Example Scenes

The following samples illustrate the usage of different features in Chaos Phoenix.

Coffee

Scene (108 MB)

This scene demonstrates how to set up a coffee scene using Phoenix. The **Liquid Source** for the coffee emission is using a Discharge modifier that modifies the **Outgoing Velocity** on the negative **Normal Z**. This way the liquid is emitted only by the parts of the emitter that are facing downwards.

The scene is built in real world scale and the coffee cup is around **7 cm** tall. The Phoenix Simulator is using a **Scene Scale** of **10** and the **Steps per frame** are set to **15** in order to account for the fast movement of the liquid.

In the Foam settings a dummy box geometry, that is non-solid and non-renderable, is used as a **Birth Volume** in order for the foam particles to be born only near the bottom of the coffee cup. The **Foam Volume** is set to **100** which is a balanced value, in order for the foam to stack up in a thick layer on top of the liquid.

A **Voxel Shader** is used to read the **Speed channel** of the liquid and render it as smoke. The faster the liquid is moving the denser the smoke will be. This emulates foam emission of tiny foam particles when the liquid is moving fast enough without having to simulate millions of foam particles. Then the liquid is used as a Cutter Geometry for the smoke so that only the parts that are inside of the liquid are rendered.

Software used: **Phoenix 5.10.00 Official Release, V-Ray 6, hotfix 3 Official Release, 3ds Max 2018**

Pool

Scene (151 MB)

This scene demonstrates how to set up a pool scene using Phoenix. The pool and the ocean simulator used for the background are both using Pure Ocean mode, which creates a flat ocean surface up to the **Ocean Level** height. It does not need loaded caches and if there are any, it ignores their content, so no simulation details will show. Thus changing frames and generating the ocean surface is very quick. A Phoenix Ocean texture is used to displace the pool and ocean surfaces and give them the characteristic water surface look.

For the rendering part, the scene uses the Progressive caustics sampler to create the caustics effect in the pool.

Software used: **Phoenix 5.01.00 Official Release, V-Ray 6, hotfix 3 Official Release, 3ds Max 2018**

Rocket

Scenes (245 MB)

This scene demonstrates how to set up a rocket launch scene using Phoenix. A Fire Source in **Volume Inject** mode is used for the smoke emission from the rocket and the boosters. Another two Fire Sources in **Volume Brush** mode are used to change the **RGB** color of the smoke near the ground to give it more variation. Since the scene has a lot of geometry and we wish only a few parts of it to interact with the simulation, the Scene Interaction is set to Include list and only the objects relevant to the simulation are picked. For the rendering the Smoke Scattering is set to **Ray-traced** in order to get more realistic scattering of the light through the smoke. The **Phase Function** for the smoke is set to **0.7** so that the light can scatter more and give the lighter smoke steam appearance. The scene contains a primitive called *"Blocker"* which is a **Solid Object** until Frame 74 and holds the smoke generated by the Boosters below the Launchpad, after Frame 75 the Blocker is set to a **Non-Solid Object** and the smoke can start travel upwards.

The attached example scene with higher resolution has a starting **Grid Resolution** of **12 million cells**, **Voxel Size** of **0.203m** and **Adaptive Grid** set to **Smoke**, so the Grid will automatically expand when it's needed. The **Grid Resolution** reaches **1.35 billion cells** in its final stage at **Frame 300**.

The attached example scene with lower resolution has a starting **Grid Resolution** of **1.7 million cells**, **Voxel Size** of **0.397m** and **Adaptive Grid** set to **Smoke**, so the Grid will automatically expand when it's needed. The **Grid Resolution** reaches **413 million cells** in its final stage at **Frame 300**.

The example videos below demonstrate the results with two **Grid Resolutions** scaled to **1.35 billion cells** and **413 million cells**.

Software used: **Phoenix 5.01.00 Official Release, V-Ray 6, hotfix 2 Official Release, 3ds Max 2018**

Grid Resolution of around 1.35 billion cells, simulated on a machine with 256GB of RAM

Grid Resolution of around 413 million cells, simulated on a machine with 64GB of RAM

Static Clouds

Scene (0.99 MB)

This scene demonstrates how to set up a static clouds scene using Phoenix. A Fire Source in Volume Brush mode is filling the cloud shaped emitter geometry over time. The smoke channel is mapped with a noise texture in order to give the cloud shape more randomized and wispy look. The Input is set to Cache Index mode so that a single cache file will be used through the whole animation sequnce. For the rendering the Smoke scattering is set to Ray-traced in order to get more realistic scattering of the light through the clouds.

