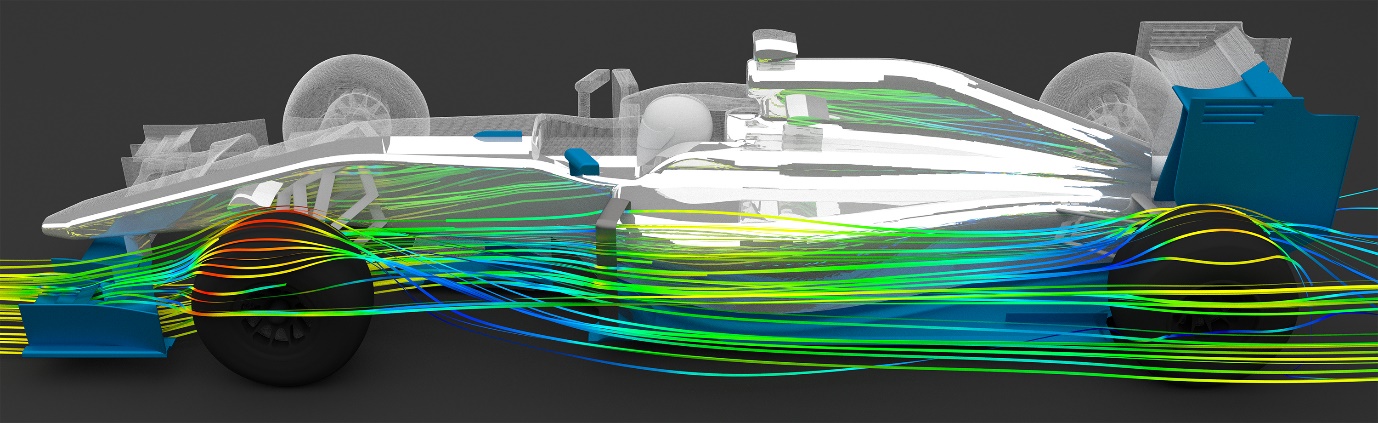
**Fluid Simulation and Physics (Unity)**

A research paper by Ryan Pallesen

Fluid simulation is a broad term looking at the simulations of fluids or fluid-like substances on a computer. Fluids are defined as any container-filling substance, such as liquids, gasses and plasma.  
  
Some of the main worries with fluid simulation and physics, particularly for game-engines, are performance, ability to scale, viscosity and accuracy

This report will be focusing on an implementation of Smoothed-Particle Hydrodynamics in the unity game-engine, with Box2D handling collision detection and resolution, as Smoothed-Particle Hydrodynamics are more suitable for games due to them being significantly more performant, though not more accurate.

Figure 1: Streamlines around an F1 Car

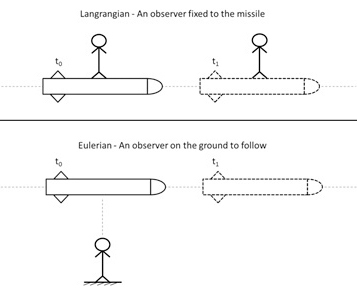


# Fluid Simulation Types & Focus

There are two main forms of fluid simulation methodologies, “*Langrangian* or *Eulerian*.” [1][2]

Langrangian fluid simulation is also know as the ‘discreteve’ methodology, or ‘smoothed-particle hydrodynamics’ and ‘SPH’, in which fluid is treated as a mass of “discrete blobs of fluid” [3] with their own mass, velocity and shape.

Eulerian fluid simulation is also known as the ‘continuous’ methodology, or ‘grid-based hydrodynamics’. Grid-based hydrodynamics “are typically highly accurate, although relatively slow compared to particle based solutions”[3]. They work by using a space partitioning (Usually grid-based, but sometimes cell-based), in which each grid square contains information about the total velocity, mass and pressure for the fluid in that grid square.

