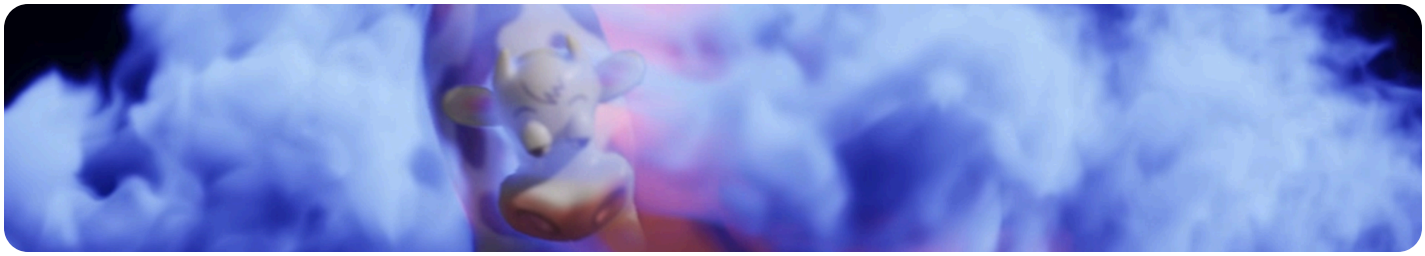
Fluid Simulation Overview

Overview of Fluid Simulation in Unreal Engine.



Learn to use this **Beta** feature, but use caution when shipping with it.

Unreal Engine 5 includes a set of tools for simulating fluid effects in real time.

This toolset is designed to be artist-friendly and includes a variety of GPU-based simulators, reusable modules, and robust data structures that are all usable within the Niagara Editor. Advanced users can take advantage of the wide range of exposed parameters to modify the simulation to their needs.

The fluid simulation system is designed to generate complex fluid effects for real-time environments that can be used in games and cinematics. The system can also be used to bake out complex simulations to flipbook textures for a variety of use cases.

Intended Audience

The various fluid (gas and liquid) simulators are designed to be useful to everyone, from graphics researchers to effects artists. Here are some applicable use cases for different

types of users:

Effects Artist

- Quickly place systems in a map for specific game elements or cinematics shots.
- Edit, keyframe, or drive various User Parameters exposed by the actor via Blueprints.
- Tweak a few system parameters to achieve the desired look.
- Add appropriate forces and collision objects so the fluid responds to the world.
- Make simple modifications to how fluid is injected into the fluid solver.

Effects Developers

- Build custom systems starting from predefined templates to achieve the desired look.
- Incorporate new forces, sourcing methods, or custom boundary conditions.
- Add and modify existing modules in fluid and sourcing emitters within a system.
- Profile and make changes to improve performance for specific systems.
- Build or modify systems and expose user parameters to empower artists on the team.

R&D Developers

- Design complex fluid behaviors by writing HLSL shader code to extend the base emitters.
- Modify all aspects of the simulation algorithm by modifying or replacing existing modules.
- Try new algorithms and quickly test them in the Niagara Editor or maps.
- Build new systems and package parameters with User Parameters and Summary View.

Key Concepts of Fluid Simulation

Overview

Fluid simulation is the process of algorithmically generating data that represents the motion of fluids, such as gases or liquids. This simulation data can either be represented as grids or particles, depending on the algorithms used.

The Unreal Engine Fluid Simulation system uses grids to simulate gases, and a mixture of particles and grids for liquids.

Grids

Gas simulations are represented by a grid, in which each cell contains data representing the density, temperature, and velocity of the medium at that location. Smaller grid cells sizes result in a higher quality simulation, however this increased quality comes with an increase in computational cost.

When rendering a smoke simulation, you are generally visualizing the density grid. Areas that have higher density are more opaque compared to areas with lower density. Fire simulations function similarly, where temperature controls the color of the fire in each grid cell. Within fire simulations, higher temperatures cause gas to rise faster due to buoyancy.

Fluid Motion

One of the main components in simulating fluids with realistic movement, is the process of "solving for pressure". This technique involves solving a system of equations that are designed to ensure the fluid correctly flows around objects and results in a realistic swirling motion.

This simulation involves an iterative process, with more iterations resulting in a more accurate simulation. However, more iterations will result in a higher computational cost. It is important to note that sometimes the technique of "pressure solving" can affect forces added to a simulation and dampen their effect.