Software used: **Phoenix 4.41.00, V-Ray 5 Update 1.3, 3ds Max 2018**

Sink

Scene (1.46 MB)

This scene demonstrates how to set up a simple sink scene using Phoenix. There are two Liquid sources, one for the faucet and another one using negative Outgoing velocity in order to consume some liquid and prevent the sink from filling up. In order for the sink to be filled with some liquid at the start, a simple box is used with the Initial Fill option enabled in its Phoenix Properties. The steps per frame are set to 12 in order to compensate for the fast moving liquid particles.

Software used: **Phoenix 4.40.00, V-Ray 5 Update 1.2, 3ds Max 2018**

Boiling Liquid using the Particle Tuner

Scene (287 kB)

This scene demonstrates how to setup boiling liquid with foam, where the foam size is based on the distance to a certain object using Phoenix. The scene uses dummy non-renderable geometry to fill the teapot with liquid at the start of the simulation using the Initial Liquid Fill option.

A Liquid Source in Volume Inject mode, using pFlow particles as an emitter is used to stir up the liquid and create the boiling effect.

The Foam particles are enabled in the Simulator. Then in the Output rollout of the Simulator the particle Velocity, ID, Age, RGB and Size channels for the Foam are enabled.

There are 5 Particle Tuners in the scene. The first two change the color of the foam particles based on their age. The third Particle Tuner takes the red foam particles that are inside of a text object and have an age of over two seconds and makes them bigger.

The fourth Particle Tuner makes all the foam particles outside of the text object smaller. Finally the fifth Particle Tuner sets the Velocity on the Z axis for the bigger foam particles to 0 - preventing them from bouncing up and down.

Software used: **Phoenix 4.40.00**, **V-Ray 5 Update 1, 3ds Max 2018**

Underwater Explosion

Scene (1.36 MB)

This scene demonstrates how to setup an underwater explosion using Phoenix. The scene uses two animated Liquid Sources in Volume Inject mode to get more interesting shape of the explosion. Each emitter has different geometry and animation for the Inject Power.

The Grid resolution is crucial for this setup. It controls the amount of particles emitted through the source and thus makes the explosion bigger or smaller. The scene scale is lowered to 0.5 to make the simulation a little bit faster in terms of speed for the water and the particles movement. The Foam and Splash particles are enabled in the Simulator.

In the Splash settings some of the particles are converted to Mist and the Foam on hit is set to 1 so that when the splashes collide with the liquid they will create foam.

In the Foam settings the rising and the falling speed of the foam contribute to foam movement and the large scale look of the explosion. To get a more interesting look for the foam the Foam Patterns are set to 0.4.

Additionally Phoenix Plain Force and a Turbulence Force are added to enhance the movement of the mist.

Software used: **Phoenix 5.01.00 Nightly from 23.08.2022**, **V-Ray 6 Hotfix 2, 3ds Max 2018**

thinkingParticles Explosion

Scene (36 MB)

This scene demonstrates how to use the Phoenix operators inside of thinkingParticles. The Phoenix TP Birth operator creates particles based on the smoke channel of the Phoenix simulation. Then the created particles are advected using the velocity data from the Phoenix simulation through the Phoenix TP Force operator. Finally the Phoenix TP Sample operator reads the data from the Phoenix simulation and uses the Speed channel to set the Size variation of the tP particles. The Speed data from the Phoenix simulation is also passed to the Vertex color of a cube geometry, used as a Shape instance in the scene.

For the rendering part, the particle material is using Phoenix Grid Texture that reads the Fire color from the Phoenix simulation and sends it to the Self-Illumination slot of a V-Ray Material. For the Diffuse part of the shader - V-Ray Comp texture is used to multiply a concrete texture with the Vertex color data.

Software used: **Phoenix 4.20.00**, **thinkingParticles V6.8.166**, **V-Ray Next Update 3, 3ds Max 2018**

Shower

Scene (7.73 MB)

This scene demonstrates how to set up a simple shower scene using Phoenix. The shower nozzles are added to the Liquid source with some noise for the **Outgoing Velocity** in order to randomize the emission. The steps per frame are set to 10 in order to compensate for the fast moving liquid particles.

Software used: **Phoenix 3.12.00**, **V-Ray Next, 3ds Max 2015**

Fountain

Scene (1.1 MB)

This scene demonstrates how to set up a simple fountain scene using Phoenix. There are four different sources with added noise for the **Outgoing velocity** in order to randomize the emission. The rendering of the Liquid simulator is disabled and the liquid particles are rendered as points using the **Phoenix Particle Shader**. For the ground material a **Phoenix Particle Texture** which uses the **Wetmap** particles is used as a mask to blend between a dry and wet material.

