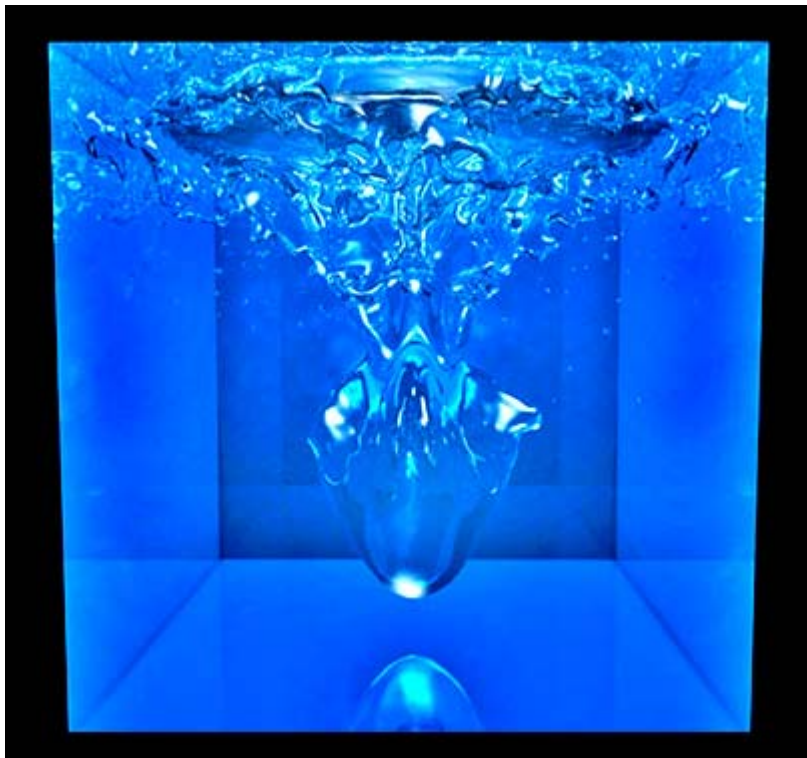


How to make a liquid levitate and dance using RealFlow and Cinema4D

Hello everybody, my name is Christian Zuppinger.

Computer graphics is my biggest hobby for some years now and I am mainly interested in the visualisation of nature, landscapes but also natural phenomena such as realistic clouds, water and fire.

The following tutorial shows the power of attractors in RealFlow4 and how they can be animated to give rise to interesting shapes and surprising animations.



RealFlow is a software from [NextLimit](http://www.nextlimit.com) which allows to simulate all sorts of liquids and physically correct dynamics of objects. The program is available for several operating system and works together with major computer graphics applications via plugins.

The purpose of this short tutorial is the creation of an animation of "magic" liquids floating in the air, dancing around and forming all sorts of strange shapes. Once you have installed the simulation and a nice rendering setup in Cinema4D, you can animate the liquids in different ways and re-run the simulation again and again until you like the "liquid-choreography".

Have a look at a finished animation, three examples of many variations possible are shown here (QuickTime movie, Sorenson codec, 6-12 Mb):

[movie 1](#) [movie 2](#) [movie 3](#)

The Workflow

This project was done using RealFlow 4.2.3. and Cinema4D R10.1 with the advanced render module on a Macintosh OSX10.4.9 system. Both RealFlow and Cinema4D are available for Mac and Windows and the files can be exchanged between the two operation systems.

This tutorial is thought for beginning users of RealFlow and Cinema4D. You need to have installed the software and the plugins.

The time needed for computer calculations in this tutorial depends on the amount of frames and on several parameters (simulation stepsize and resolution/particle count being the most important ones), but it will be in the range of some hours.

The amount of storage space needed is also very much depending on the the mesh resolution and on which types of cache files you decide to export during simulation... but it is always advisable to reserve several gigabytes for RealFlow projects.

