

Exploding Water Balloon

This page provides a tutorial on creating an Exploding Water Balloon simulation in 3ds Max.

Overview

This is an Intermediate Level tutorial. Even though no previous knowledge of Phoenix is required to follow along, re-purposing the setup shown here to another shot may require a deeper understanding of the host platform's tools, and some modifications of the simulation settings.

The instructions on this page guide you through the process of setting up a Bursting Water Balloon simulation. We use a 3ds Max Cloth Modifier to simulate the pressurized balloon tearing apart, and then we use that Cloth simulation as a collider for the Phoenix Liquid. The fluid dynamics governing the behavior of the simulation is explained in detail. Moving on, we adjust the settings of the simulation so to re-create another type of water balloon explosion, which is much more intensive.

Finally, we go over one of the most common issues that arises when simulating physics - interpenetration - and discuss methods for resolving common problems.

[Download Project Files](#)



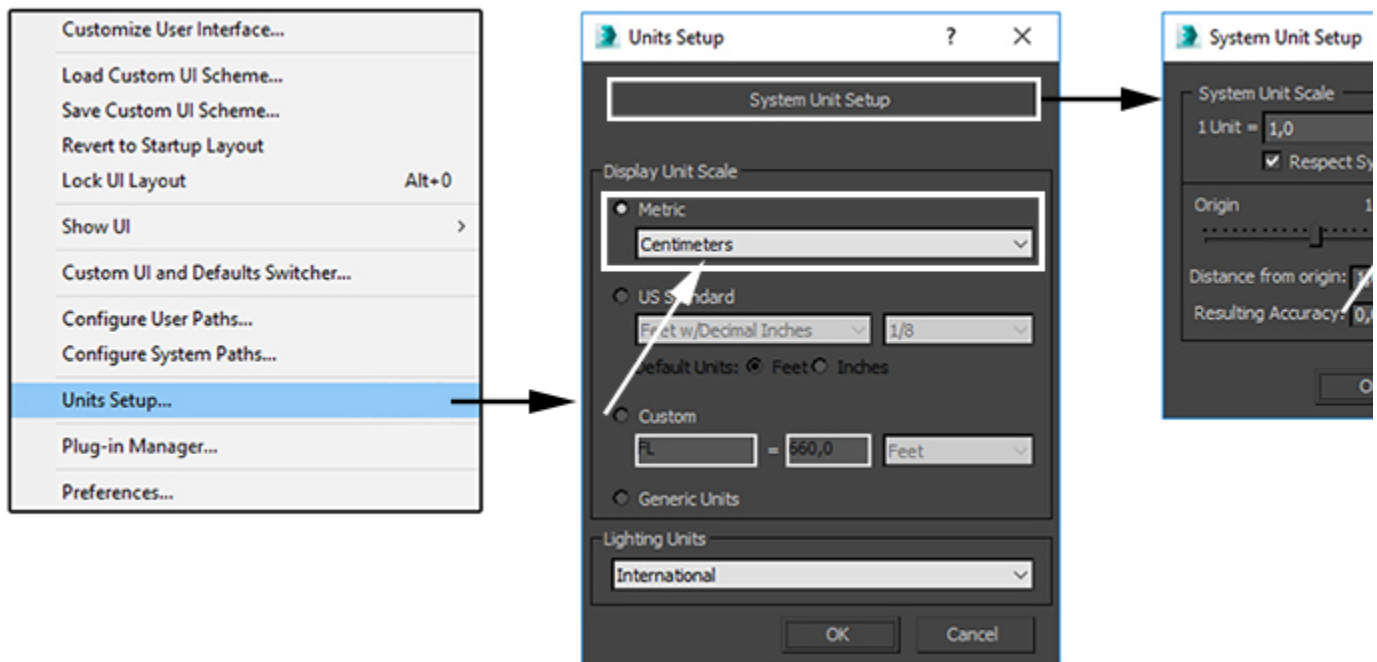
Units Setup

Scale is crucial for the behavior of any simulation. The real-world **size of the Simulator in units** is important for the simulation dynamics. Large-scale simulations appear to move more slowly, while mid-to-small scale simulations have lots of vigorous movement. When you create your Simulator, you must check the **Grid** rollout where the real-world extents of the Simulator are shown. If the size of the Simulator in the scene cannot be changed, you can cheat the solver into working as if the scale is larger or smaller by changing the **Scene Scale** option in the **Grid** rollout.

The Phoenix solver is not affected by how you choose to view the Display Unit Scale - it is just a matter of convenience.

Go to **Customize Units Setup** and set Display Unit Scale to **Metric Centimeters**.

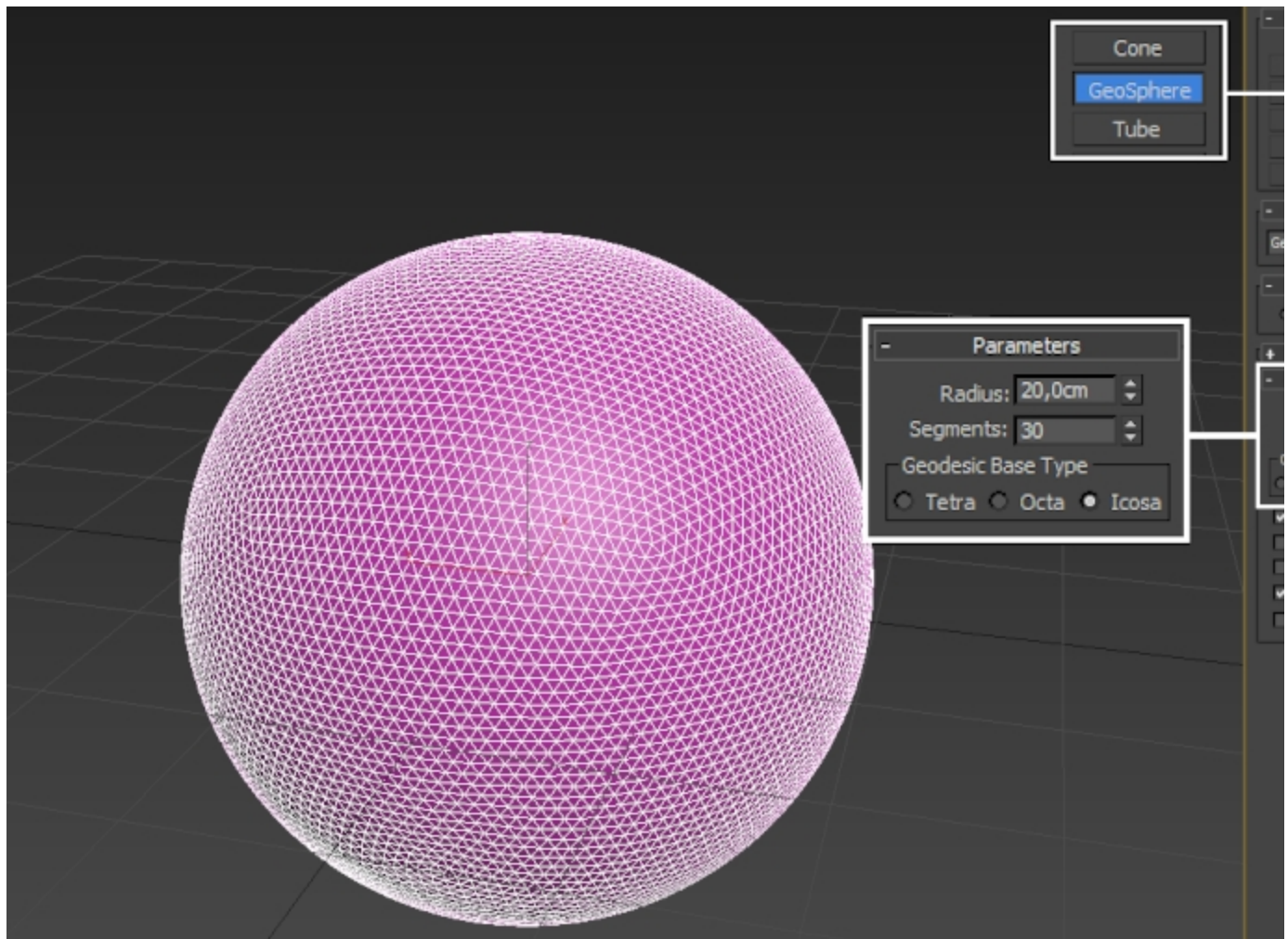
Also, set the **System Units** such that **1 Unit** equals **1 Centimeter**.



Cloth Preparation

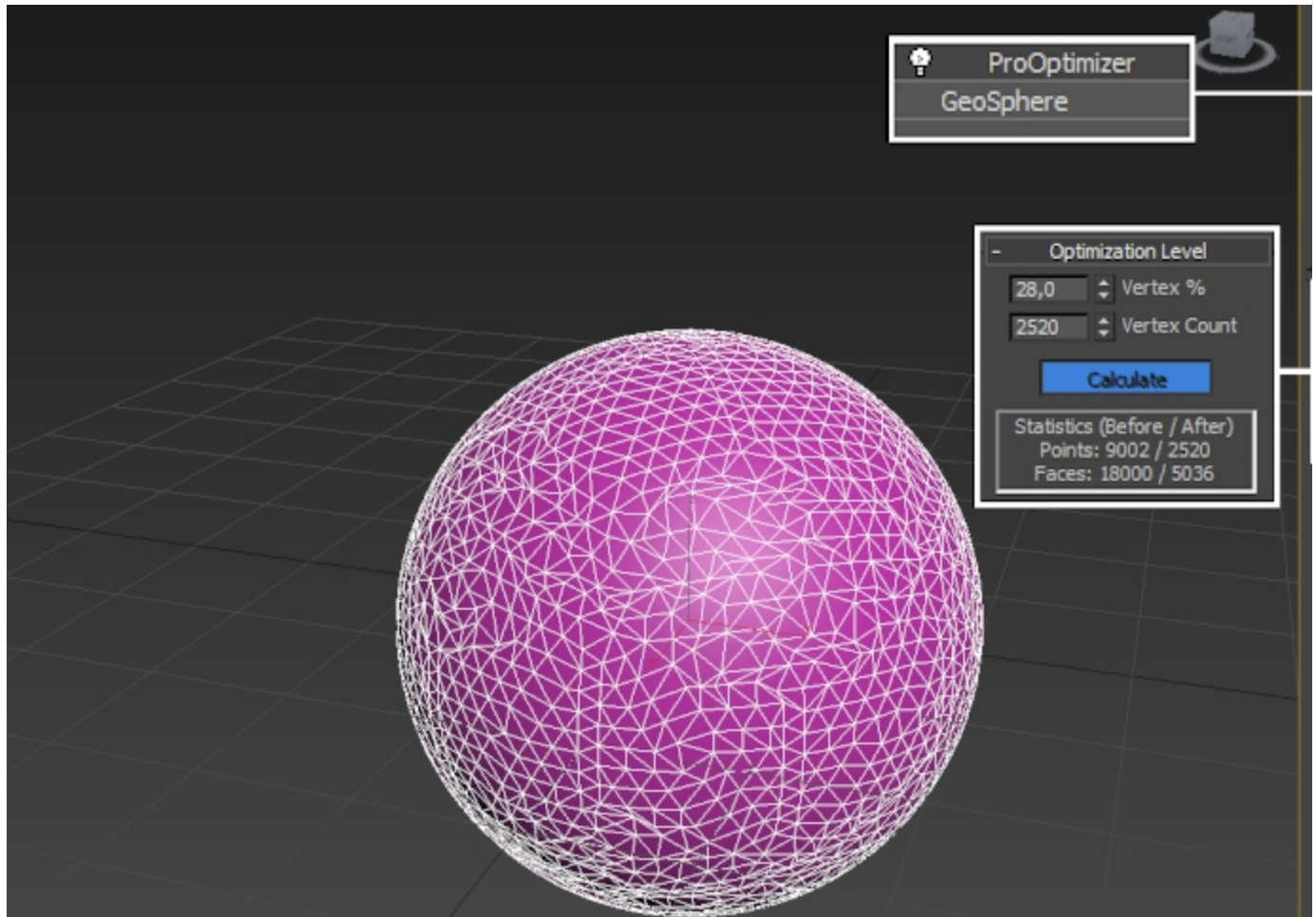
To represent a balloon, we need geometry with topology suitable for cloth simulations.

