

Liquid Simulator

This page introduces the LiquidSim object and its rollouts.

Overview

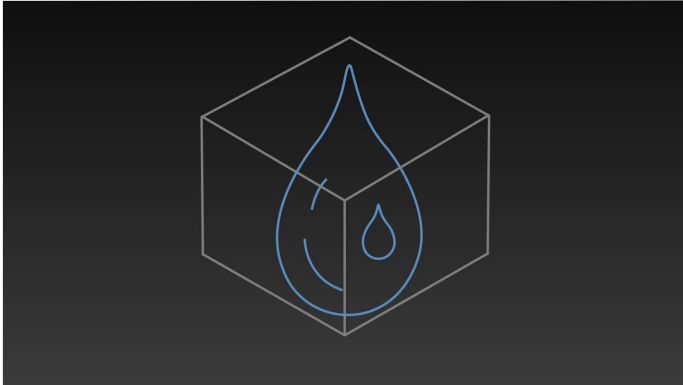
Liquid Simulations can create pouring or flowing liquids, or any simulation that needs foam or mist, such as coffee or waterfalls.

You can think of the simulator as a 3D box, inside which simulations of fluids and Rigid Body Dynamics are performed. The simulator consists of voxels that contain the fluid's properties at a position and given time, such as the fluid's velocity, its viscosity, its color, and so forth.

The fluid properties are written inside what are called Grid Channels. Each channel stores a type of value, such as Liquid or Velocity, with its own [Grid Channel Range](#), which is the range of possible values that is most efficient for that specific channel type. Phoenix determines the fluid's behavior at a given time, based on the content of these Grid Channels.

Liquid Simulators also generate particles for liquids that drive the simulation, by writing simulation data inside each voxel's Grid Channels, so that the Grid Data is built from those particles.

Since Phoenix 3.0, Liquid Simulations have used a hybrid solver under the hood, called FLIP, which provides more realistic and faster simulations, with no stepping artifacts. In essence, the FLIP solver combines a Grid simulator with Particles, in order to take advantage of the benefits of both.



Note that the simulator is represented as a single object for convenience. Internally, it contains two completely separate parts: a simulator component, and a rendering component responsible for the process of converting the Liquid particles into a renderable surface. Parameters that control the simulation are separated from those associated with the meshing and rendering process. As a result, no rollout will contain mixed parameters, and no parameter will affect both the simulation and rendering.

Phoenix generates particles for liquids because they are useful for representing the characteristics and behavior of a fluid, and as a result look more natural when rendering liquids, especially when generated in very large amounts. The particles emulate real-world fluids by moving through 3D space within the simulation grid, and unlike voxels, each particle carries its own fluid properties with it, such as Position, Age, Velocity and so forth. These properties contained in each of the particles are called Particle Channels, and their values can change every frame, based on each channel's Particle Channel Range.

Besides Liquid Particles, the simulator provides built-in helper simulators for Foam, Wetmap, Splash, and Mist Particles. In certain cases, the splash and mist can be considered as part of the liquid simulation, because liquid, splash and mist can be converted to each other during the simulation process. You can go through the [FLIP Particles Life Cycle](#) page for an in-depth explanation of this process.

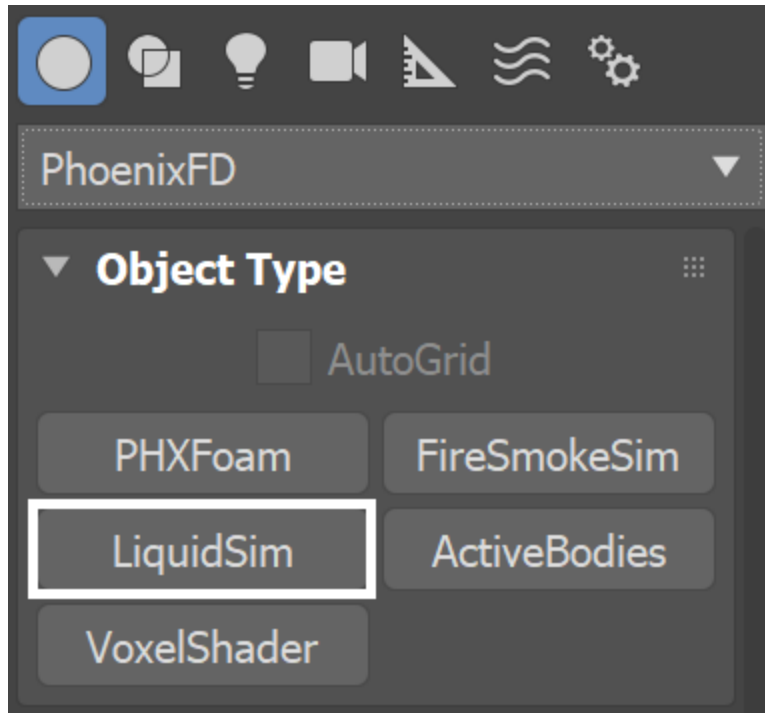
When running a simulation, the Liquid sim will typically output an entire animation sequence. That sequence is saved as individual files, called caches, which contain raw simulation data for each timeline frame. When the Liquid Simulator saves its caches, the Liquid Particles get turned into grids, so in essence, the cache data contains grids and particles, which describe the fluid's behavior.