Collision Objects

In order to have the fluid respond to nearby objects, you must first set up what is known as "boundary conditions".

Setting boundary conditions for the simulation involves masking out areas where objects are located within the simulation area, to ensure that no fluid can occupy the same space. These collision objects can move around in the scene and their velocity is used to displace the fluid around them.

A common example involves an object falling into a pool of liquid, causing splashes and ripples around the object. The system can use many object types as colliders, such as static meshes, Geometry Collections, and depth maps.

Liquids

Niagara Fluids use a hybrid particle and grid algorithm called FLIP (Fluid-Implicit-Particle) to simulate liquids, such as flowing water. In this system, the velocity of the fluid is solved on a grid, and then sampled back to particles that represent the shape of the fluid.

Note that there are several similarities between liquid and gas simulations, so the previous sections on collision objects and fluid motion apply to liquids as well.

Key Niagara Concepts

All the fluid simulators are accessible from the Niagara Editor, and make use of a variety of Niagara features.

> To learn more about the Niagara VFX System, visit the documentation.

Grid Data Interfaces

Grid 2D Collection and **Grid 3D Collection** are data interfaces for storing named attributes in 2D and 3D grids, respectively. These are used for all the required computations when solving fluids.

Simulation Stages

For each step in the gas or liquid simulation algorithm, the Unreal Engine Fluid Simulation system makes sure all grid cells are processed before moving on to the next step. The system uses what are known as **Simulation Stages** within Niagara to do this. Note that Simulation

Stages can be run several times before proceeding to the next stage by modifying the stage's **Num Iterations** attribute.

Reusable Modules

Much of the fluid behavior is defined in modules, which can be moved around in the Niagara System stack. Some modules are used across 2D, 3D, gas, and liquid simulators, and are generally written to be very generic. Advanced users can modify core fluid solver behavior by modifying or replacing various modules.

Fluid Parameters

All of the template systems exposed in the **New Niagara System** menu have user parameters that allow for modification on the actor when placed in a map. Values can be edited directly in the **Details** panel. There are more parameters exposed on the fluid emitter's **Summary View**, which can be modified in the Niagara editor.

Simulator Types

2D Gas

This is a gas simulation where flow only occurs in a two-dimensional space. These simulations are generally much faster than 3D simulations and are best suited for games and real-time use. However, they generally lack the complex turbulent flow that 3D simulations can exhibit.

It is important to note that 3D objects can interact with 2D simulations. These types of simulations can be set to always be oriented toward the camera and mimic 3D behavior. This is commonly used when creating torches or other flame effects where depth is necessary.

2D gas simulations are rendered on planes that are either camera facing or oriented in the world.

3D Gas

This is the most common type of gas simulation. Compared to its 2D counterpart, 3D Gas simulates depth and a more complex flow at a higher memory and GPU cost.

This type of simulator is best used for hero effects in real-time applications, and for cinematics. The results can also be baked into textures for increased performance in real time.

Gas simulations that use baked lighting accomplish self-shadowing by baking the shadows into a grid. This data is then passed to the material for rendering. Ray marching materials are responsible for generating images of the volume by accumulating visible density and temperature.

2D FLIP

2D FLIP samples 3D particles and generates a 2D simulation domain where pressure is solved and the FLIP algorithm is applied to update the particles. Typically, the domain is camera aligned, which allows for complex 3D-looking flow, despite the use of 2D fluid forces. This can be used for simulating convincing splash effects, but not for pools of water.

2D FLIP simulations are generally rendered as 3D particles.

Shallow Water

Unlike 2D FLIP, Shallow Water is useful for simulating pools of water that don't contain many splash effects. This can be used to simulate boat wakes, or simple interactions with objects moving through the water.

Shallow Water simulations are height fields that are rendered as displacements to a water surface.

3D FLIP

3D FLIP simulations are robust liquid simulations that are capable of simulating a wide range of effects, such as waves on a beach, rivers, and complex object interactions. Note that these complex simulations can be computationally expensive. These are particle simulations where velocities are determined by a grid solver that ensures the fluid doesn't compress.

3D FLIP simulations are rendered by splatting particles into a grid, then rendering the grid as a surface using ray marching.