Create a **Geosphere** primitive with 30 segments and set its **Radius** to **20cm**.

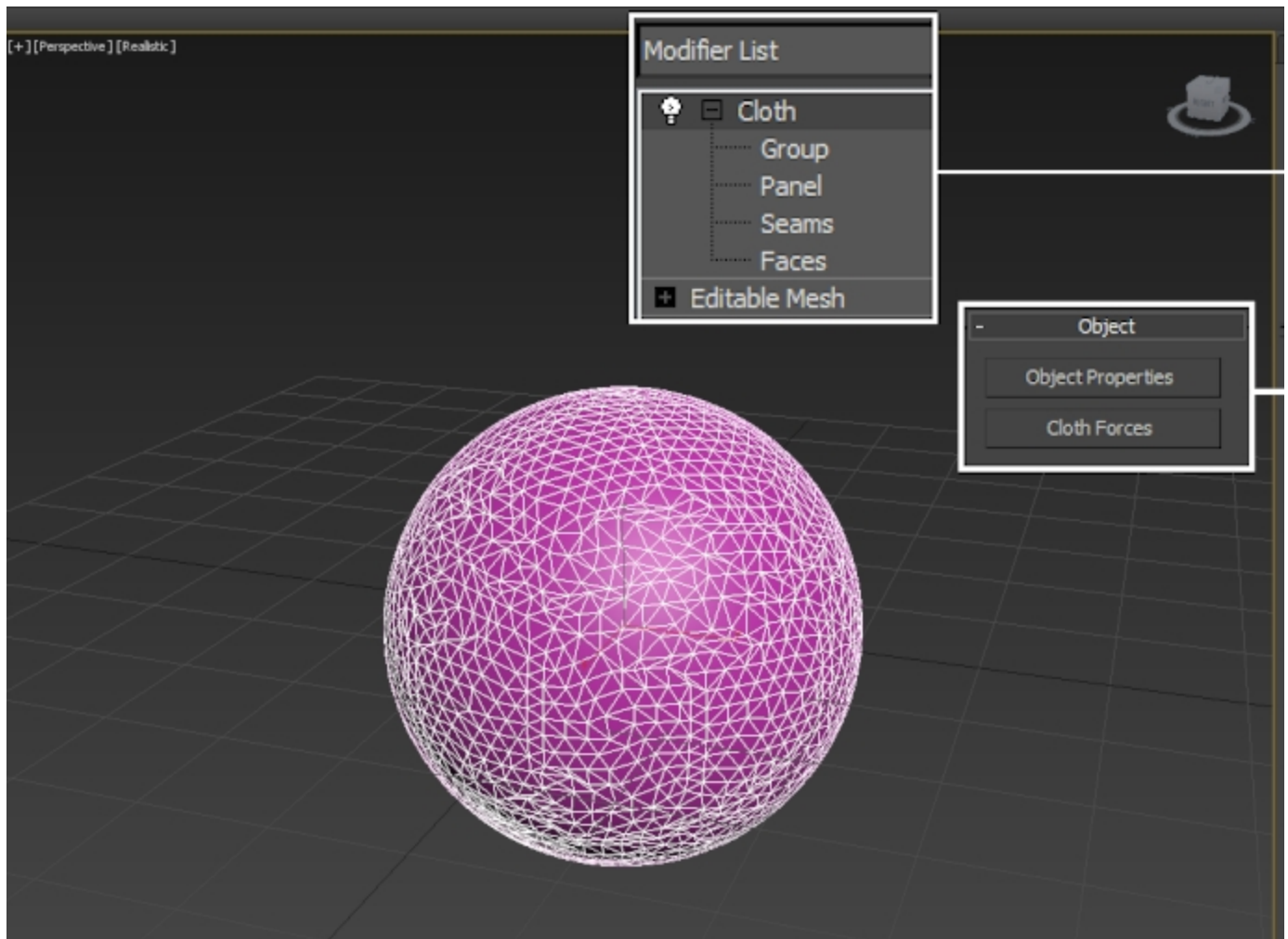


Apply a **ProOptimize** modifier on the geosphere.

Set the **Vertex%** value to around 28 to produce a topological pattern suitable for cloth simulations.



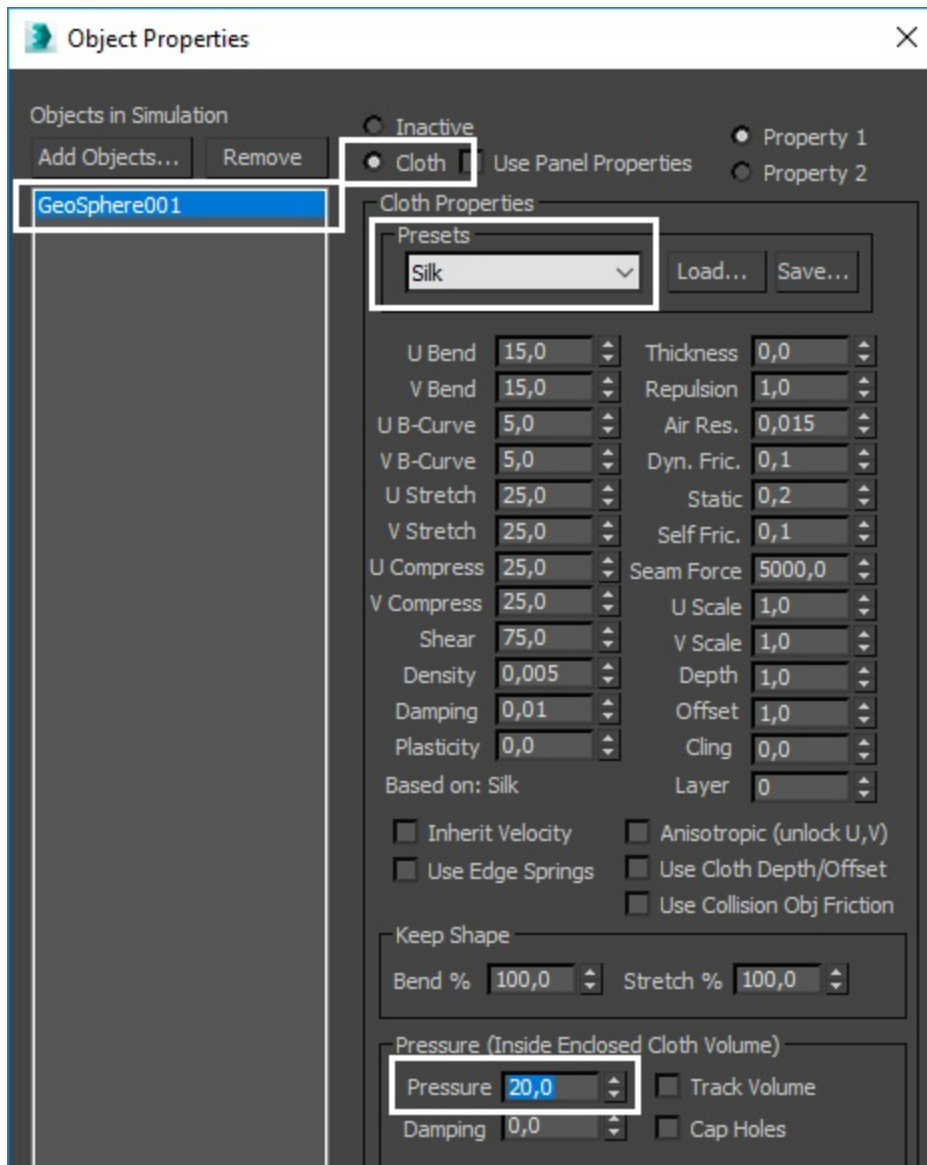
Assign a **Cloth** modifier to the sphere and head over to the **Object Properties** menu.



Select the Geosphere in the list and then set it to **Cloth**. This sets the Geosphere as a dynamic cloth object in the simulation.

Under the **Cloth Properties** panel, set the **Preset** to **Silk**. The Silk preset allows you to easily produce detailed wrinkles. You can experiment with other presets and settings.

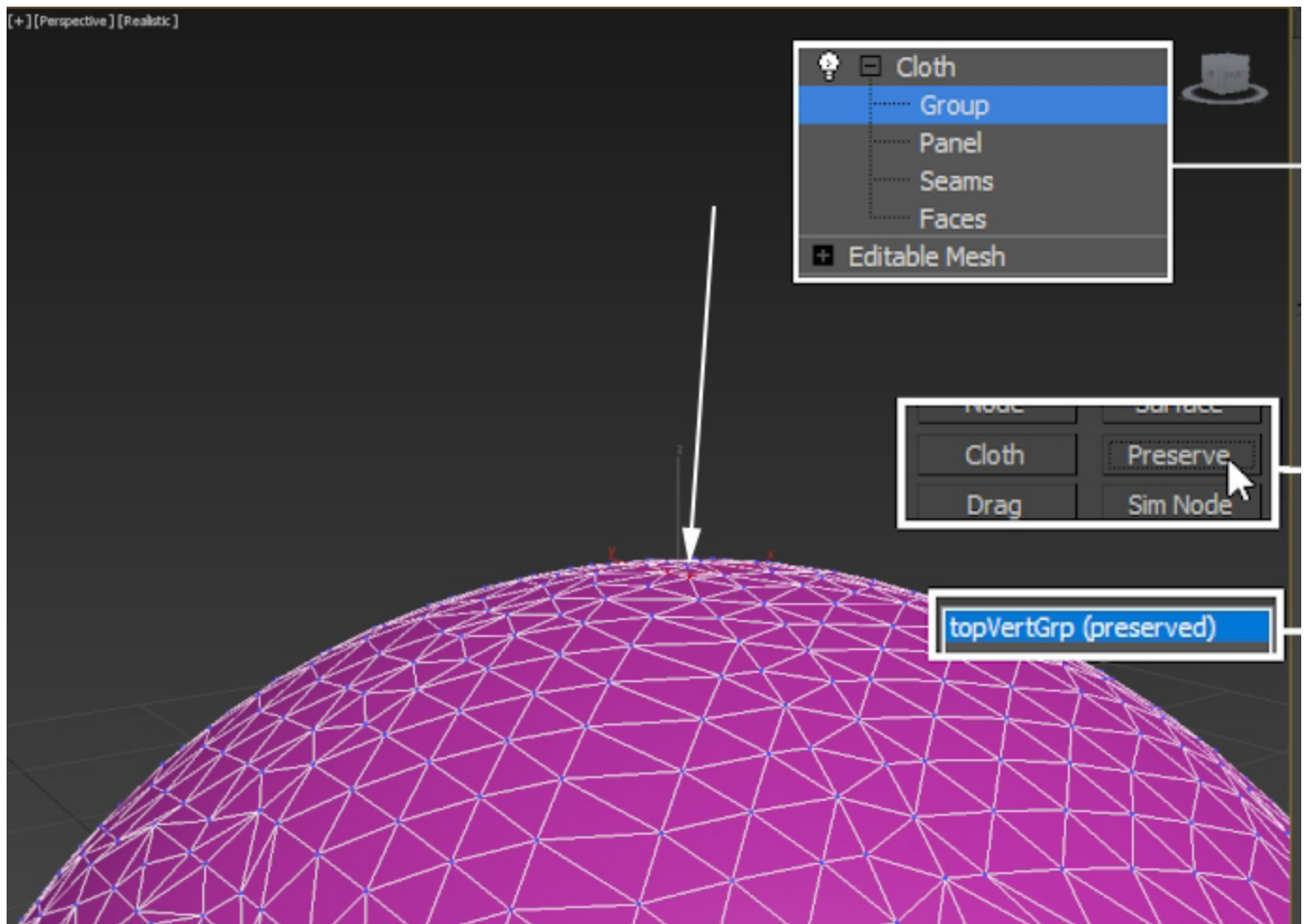
Set **Pressure** to **20**. The Pressure parameter keeps the balloon from deflating and collapsing on itself.