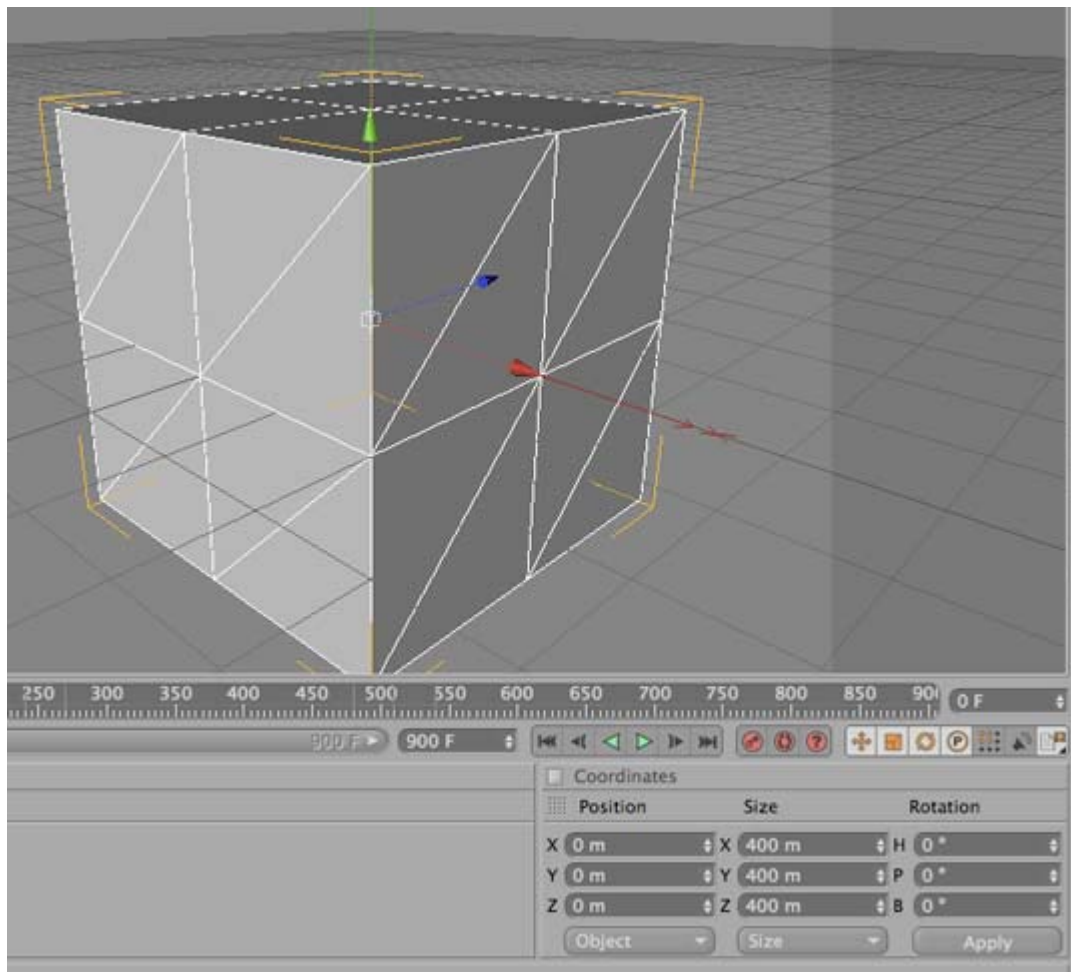
In the end, a little movie will be made, and this can be done directly in Cinema4D or with a program that assembles single images into a movie.

This is the detailed workflow of the project:

- 1) Creation of an object (a simple box) in Cinema4D and export as SD-file by the plugin from Nextlimit.
- 2) Setting up a project in RealFlow with all the necessary components such as objects, emitter, forces (daemons) and the liquid mesh.
- 3) Setup of keyframes and animation curves in RealFlow
- 4) Running the simulation, which produces the liquid (meshes) that we can import later into Cinema4D for rendering the movie.
- 5) Finishing the render setup in Cinema4D including lights and materials, importing the meshes.
- 6) Rendering the animation in Cinema4D

Preparations

Let's start with making the box (Objects>Primitive>Cube) in Cinema4D.

