

# Intractable Fluid Simulations in VR

Andy Danforth | VR Systems CSE493V Spring 2023 University Of Washington

## Overview

Realistic fluid simulations can serve as a great visualization tool and are a vital part of many visual effects in movies and games. I wanted to try and implement my own realistic fluid simulation in virtual reality and to have it update in real time in order to make full use of the medium via user interaction.

## Method

There are two typical ways fluids are simulated: Eulerian grids and Lagrangian particles. Eulerian methods keep track of fluid variables on a fixed grid, and Lagrangian methods store fluid variables on individual particles. I chose a Fluid-Implicit-Particle (FLIP) method as described by Zhu and Bridson which combines the two prior methods by storing position and velocity on the particles, transferring these values to a staggered grid for calculations and then back to particles [1].

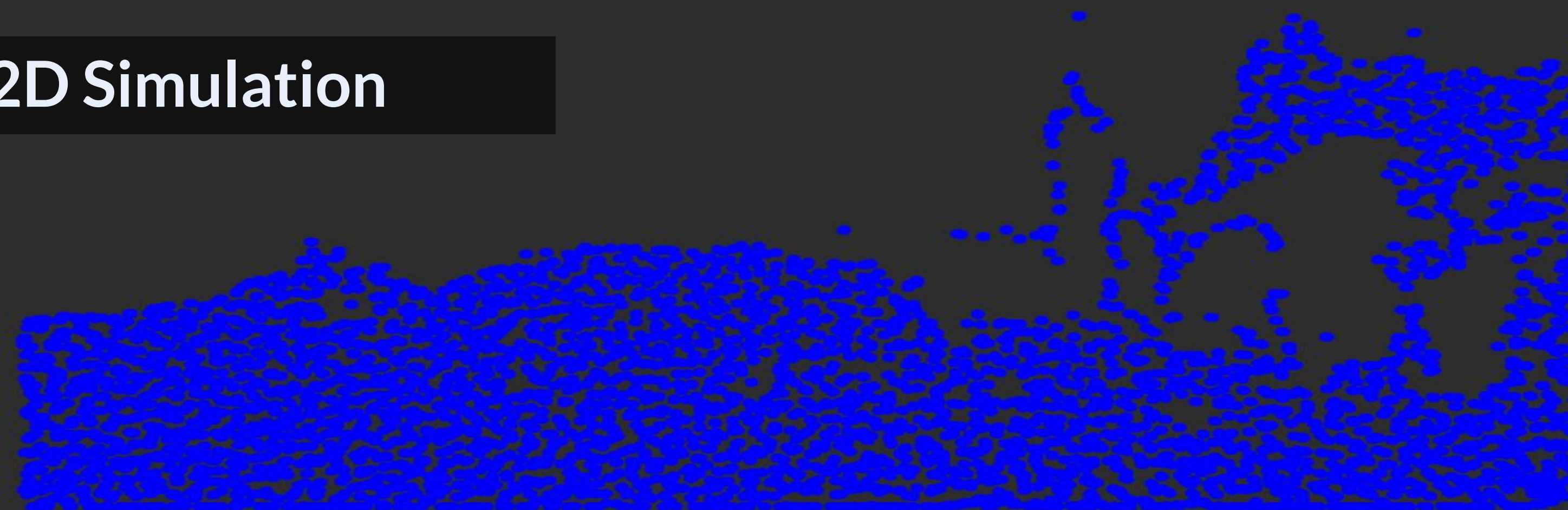
## Results

Some of my results are shown on the left

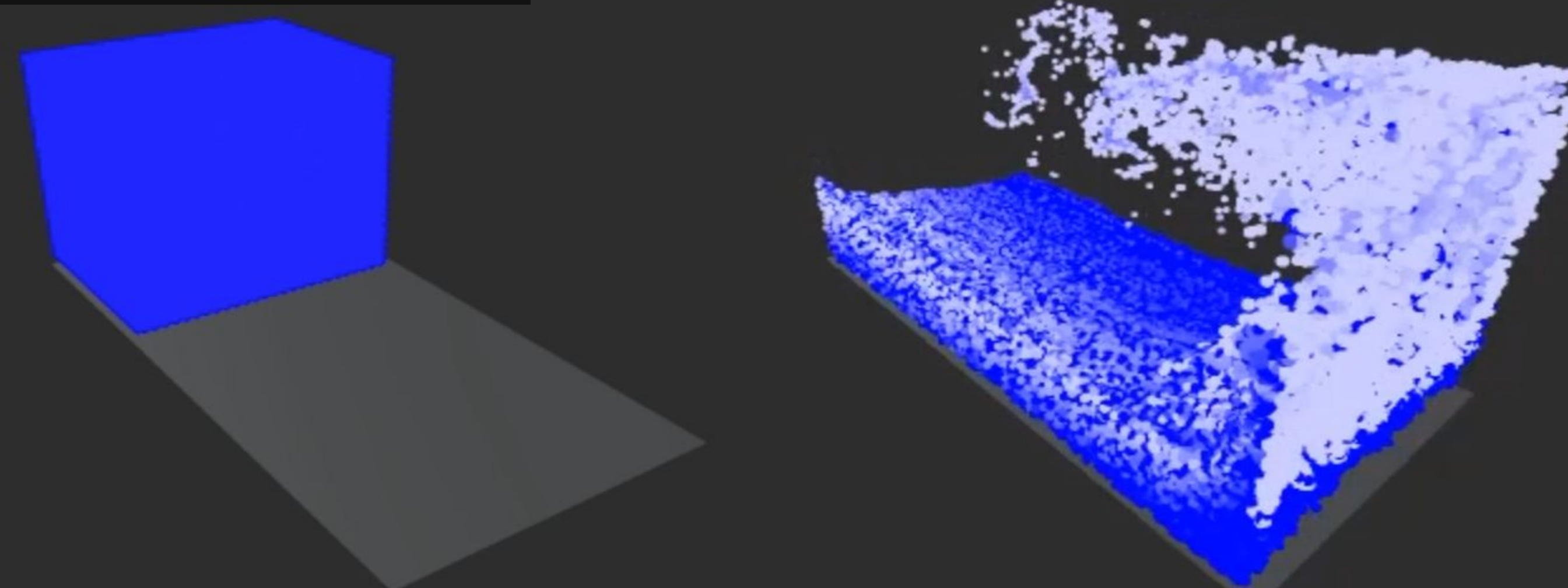
- The first image shows my 2D fluid simulation mid run
- The second pair of images shows a “Dam Break” set up where all of the particles are set up in a wall and once the sim starts, they will crash into the opposing side
- The third pair of images shows the 3D rendering in Stereo for VR compatibility, the left and right images are shown to their respective eyes.

Making a fluid simulation work without exploding or collapsing in on itself is harder than I wanted it to be. (It still kinda doesn't work)