Liquid sim cache files can then be previewed in the viewport, as well as rendered, in order to create captivating and realistic images. The simulator also offers multiple render modes, where the mesh mode is typically used to render Liquid simulations, and the volumetric render modes are typically used for rendering Fire/Smoke simulations. Phoenix offers a lot of flexibility though, so there is the option to render Liquid simulations as volumes and Fire/Smoke simulations as meshes, in order to achieve even more diverse effects.

When rendering other particle types besides Phoenix Liquid Particles, Phoenix also has a separate [Particle Shader](#) geometry type, which is used to shade particles.

The Particle Shader component is most often used in conjunction with the Liquid Simulator to shade particles such as Splash, Mist and Foam. However, it can also shade particles created from the Fire/Smoke Simulator, as well as standard particle systems such as Particle Flow, or particles from plugins like tyFlow, thinkingParticles, Krakatoa, etc. Note that while it is a geometry type in 3ds Max, the Particle Shader is not a simulator.

UI Path: `[[Create panel]] > Geometry > PhoenixFD category > LiquidSim button`



Rollouts

After creating a LiquidSim object, the following rollouts can be accessed in the Modify panel:

- [Simulation](#) - Controls the simulation and displays statistics such as simulation times, as well as information about the data contained in the sim for the current frame.
- [Simulation Speed](#) - Helps identifying the heaviest computation phases of the simulation and provides tips on how to get results faster and more efficiently.
- [Resimulation](#) - Modifies an existing cache sequence to enhance features or change specified particle systems.
- [Grid](#) - Controls the boundaries, size and resolution of the simulation grid.
- [Dynamics](#) - Offers parameters to control the fluid's behavior when simulating, such as air effects, viscosity, surface tension, drying time, etc.
- [Foam](#) - Controls the simulation of bubbles and foam.
- [Splash | Mist](#) - Controls the simulation of splash and mist.
- [Scene Interaction](#) - Specifies how the LiquidSim interacts with other objects in the scene.
- [Output](#) - Specifies how and where the simulation cache files are saved.
- [Input](#) - Determines the path of the input files for rendering and previewing, and offers settings and controls for playback..
- [Preview](#) - Controls how the simulation is displayed in the viewport.
- [Rendering](#) - Controls how the simulation is rendered.
- [Export](#) - Controls how meshes and particles are exported from Phoenix.
- [Authorization](#) - Displays information when license information cannot be obtained.

Upgrading from Previous Versions of Phoenix

Due to many improvements to the Phoenix solver, recreating simulations the same way you did in older versions of Phoenix may not be possible because of changes to the solvers and the UI. Here are some things to be aware of when upgrading from older versions of Phoenix.

- If you have a scene created with Phoenix version 3 and you open it with version 4, it will keep all the old settings for you, including some which are not visible to newer versions, in order for the results to be roughly the same.
- If you try to manually replicate an older simulation setup with a new Phoenix simulator using Phoenix version 4, and set all the parameters with the same values as the older version, you may not obtain the same result as in Phoenix version 3 or older.
- Since the release of Phoenix version 3, Phoenix started using a new solver for liquids and fire/smoke simulations, so simulation results will be different between Phoenix version 3 and version 2.