Under the **Cloth** modifier, go to the **Group** sub-object.

Select the **top vertex** of the balloon (as the arrow indicates) and press the **Make Group** button.

Then press **Preserve** so that this vertex won't be affected by the Cloth simulation and the rest of the cloth would hang from it.



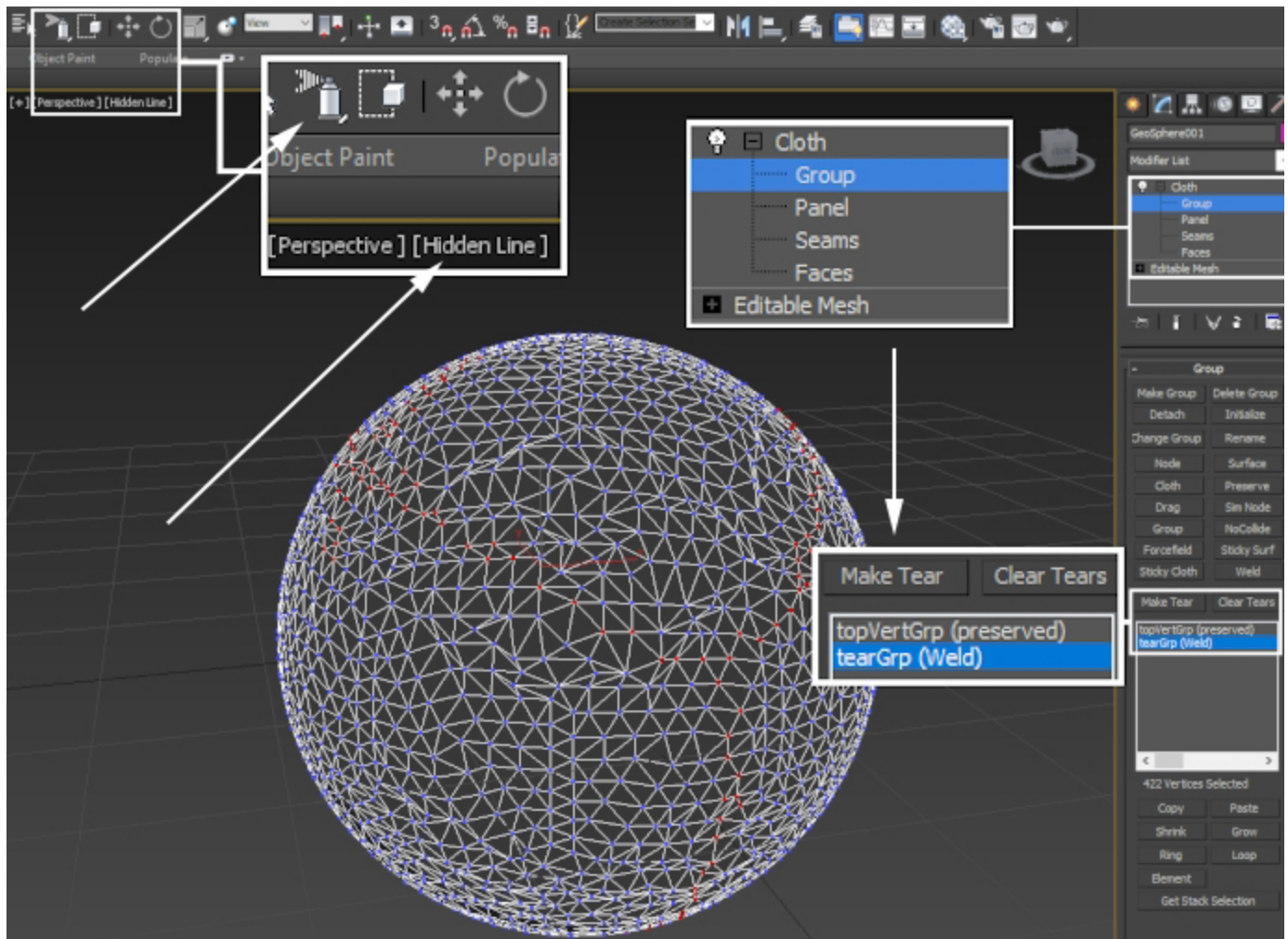
Change your mouse selection mode to Paint Selection. You can control the size of the brush by Right-Mouse-Button clicking on the Paint Selection icon and editing the **Paint Selection Brush Size** parameter from the General tab.

Optionally, you can set the Viewport shading mode to Hidden Line if you find it hard to see your selection.

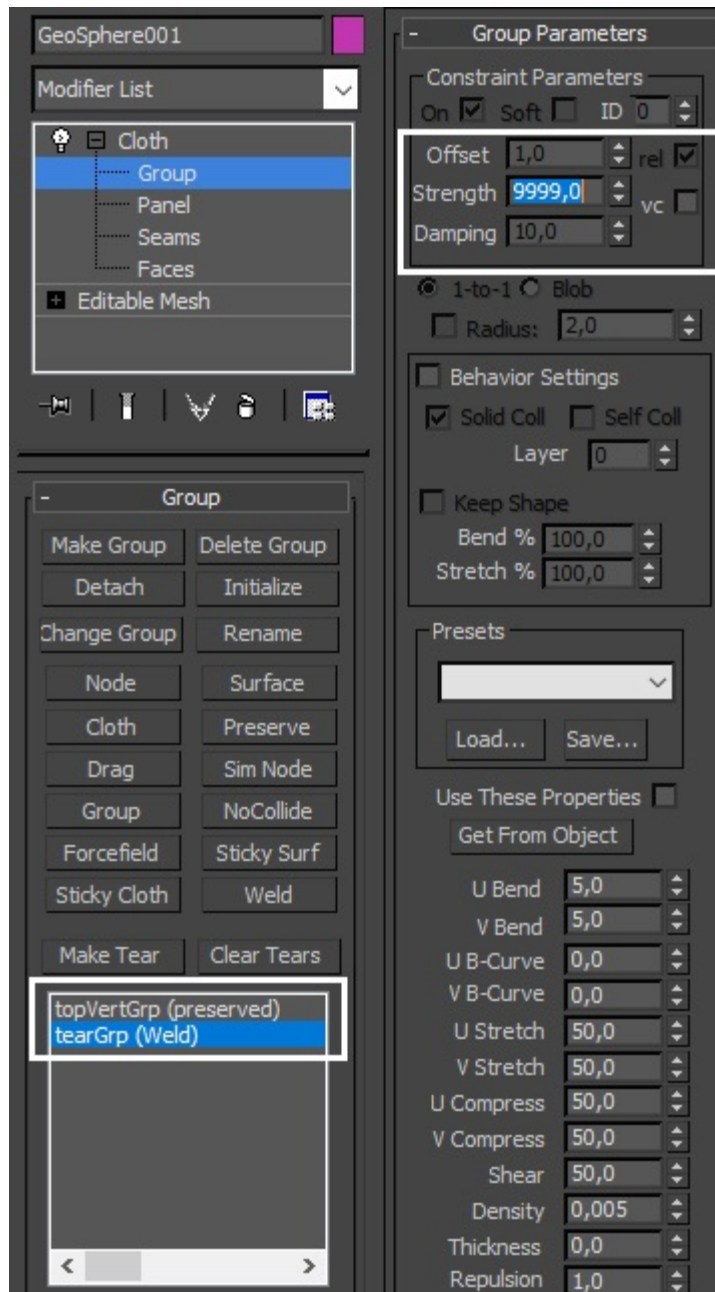
In **Group** sub-object mode, manually select the vertices where you want the balloon to tear apart.

With the vertices selected, press the **Make Tear** button.

This creates a new **Weld Group**. The vertices in that group are split and welded together. Once the **Constraint strength** is not enough to hold them, they will start to break apart.