Figure 2: Langrangian vs Eulerian

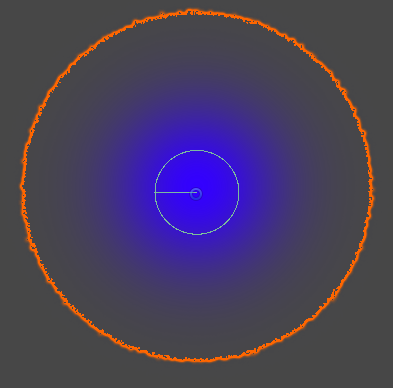
# Basics of the simulation setup

The basics of this simulation are to spawn a large amount of 2D circles with colliders and allow standard collision detection and resolution from Box2D.

The colliders are given 0 friction and 0 bounciness, allowing to slide freely between other particles of the same fluid, and simulating real-life fluid’s lack of bounciness

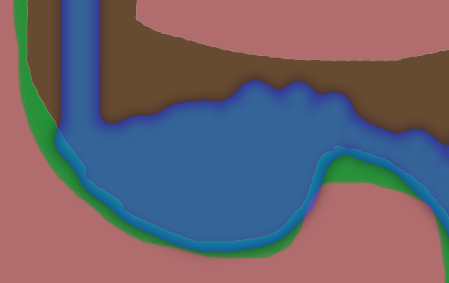
Figure 2: Physics Material Setup

Next, the fluid particle itself is given graphics that go beyond it’s collider in order to create more of a ‘blob’-like effect, rather than it appearing as a bunch of circles.

Figure 3: Collider [Green] vs Image bounds [Orange]

And finally, many ‘fluid particles’ are spawned into the scene together and allowed to collide.

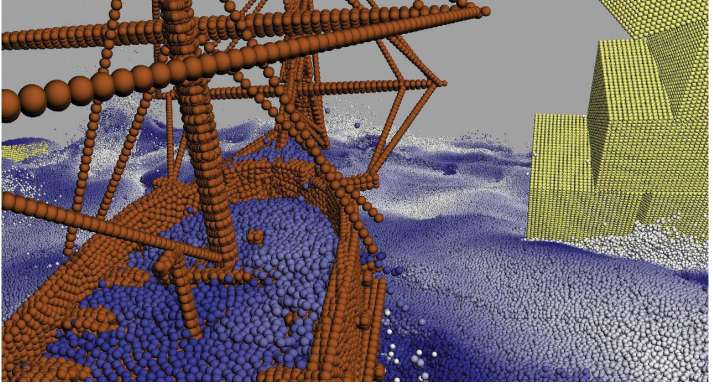
Figure 4: Particle Fluid Collision



# Further Understanding

While great for 2D games, once this technique is translated into 3D there can be significant issues with performance, such as this 50 second clip with “10 million fluid + 4 million rigid particles [that had a] 50 h computation time on a 16-core PC” [4] by Freiburg university.

Figure 5: exposed colliders in a fluid simulation

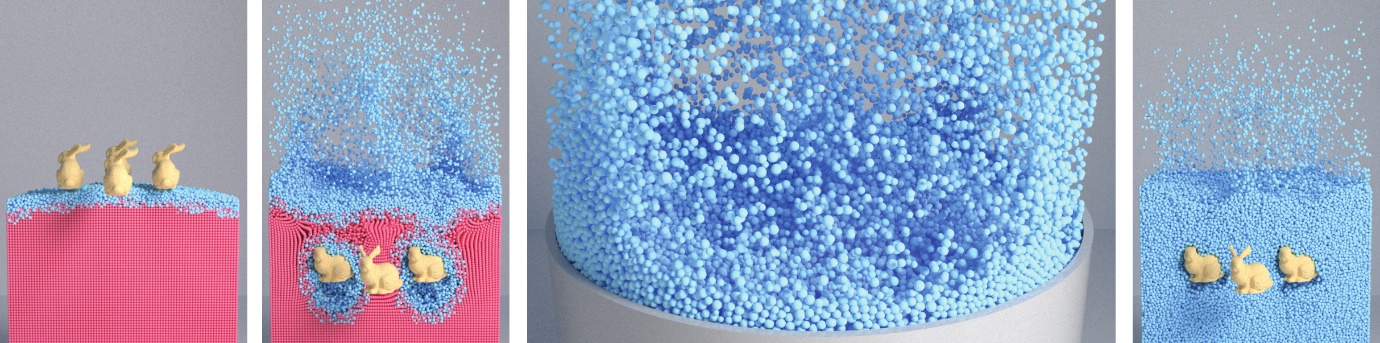


As This is far too computationally expensive to be run in a real-time 3D game, Commonly, games do not use fluid simulations, but instead ‘fake’ fluids using animations.