Software used: **Phoenix 3.10.00**, **V-Ray 3.60.04, 3ds Max 2015**

Beach waves

Scene (87 MB)

This scene demonstrates how to use the Phoenix Wave Force to create simulated waves on a shore. The simulated waves create Splash particles which in turn create Foam particles by using the **Foam On Hit** parameter of the Splash particles. Other important settings for the setup are the **Droplets Surfing** option which is enabled so that waves would slide upon the water surface instead of directly mixing with the water volume, and also the **Foam Patterns** which help create a more diverse surface of the foam left behind by the waves. The Foam **Rising Speed** is tuned to 35 cm/sec so the Foam remains underwater for a short while and can be tinted using the water material's fog color.

The Foam and Splash particles are rendered using the Phoenix Particle Shader in Point mode, which is the fastest particle render mode and is recommended for large scale scenes where individual bubbles are not visible and vast volumes of particles must be rendered. The settings are tuned in such a way that you can quickly switch to **Bubble** mode for the Foam and **Splash** mode for the Splash particles which are a bit more realistic but will take much longer to render. The Point **Shadow Strength** is boosted to 3.0 so the volume of the foam volume stands out and the foam is not rendered flat. The Point **Alpha** is lowered to 0.1 so individual foam particles don't pop up in the render as bright points, and only larger masses of foam are rendered more opaque. The **Volume Light Cache** of the Particle Shader is also enabled and uses a high **Light Cache Speedup** in order to improve the render times.

The liquid also creates WetMap particles over the shore geometry which are used to mask wet and dry materials using the Particle Texture. **Mesh Smoothing** is enabled in order to remove noise from the liquid mesh's surface, and the Mesh Smoothing **Particle Size** is increased so the mesh doesn't shrink and reveal air pockets between the liquid and the bottom which will become visible in the rendering. The preview of voxels and the Liquid and WetMap particles is switched off in order to speed up simulation and only the preview of Foam and Splash particles remains enabled. You may re-enable the preview if you want to observe the simulation process, or alternatively, you can speed up the simulation even more by setting **Read Cache for Preview** to **Disable During Sim** from the Preview rollout.

Software used: **Phoenix 3.10.01 nightly (24 Mar 2018)**, **V-Ray 3.60.04, 3ds Max 2014**

Volcano

Scene (48 MB)

This setup uses a few Sources with animated noise textures as masks for the discharge so that the smoke and fire emission are randomized. In order to get a good rolling from the smoke, high **Conservation Quality** is used, along with **PCG Symmetric** conservation.

To add detail to the initial simulation of a relatively low resolution, make sure to enable the Resimulation and run the simulation again.

Software used: **Phoenix 4.41.01 nightly from 5 Oct 2021, V-Ray 5 Update 2, 3ds Max 2018**

Smoke and fire following a path

Scene (0.5 MB)

This setup uses the FollowPath helper in order to guide two separate simulations of smoke and fire along spline curves. The smoke simulation must be run before the fire simulation. Note that the FollowPath force can be used for liquids as well.

Car tire burnout

Scene (0.3 MB)

The tire is made **Solid**. Another cylindrical geometry object is created around the tire in order to drag the smoke around it. The surrounding body is made **n on-Solid** and non-renderable. It is connected to a PHXSource and everything on the source is turned off except for **Motion Velocity** so that the body affects the smoke's velocity when spinning. The surrounding body must be connected to the wheel and spin together with it. The simulator's Object voxels are set to **Inscribed** so that the smoke would enter the real renderable wheel's volume a bit, otherwise, there would be a visible gap between the smoke and the tire. You can control how much the smoke is dragged by the wheel using the Motion Velocity multiplier on the source.

A non-Solid, non-renderable box is placed at the contact patch between the wheel and the ground. It is connected to a second PHXSource and the source is set in **Inject** mode as it discharges smoke with added pressure.

The scene uses classic **Vorticity** for this one. **PCG Symmetric** conservation is used as it is more detailed than Smooth. The **Conservation Quality** is set to *20* so the smoke rolls better. Simulation steps are set to *2 - 1* step is not enough and the smoke starts becoming grainy due to the high velocity, but more than 2 starts to smooth out the smoke a bit too much.

Lava lamp

Scene (0.7 MB)

Three forces are used in the scene. Two BodyForce helpers on the top and bottom of the lamp to give the fluid its vertical motion, and a Turbulence field that adds chaotic changes in the velocity field to break the bubbles apart.