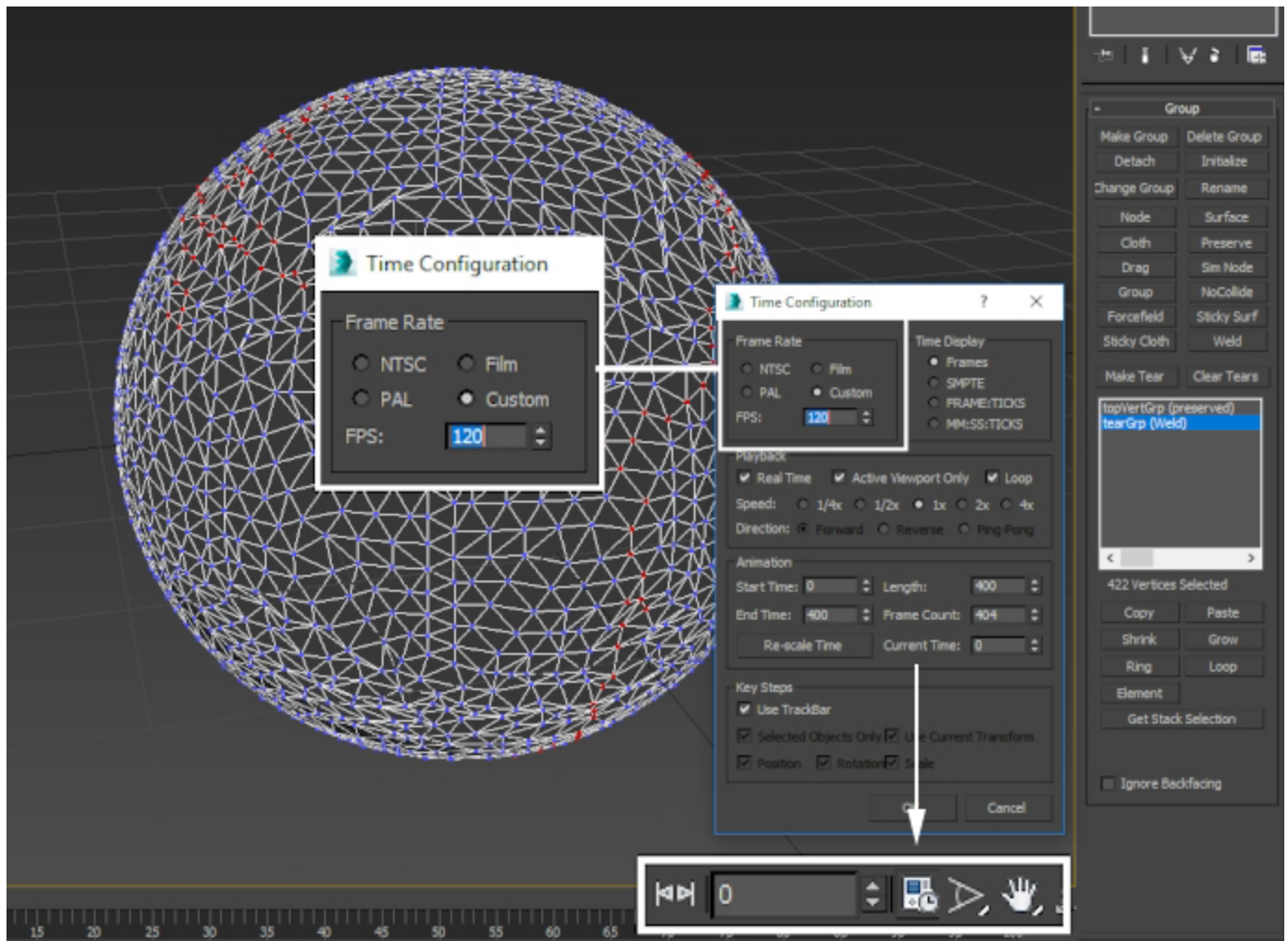


Set **Constraint Strength** to 9999 so that the balloon will hold together at the start of the simulation.



Set the animation **Frame Rate** to **FPS: 120**.

By doing so, we can simulate the cloth in slow motion and speed it up after baking.



Cloth Simulation

Reduce the **Step** parameter to 0.008 or you will get a warning that the Step cannot be greater than frame length. The frame length at 120 FPS is 0.0083(3).

Increase the Scene Scale by setting the **cm/unit** parameter to **10**. This parameter can be used to easily control the "apparent" weight of the balloon.

From the **Simulation Parameters** rollout, set **Subsample** to about 10 - 20.

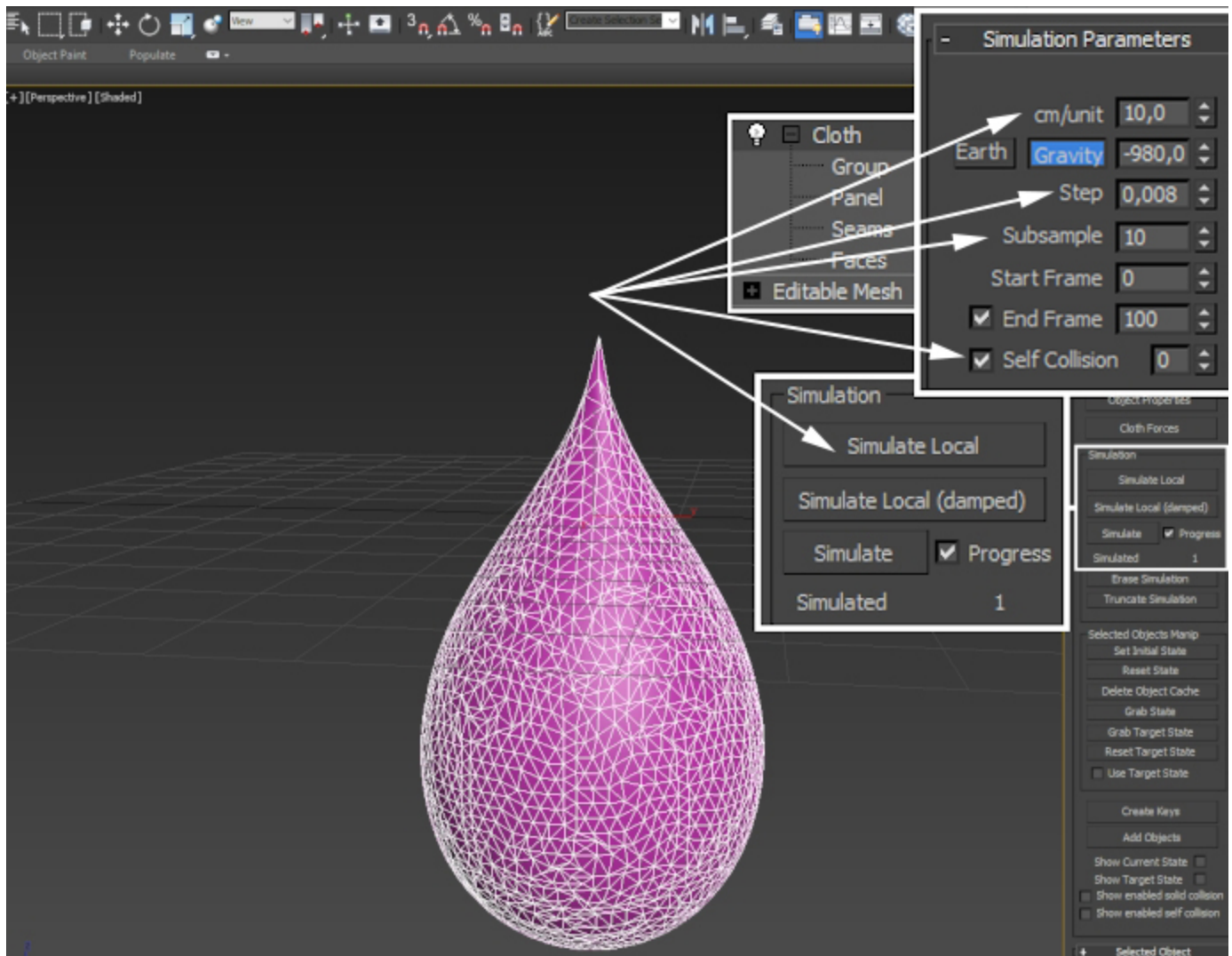
Higher **Subsample** values prevent the cloth from over-stretching.

Enable **Self Collision** to prevent the cloth from intersecting with itself.

Perform a local simulation to allow the balloon to form an initial resting position, as shown on the image.

Click **Simulate Local**.

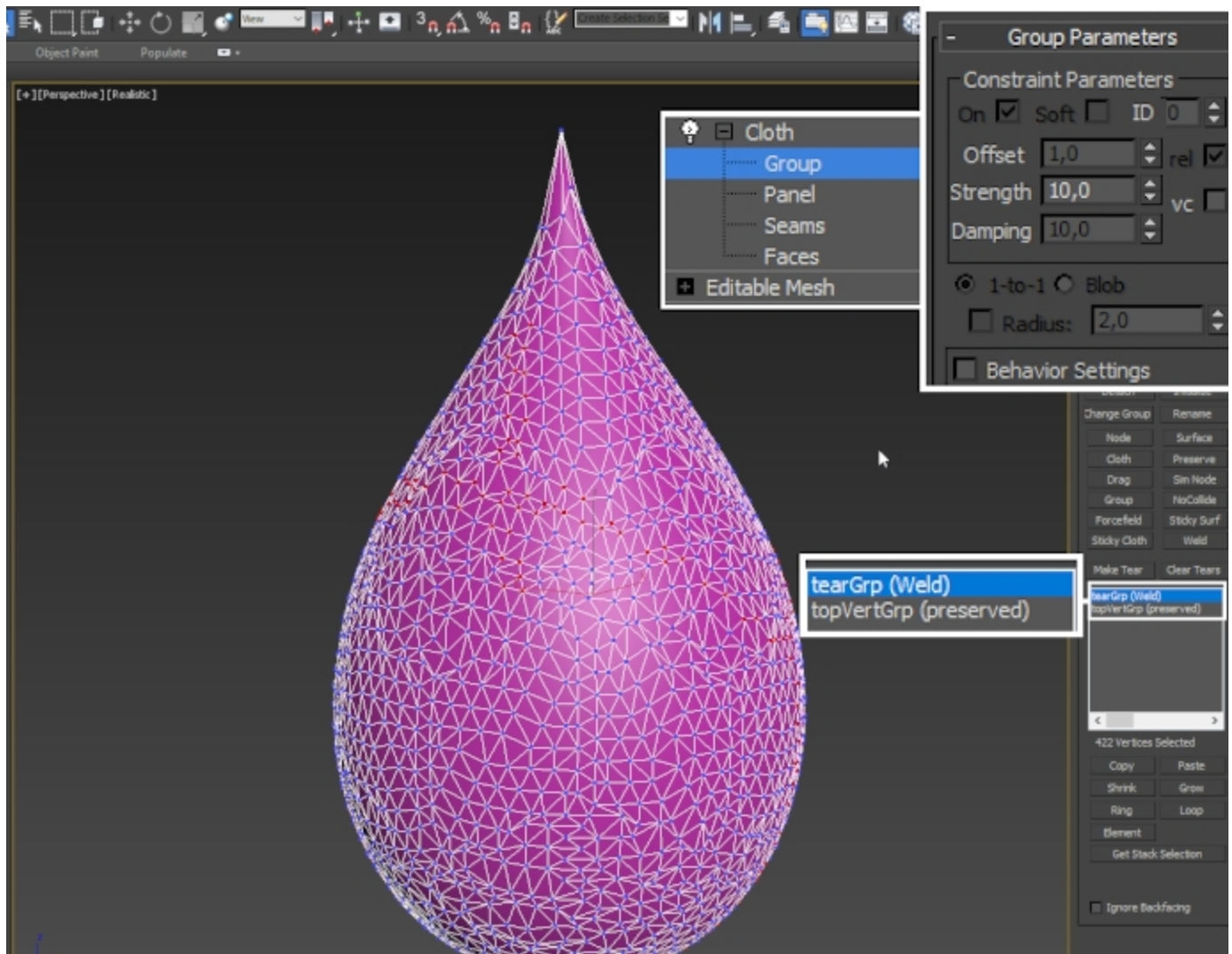
Once the balloon is in the desired shape, we can do a simulation over time.



Now that we are happy with the shape of the balloon, we can reduce **Constraint Strength** for the **Weld Group** to 10.

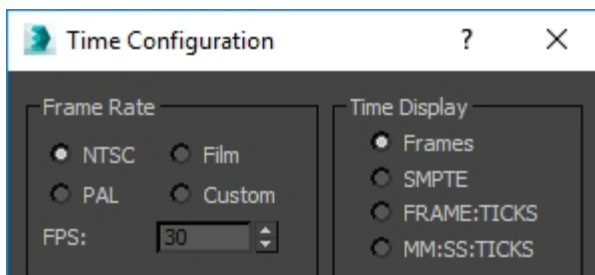
Press the **Simulate** button.

