B. Lucy, "A numerical approach to the testing of the fission hypothesis," *Astronomical Journal, vol. 82, Dec. 1977, p. 1013-1024.*, vol. 82, pp. 1013–1024, 1977.

[3] R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics: theory and application to non-spherical stars," *Monthly notices of the royal astronomical society*, vol. 181, no. 3, pp. 375–389, 1977.

[4] M. Müller, D. Charypar, and M. Gross, "Particle-based fluid simulation for interactive applications," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation.* Citeseer, 2003, pp. 154–159.

[5] S. Clavet, P. Beaudoin, and P. Poulin, "Particle-based viscoelastic fluid simulation," 2005. [Online]. Available: http://www.ligum.umontreal.ca/Clavet-2005-PVFS/pvfs.pdf

[6] D. Koschier, J. Bender, B. Solenthaler, and M. Teschner, "Smoothed particle hydrodynamics techniques for the physics based simulation of fluids and solids," 2019. [Online]. Available: https://github.com/InteractiveComputerGraphics/SPH-Tutorial/blob/master/pdf/SPH_Tutorial.pdf

[7] R. Bridson, *Fluid simulation for computer graphics.* AK Peters/CRC Press, 2015.

[8] J. M. Cohen, S. Tariq, and S. Green, "Interactive fluid-particle simulation using translating eulerian grids," in *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, 2010, pp. 15–22.

[9] C. Braley and A. Sandu, "Fluid simulation for computer graphics: A tutorial in grid based and particle based methods," *Virginia Tech, Blacksburg*, 2010.