The BodyForce helpers are set up such that each one affects only half the lamp. The bottom one pushes the liquid upwards, and the top one pushes it back down. After a while, the fluid loses its momentum and the system reaches equilibrium. To avoid this, a weak turbulence has been added that prevents the system from balancing and introduces additional fluid splitting forces.
A polygon grid has been added at the bottom of the lamp to help the fluid collect there, just like it does in real Lava Lamps.
The Liquid Source is in Volume Brush Emit Mode, connected to a Sphere. The "Non-Solid" option is enabled on the Sphere for the Volume Brush mode to work.
The discharge parameter is animated - if you'd rather have more/less liquid in the lamp, you can simply move the key along the timeline or input a different value for this parameter.
Play Speed is set to 0.4 to slow down the playback of the simulation.
You can play with the Random Seed value on the Turbulence node to get different looking simulations with little effort.

Liquid morphing

Scene (1 MB)

This scene shows how to shape a liquid into a geometry volume using the BodyForce helper.

Both solid and non-solid modes are supported. When the object is solid, the liquid will be pushed to its surface. When the object is non-solid, the liquid would fill the object. This scene uses non-solid objects which are made non-renderable and their volume is filled. The strength of each force is animated in order to produce the morphing. The forces are activated sequentially and the liquid takes the shape of the currently active force.

Fireplace

Scene (236 kB)

This scene demonstrates how to set up a Fireplace simulation.

For this scene, the Conservation Method is set to Buffered as it produces the best detail for fire simulations. The Steps per Frame option is set to 5 because of the fast motion of the flames. A noise texture is used for the Outgoing Velocity and Temperature slots of the Source so that the fire emission is distributed randomly along the logs' surface which adds more diversity.
For rendering, the Fire opacity mode is set to Use Own Opacity and the render curve is adjusted to bring out the detail of the fire. The Fire opacity is multiplied by a V-Ray Distance texture in order to make the fire transparent near the logs.

Software used: **Phoenix 4.40.00, V-Ray 5 Update 1, 3ds Max 2018**

Chocolate

Scene (0.3 MB)

This example shows how to simulate the process of covering a cookie with chocolate. The parameter that makes the liquid thick is the Viscosity. The bigger the viscosity, the thicker the liquid.

When simulating viscous liquids, you have to enable the Wetting and the Sticky Liquid. Otherwise, the liquid will not stick to the objects. Another important point in this scene is the Mesh smoothing. It is very important to enable Liquid Particles for smoothing, because otherwise, the animation may flicker. To use particle-based smoothing, the liquid particles must be exported. See the Output rollout for more information.

Ink in water

Scene (0.3 MB)

This example demonstrates a technique for rendering thin smoke layers, ink in water, etc. The technique is particle-based and uses the Point mode of the Particle Shader. To achieve good smoothness, more than 50M particles are used. This produces huge cache file sizes of up to 1GB per frame. Thus, the Preview is switched off because loading of the file in the memory can take longer than the simulation itself. You may re-enable the preview if you want to observe the simulation process.

Nuke

Scene (3 MB)

This scene demonstrates how to create a highly symmetrical nuclear mushroom cloud. The setup contains a spherical emitter which creates the fireball, as well as a particle system, created using PFlow which expands in the shape of a ring and creates the blast wave. The scene uses Direct Symmetric Conservation with high Quality in order to produce good rolling of the vortex ring that forms from the fireball, and Massive Vorticity is used in order to give more detail to the smoke.

Wine

Scene (2 MB)

This example shows how to connect two simulators in a cascading way and how to avoid the moving container problem. The scene uses two simulators. You have to run *bottlesim* first and once it finishes, run *glasssim*. The liquid transfer is achieved by setting the first simulator in the **Cascade Source** slot of the glass simulator's Grid rollout.

RGB Explosion

Scene (1 MB)

This setup uses several PFlow particle systems that are connected to separate Phoenix Sources, each one emitting different RGB color. As the explosion unfolds, the colors are mixed in order to produce a more realistic look, as actual explosions usually involve different materials which have different colors as well. A Plain Force helper is used to produce wind which directs the smoke produced by the initial blast sideways.

Looped bubbles

Scene (1.05 MB)

When creating flowing and repeated effects such as fireplaces, campfires or torch fires, water in fountains, waterfalls or boiling liquid you can save a pretty good amount of simulation time by rendering a short looped sequence. In the Input roll-out, simply select the **Loop** mode in the Time Bend Controls and adjust the looped sequence. In this mode, the **Cache Origin** parameter specifies the beginning of the looped sequence, the **Length** parameter specifies the length of the loop, and **Loop Overlap** specifies the number of overlapped frames that ensure smooth transition between the end and the start of the loop. Note that you need to have simulated at least **Cache Origin** + **Length** + **Loop Overlap** cached frames for this mode to work correctly. When looping particles, make sure to export the particle **ID** channel in the Output rollout.

Software used: **Phoenix 4.41.02 Nightly from 02.09.2021, V-Ray 5 Update 1, 3ds Max 2018**