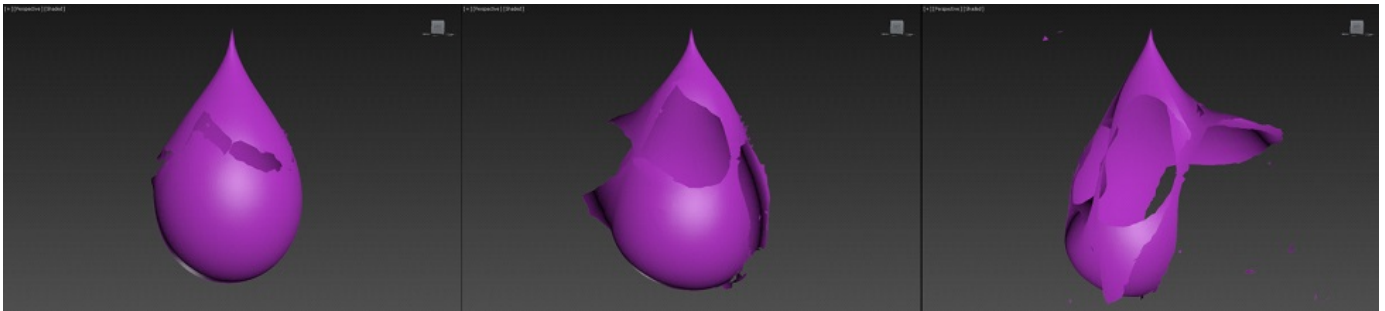
The balloon now slowly tears apart as the inflation pressure increases during the simulation.



Congratulations!

Once the simulation is complete, you can change the **Frame Rate** back to **NTSC**.

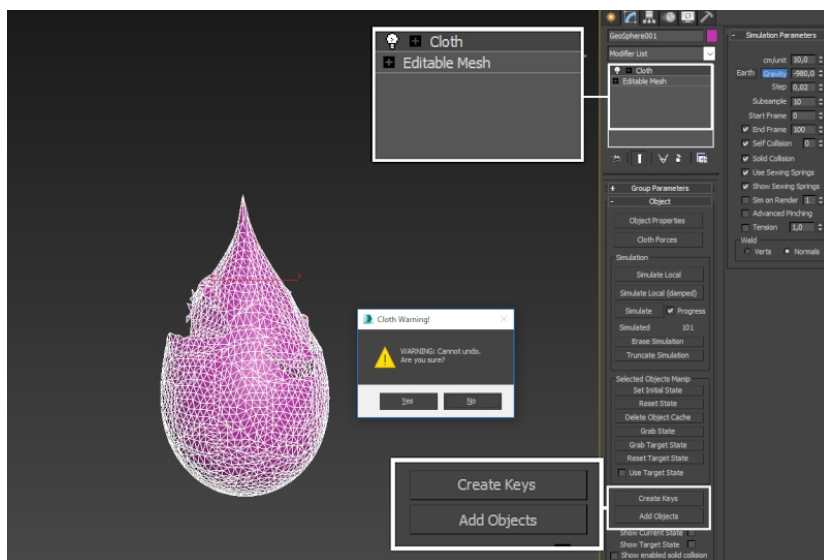




The following operation cannot be reverted!
Consider saving a backup file first!

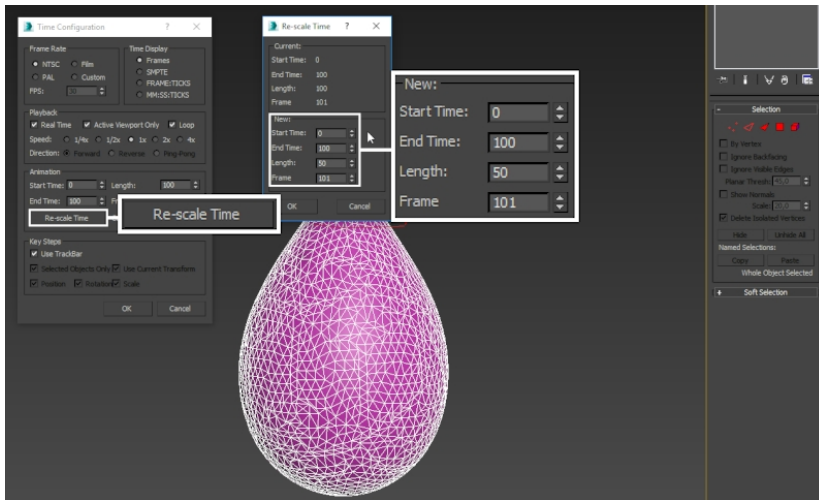
Under the **Cloth** modifier, click the **Create Keys** button.

This creates vertex animation keys and essentially bakes the cloth simulation.



From the **Time Configuration** window, click the **Re-scale Time** button.

Set the **Length** parameter to 50 to speed up the animation playback.

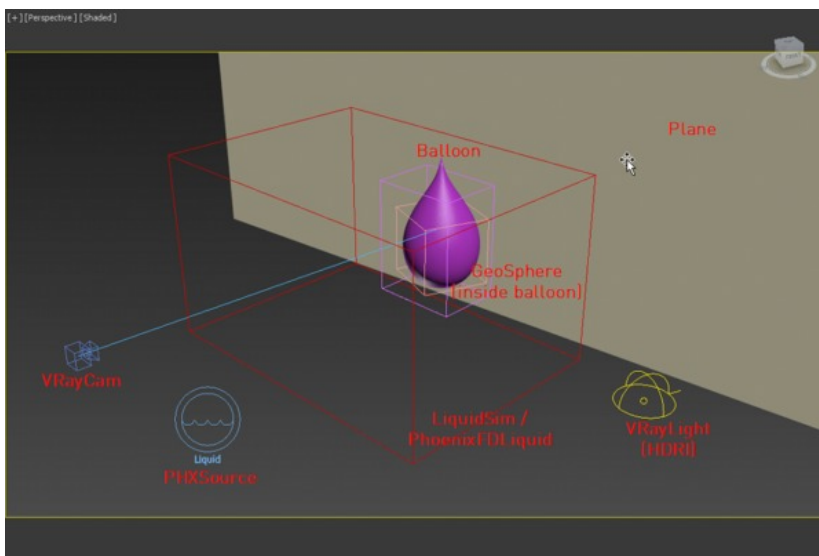


Phoenix Liquid Simulation

Again, make sure the system units are set up correctly (1 Unit = 1 cm).

Scene Setup

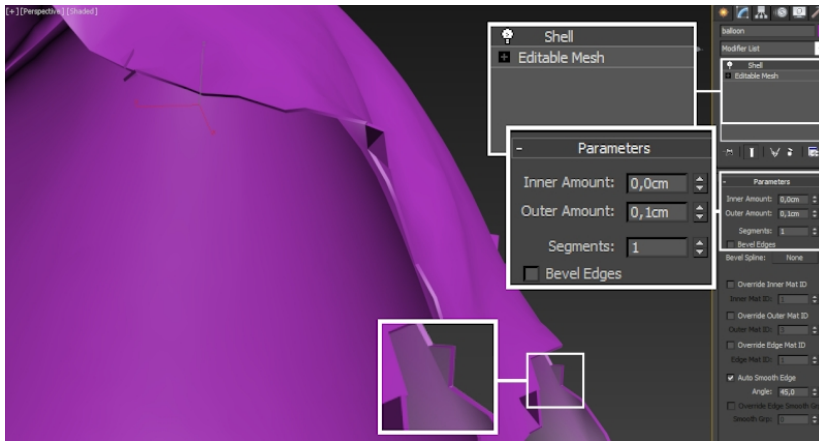
The following image shows the overall configuration of the scene.



Since the simulated balloon geometry has no thickness, we can't use it as-is in the Phoenix simulation.

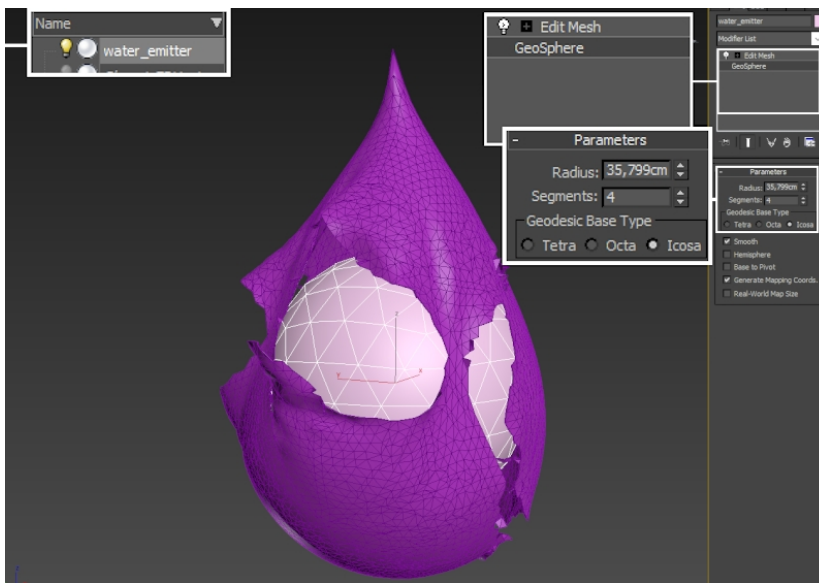
Apply a **Shell** modifier to the balloon and set **Outer Amount** to 0.1 cm.

This value can be modified depending on the size of the balloon object and the resolution of the Phoenix simulation.



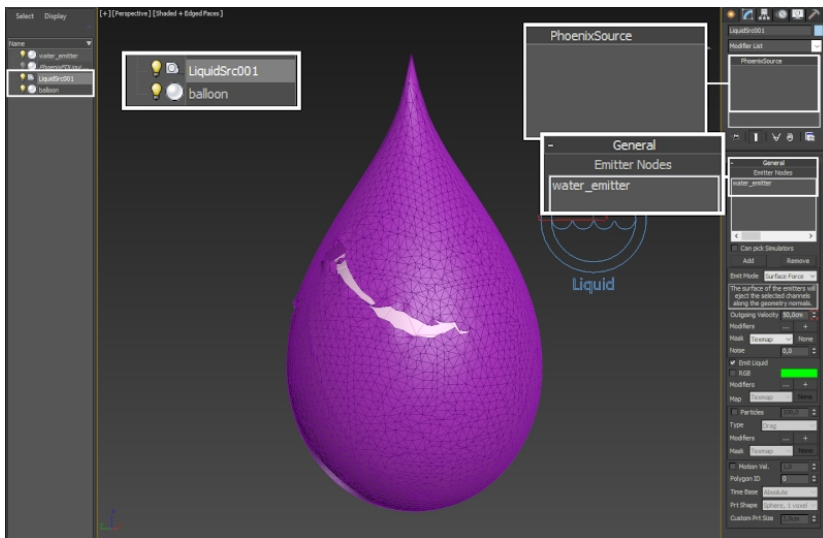
As the water in the balloon is supposed to be pressurized, place an emitter inside and use the **Outgoing Velocity** parameter to mimic the behavior of such a fluid.

Place a **Geosphere** inside the balloon.



Create a **LiquidSrc** helper from the Helpers tab of the Command Panel of **Phoenix**.

Add the Geosphere as an **emitter** with the **Add** button.