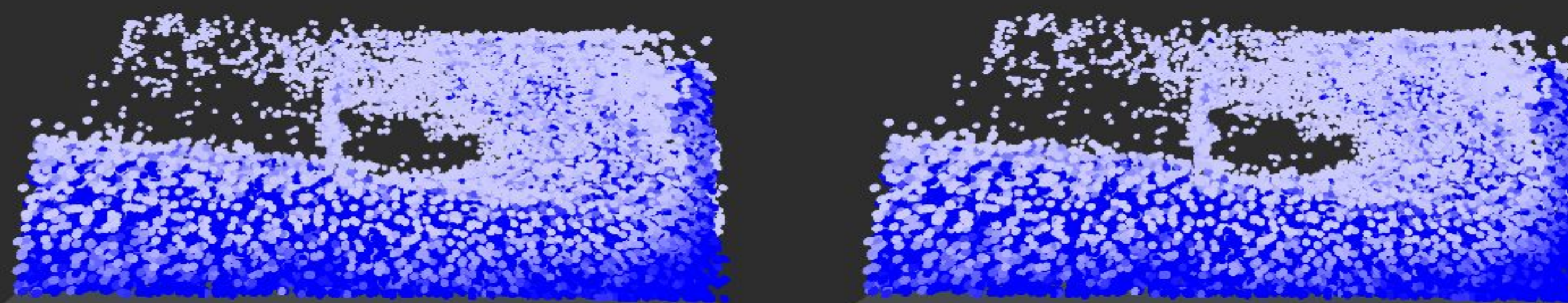
### 2D Simulation



### 3D Simulation



### Stereo Rendering for VR



## My Approach

I was familiar with Unity, and knew Unity had good VR support, so I opted to use it for this project. Unlike most existing fluid simulations, I will be using Unity and its rendering system. This imposes a larger overhead than if I implemented it all directly using a graphics library so I had to take the following considerations:

- Implement all rendering using GPU instancing to improve performance [2].
- Implement all simulation steps using the GPU/compute shaders.
- Define custom shaders to do stereo rendering for VR.
- Add satisfying interaction methods using oculus controllers.

## Related Work

- Matthias Müller's 2D Flip Implementation  
A very well done tutorial on the basics of FLIP simulations and basic working code in JavaScript. [3]
- David Li's 3D Flip Demo [4]  
A well made 3D demo of a Flip solver With mouse interactions.
- Zhu and Bridson's paper Animating Sand as a Fluid  
A well referenced paper which discusses the FLIP method in detail

## References

- [1] Yongning Zhu and Robert Bridson. 2005. Animating sand as a fluid. *ACM Trans. Graph.* 24, 3 (July 2005), 965–972. <https://doi.org/10.1145/1073204.1073298>
- [2] Spencer, M. 2023. *How to render 13,086,178 objects at 120 fps*, YouTube. Available at: [https://www.youtube.com/watch?v=6mNj3M1il\\_c](https://www.youtube.com/watch?v=6mNj3M1il_c)
- [3] Matthias Müller. 2022. 18 - how to write a flip water / fluid simulation running in your browser.
- [4] David Li. 2016. <http://david.li/fluid>