Many optimizations have been proposed in order to make real-time large-scale fluid simulations a viable choice. These optimizations include threading, estimations, compute shaders and complete reworks and redefinitions of the existing methodologies and algorithms that are being used.

The perhaps most popular of these optimizations is the ‘Hybrid Grains’ Method, that “Exploits the dual strengths of discrete and continuum treatments” [5].  
  
In this form of simulation, unused, or unlikely to be changed, particles are not running any computationally expensive algorithms, optimizing away the significant number of the concurrent collisions that occur constantly in a standard Smoothed-Particle Hydrodynamics simulation.

Figure 6: Hybrid Grains demonstration



Another common issue with fluid simulation is ability to correctly simulate viscosity of fluids, which is being solved using “A Geometrically Consistent Viscous Fluid Solver with Two-Way Fluid-Solid Coupling” [6]. This methodology utilizes a Eulerian simulation “for simulating viscous materials and their interactions with solid objects.” [6]

Figure 7: Three gears interacting with viscous fluids



Viscous fluids are still very computationally expensive, and not very feasible for any form of real-time simulation or rendering, particularly in a game engine with many other scripts keeping the computer busy. Figure 7 clearly shows a very realistic rendering of refraction and reflection alongside the high fidelity simulation. As such this form of simulation is ideal for physical simulations for engineering, and for pre-rendered productions.

# Conclusion

There are many forms of fluid simulation and many ways to modify it to suit your needs.

Langrangian simulation is best for real-time approximations of fluids in 2d or 3d settings, but can become significantly imperformant at larger scales, despite still being notably more performant than eulerian solutions.

Eulerian solutions are significantly more accurate than Langrangian simulations, making them optimal for realistic renderings and simulations for engineering or data-gathering and prototyping, but not the optimal choice for real-time simulations in games.

Luckily, despite Langrangian simulations being imperformant at larger scales, there are many tricks to optimize and deal with this imperformance, allowing larger scale fluid simulations to be feasible in real-time game applications.

Eulerian simulations also allow for significantly easier representation of viscous fluid coupling and decoupling for realistic simulations than Langrangian simulations would, which extends it’s importance in engineering and rendering rather than in games development.

# References

[1] Brochu T, Batty C, Bridson R. (2010) *Matching Fluid Simulation Elements to Surface Geometry and Topology.* Available at : <https://www.cs.ubc.ca/~rbridson/docs/brochu-siggraph2010-voronoi.pdf> [Accessed 2 Mar. 2020]

[2] Simscale.com. (2020). *What is CFD | Computational Fluid Dynamics? — SimScale Documentation*. [online] Available at: https://www.simscale.com/docs/content/simwiki/cfd/whatiscfd.html [Accessed 2 Mar. 2020].

[3] Braley C, Sandu A. (Unknown) *Fluid Simulation For Computer Graphics: A Tutorial in Grid Based and Particle Based Methods* Available at : http://users.encs.concordia.ca/~grogono/Graphics/fluid-5.pdf [Accessed 2 Mar. 2020]

[4] Teschner, M. (Unknown) *Simulation in Computer Graphics: Particle-based Fluid Simulation* Available at : <https://cg.informatik.uni-freiburg.de/course_notes/sim_10_sph.pdf> [Accessed 2 Mar. 2020]

[5] Yue Y, Smith B, Yichen Chen P, [Chantharayukhonthorn](http://web.mit.edu/kkamrin/www/people.html) M, Kamrin K, Grinspun E. (2019) *Hybrid Grains: Adaptive Coupling of Discrete and Continuum Simulations of Granular Media* Available at : http://www.cs.columbia.edu/~smith/hybrid\_grains/ [Accessed 2 Mar. 2020]

[5] Takahashi T, Lin M C. (2019) *A Geometrically Consistent Viscous Fluid Solver with Two-Way Fluid-Solid Coupling* Available at: gamma.cs.unc.edu/ViscTwoway/ [Accessed 2 Mar. 2020]