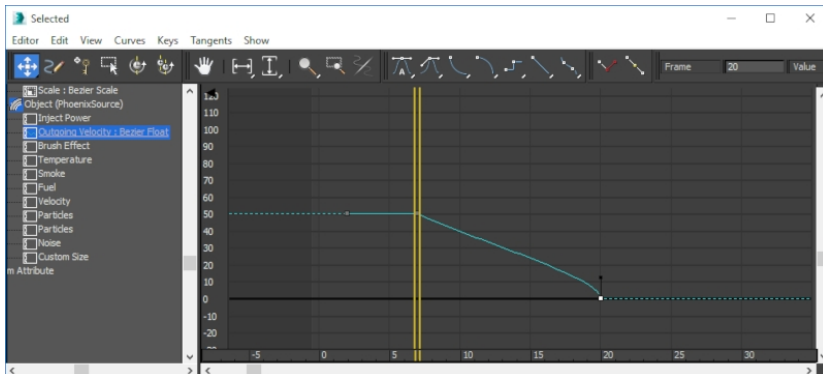


Keyframe the **Outgoing Velocity** parameter of the **LiquidSrc** helper such that it runs fast in the beginning and then suddenly stops.

The entire point of this simulation is to mimic the effect of water bursting out of a balloon. Find the best value for the LiquidSrc Outgoing Velocity and the overall timing.

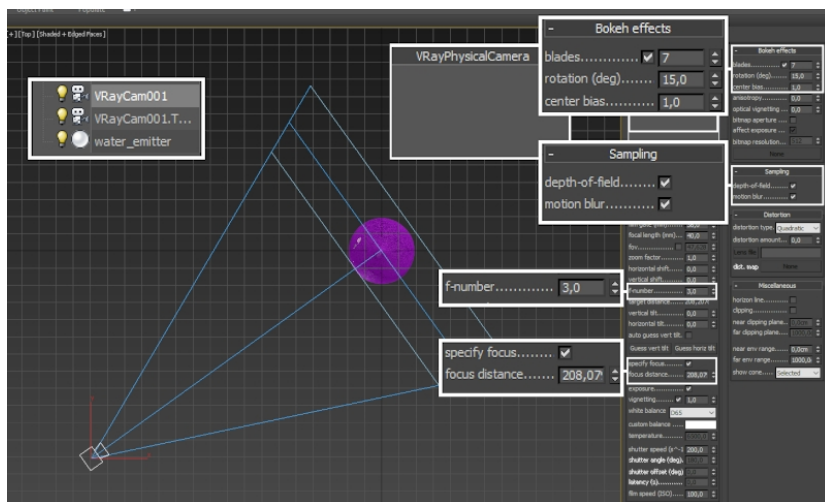
Find a good reference image or video and pay attention to the water bursting out of the balloon.

Bonus tip: **turn off the LiquidSource's Outgoing Velocity before the balloon is fully broken.**



Create a **V-Ray Physical Camera** and set the settings based on your needs.

Enable **Depth of Field** and **Motion Blur**.
Set the F-Number, Shutter Speed and ISO.



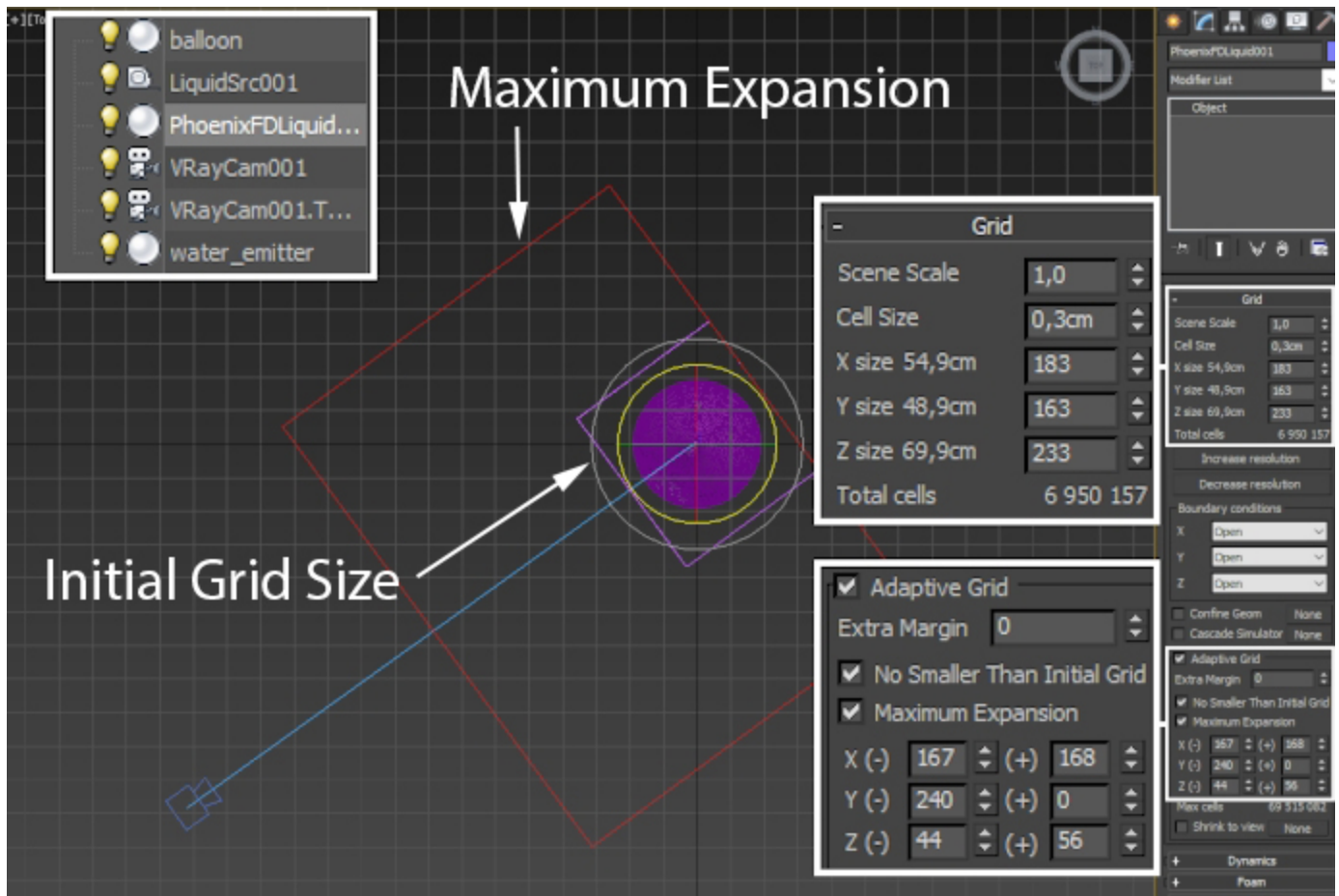
Simulator Properties

Create a **Liquid Simulator** from the **Phoenix Toolbar** or the **Phoenix FD** menu of the **Create** panel.

Adjust its size so that it encompasses the balloon geometry and try to align it to the camera.

The **Cell size** parameter does not have to be excessively small during the RnD stage - you can set it to 0.3 cm. Enable **Adaptive Grid** and configure the range of **Maximum Expansion** for X, Y and Z as shown in the picture.

Notice that the back of the balloon is partially outside of the simulation grid. This way we can save some resources and reduce the required memory for the simulation. The back side is pretty much irrelevant considering the camera angle.



Set **Gravity** to a low value (0.05) so the liquid is almost weightless.

Set the **Steps per Frame** parameter to 2.

Low values will give you a 'noisy' result which is ideal for explosions. High values (10 or more) tend to smooth out the movement of the liquid. A value of 2 gives a good result, so that's what is used in this tutorial. You can increase it if needed.

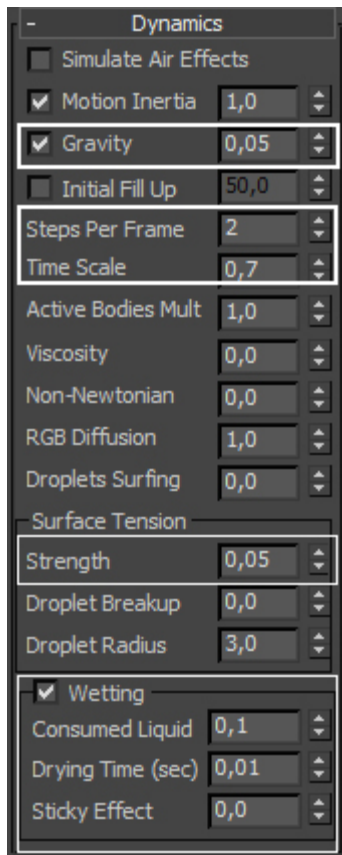
The **Time Scale** is set to 0.7 to slow things down a bit.

Set **Surface Tension** to 0.05 (as a comparison, the surface tension of water is 0.05 ~0.10).

Wetting is **enabled**, with the **Consumed Liquid** option set to **0.1**, and the **Drying Time** to **0.01** seconds.

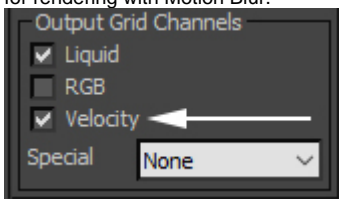
Wetmap particles are born on the surface of objects which come into contact with the liquid. A Wetmap particle group is created which can be used at render time to drive the options of a material using a Phoenix Particle Texture.

Wetmap particles also allow the **liquid** to **stick** to the **surface of objects** when Surface Tension and/or Viscosity is above 0.

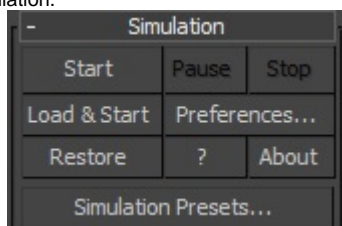


From the **Output** rollout, enable **Velocity** under **Output Grid Channels**.

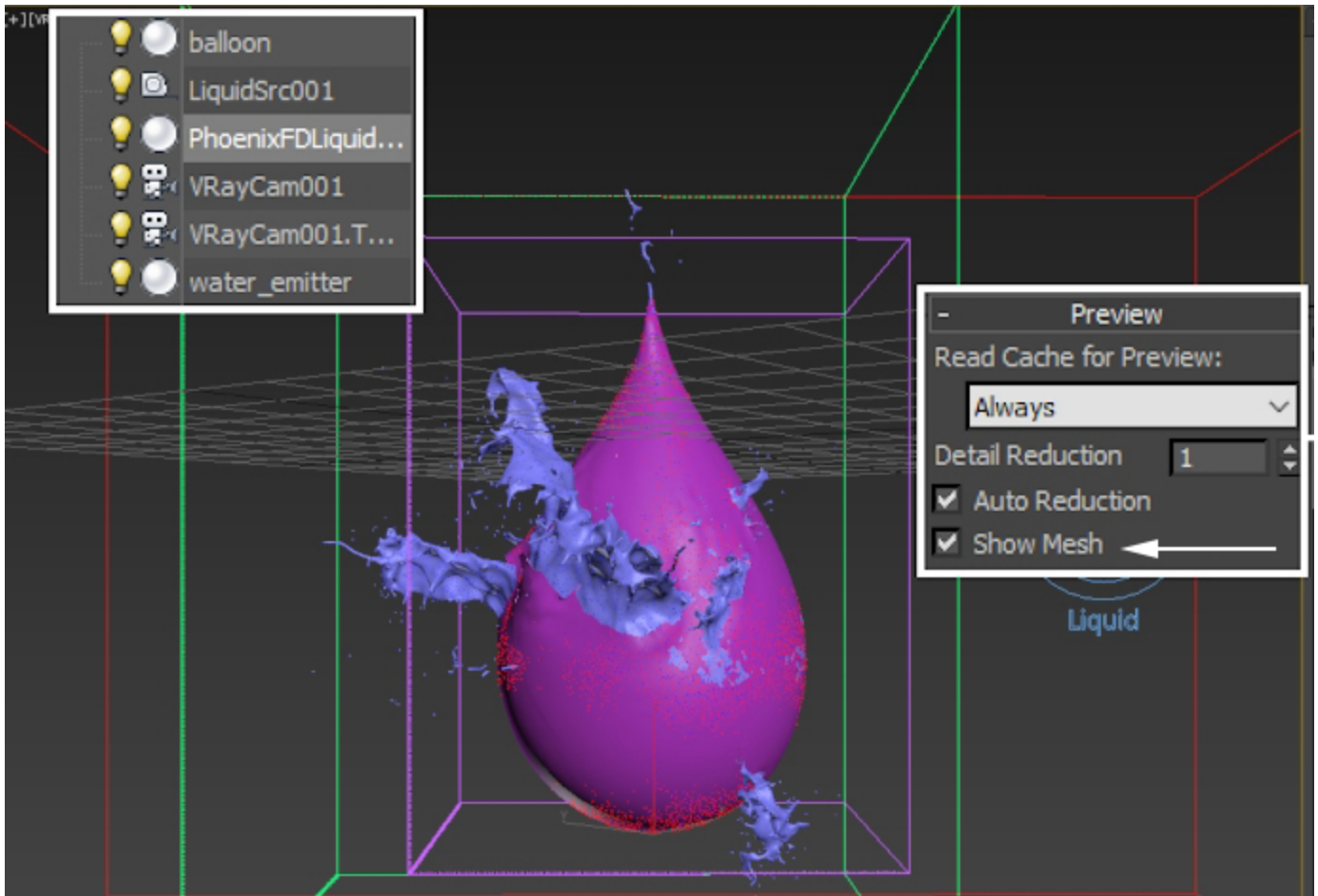
Velocity output is required for rendering with Motion Blur.



Press **Start** to run the simulation.



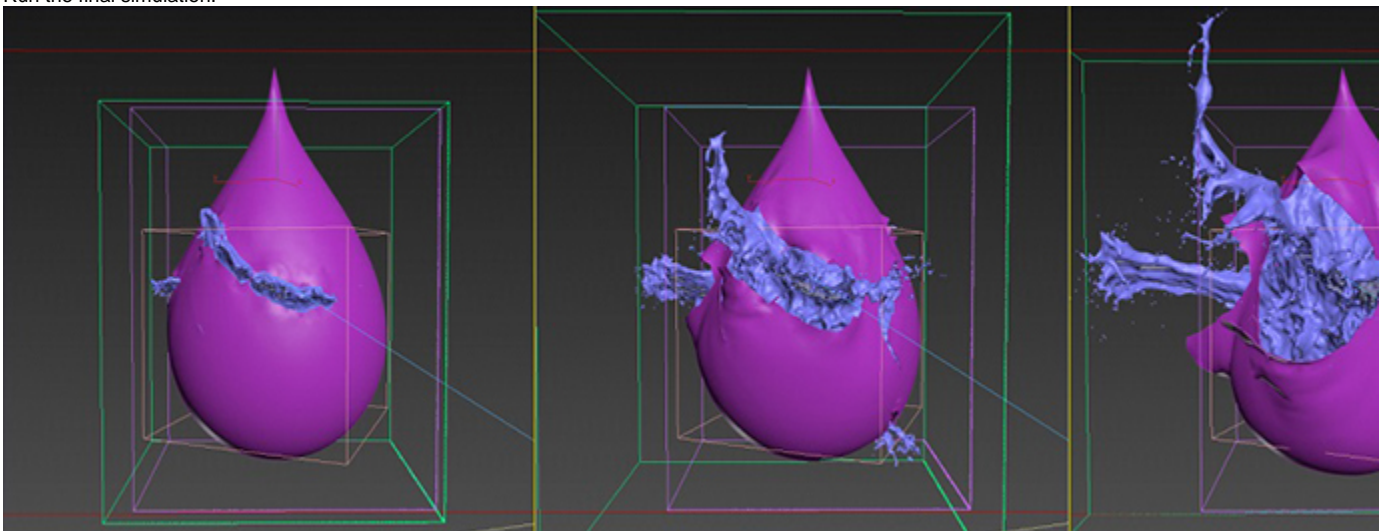
Under **Preview**, enable **Show Mesh**. This allows you to preview the liquid mesh during simulation.



Once happy with the result, increase the grid resolution by reducing the **Cell Size**.

Set the **Cell size** to around **0.2 cm**.

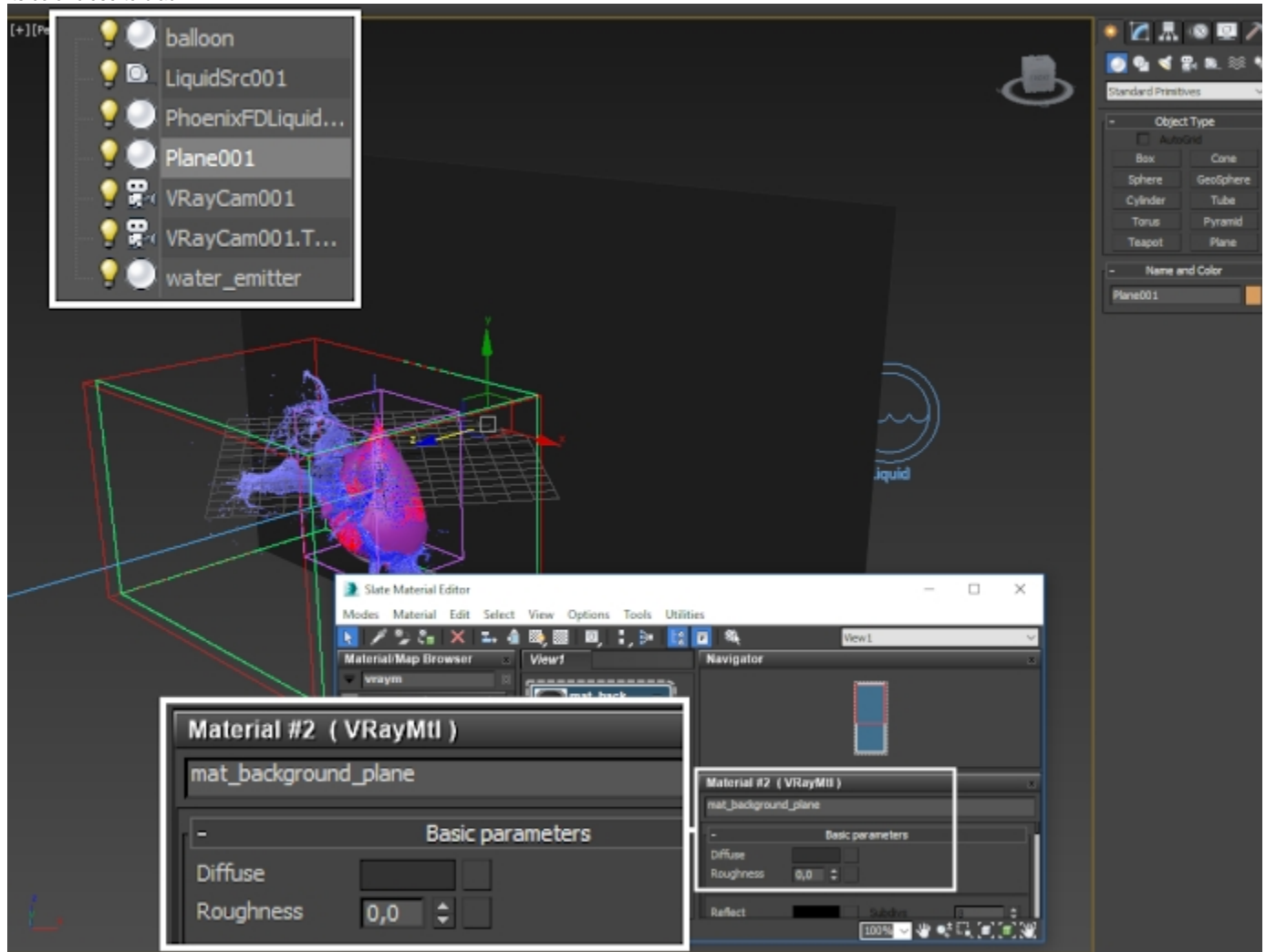
Run the final simulation.



Materials and Lighting

The lighting in the tutorial scene includes a **V-Ray Dome** light with HDRI assigned.

Optionally, you can place a plane in the background and set its color close to black.



Balloon Material

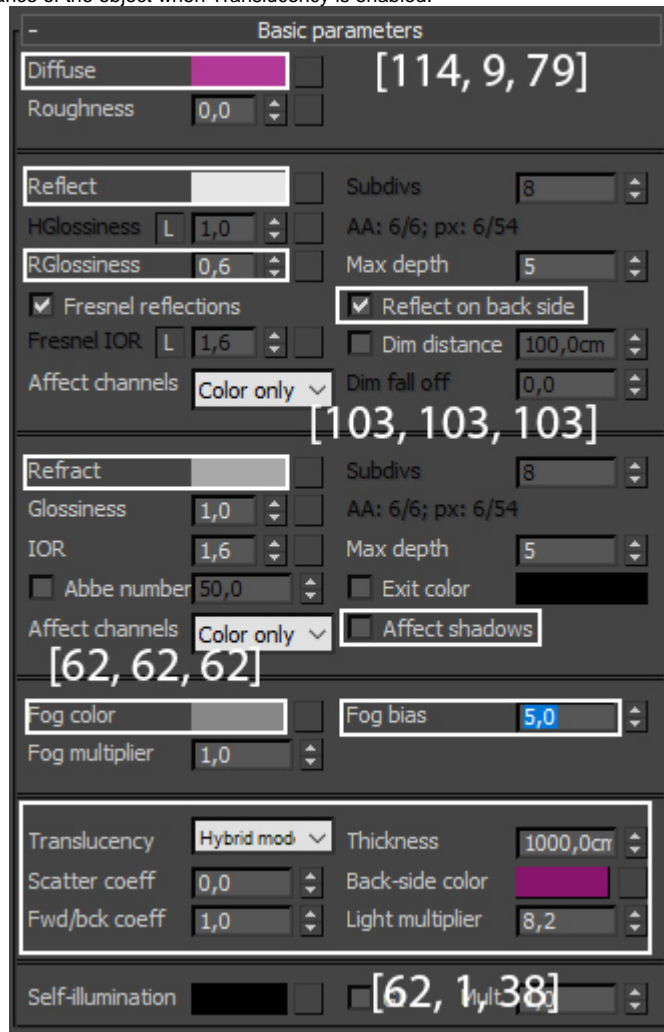
As balloons are usually made of thin rubber and thin rubber tends to be translucent, enable **Translucency**.

Set the type to **Hybrid model** and the **Back-side color** to a value slightly darker than the **Diffuse** color.
Adjust the **Light multiplier** until you're happy with the result.

Reflection is set to **White** as rubber is a dielectric and those usually reflect in grayscale as opposed to metals, which have colored reflections.

Refraction is set to a mid-gray value, required for the Translucency effect.

Set **Fog Color** accordingly and try different values of the **Fog Bias**, until the result suits you. Both of those affect the appearance of the object when Translucency is enabled.

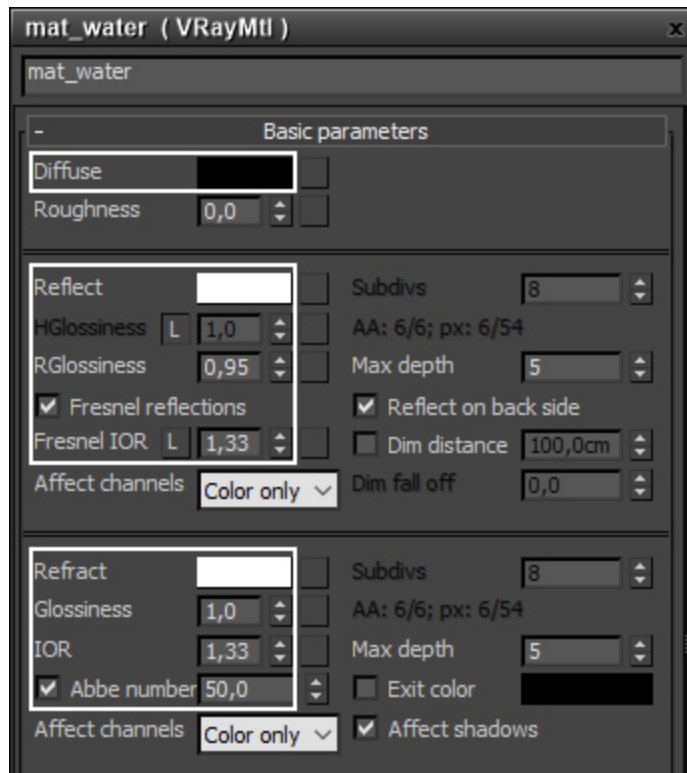


Water Material

The **Index of Refraction** (IOR) of real water is approximately 1.33. Set it accordingly.

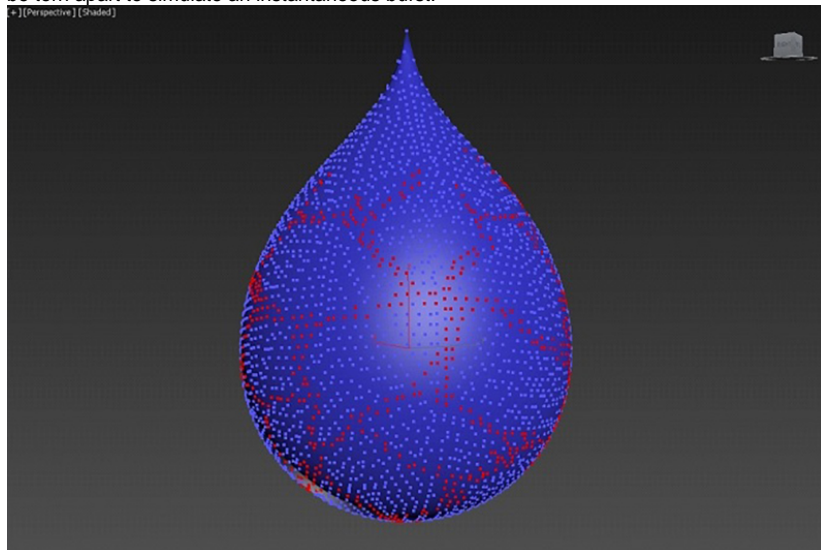
The **Reflection** and **Refraction** colors are set to **white**, which is physically correct.

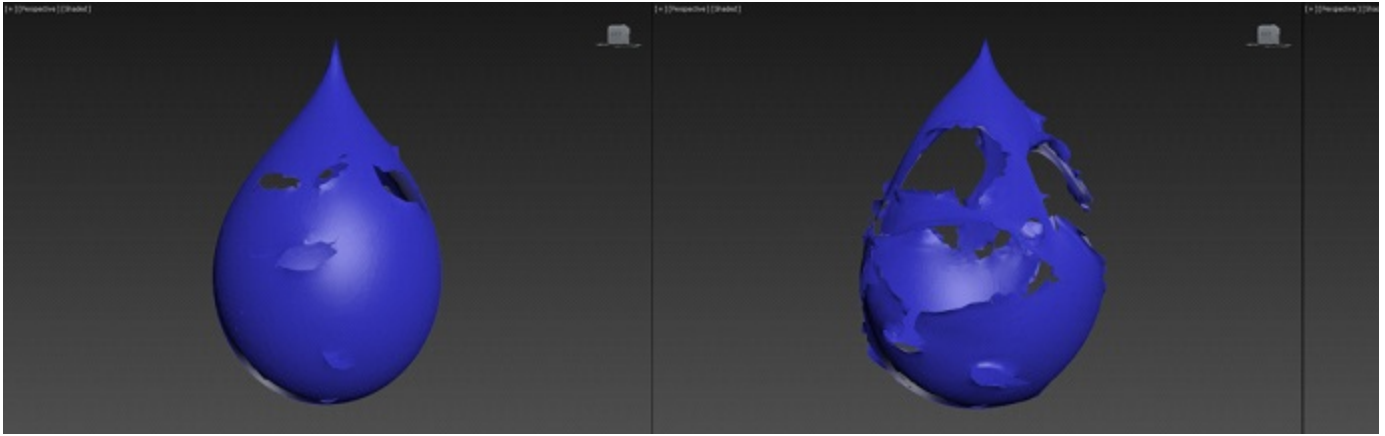
If you want to create a material for a different type of liquid, check the [Wine Glass Explosion](#) tutorial (and more specifically - the Material Setup section), where the Fog color is used to create a translucent liquid.



Extra: Instantaneous Burst Simulation

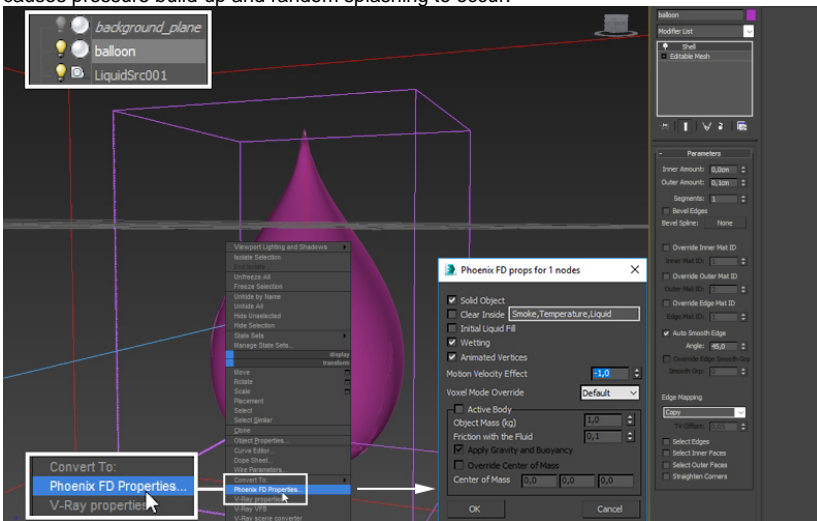
During the **Cloth** preparation stage, select more vertices to be torn apart to simulate an instantaneous burst.

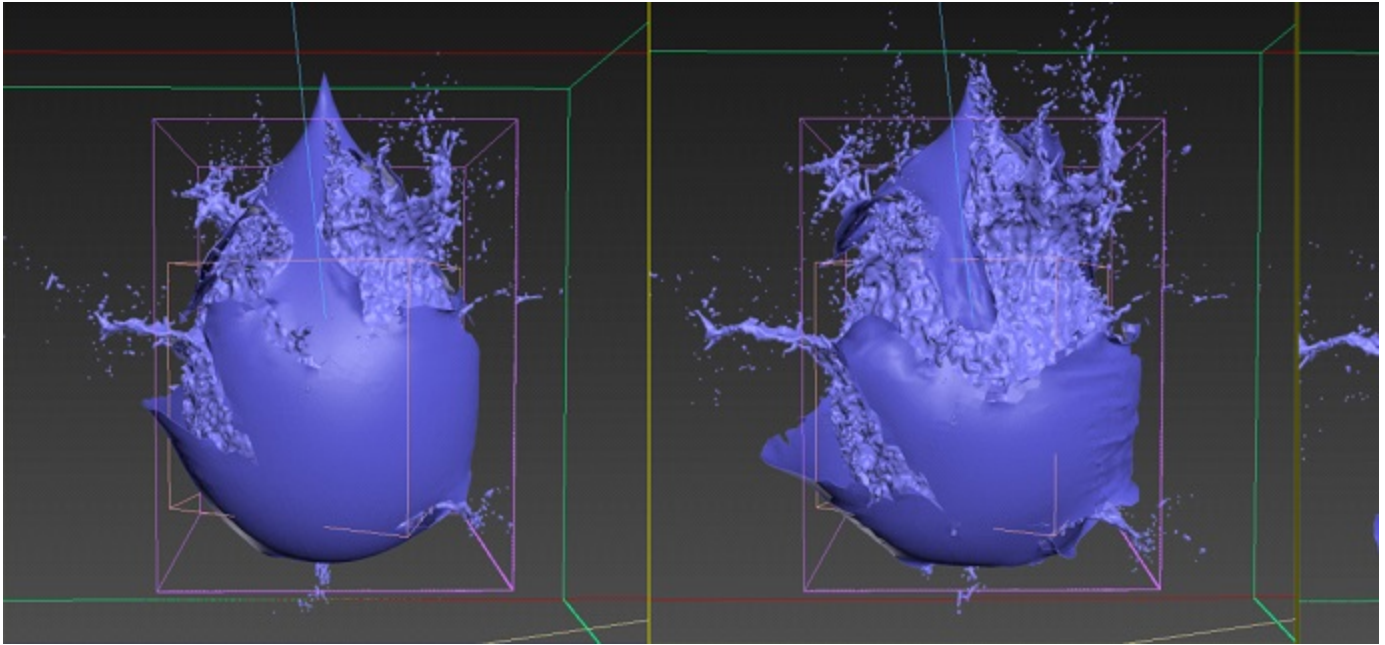




On the **Phoenix Properties** of the balloon geometry, set the **Motion Velocity Effect** parameter to -1.0.

When the rubber pieces move away from the fluid, the velocities sourced into the simulation point towards the center of the liquid, as opposed to the previous setup, where the balloon geometry is dragging the liquid outward. This causes pressure build-up and random splashing to occur.





Troubleshooting

In case there are liquid particles penetrating through the balloon geometry, the easiest solution is to increase the thickness of the balloon geometry.

Increase the inner amount on the **Shell** modifier to 1.0 cm. Once the simulation is complete, you can set the **Inner Amount** back to 0 and the **Outer Amount** to 0.1 cm for final rendering.