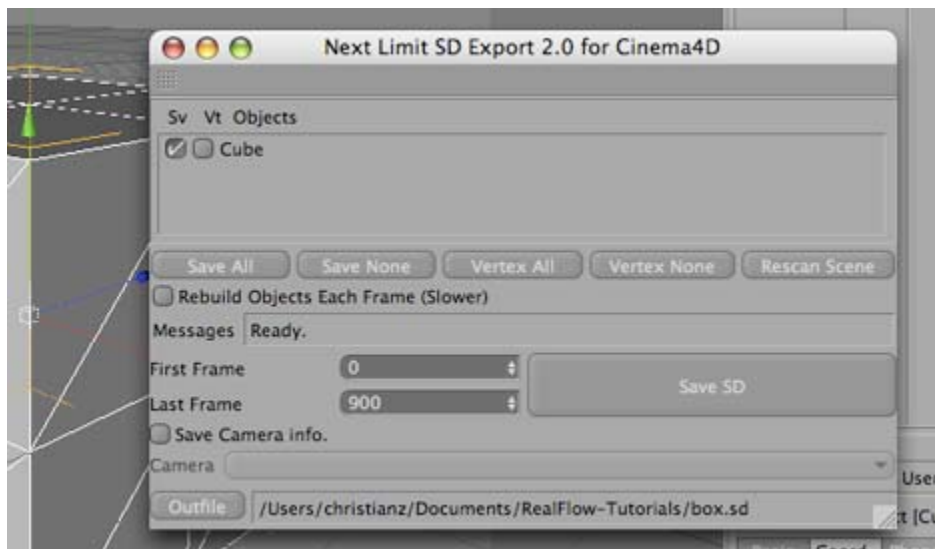


This is the simple box where the entire simulation is happening. The size of the cube in my example is 400m x 400m x 400m. Later we will use the scene scale setting in RealFlow (usually at 0.01 for Cinema4D objects) to fit the "water" exactly inside the box.

Now it is needed to make the box editable (turn into polygons), and to use "triangulate" in the functions-menu, since RealFlow expects triangular polygons in all objects.

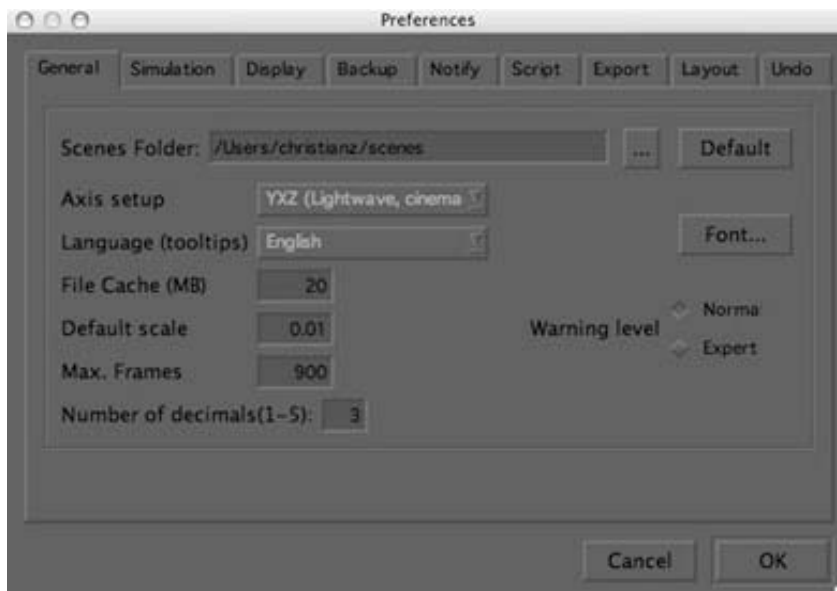
I have set the length of the animation in the timeline to 900 frames. That is already everything we have to prepare for this specific simulation. In principle, it would be possible to animate the object at this stage (while recording keyframes) and the saved animation would be transferred to RealFlow as well.

Next we have to export the object with the SD-export command from the NextLimit plugin menu (Plugins>NextLimit>[NL] RealFlow SD Export). I have set here 900 frames as the duration of the animation, but you can simulate more (or less) frames in RealFlow and simply change the number of frames when you come back to Cinema4D for rendering.



In this dialog, we need to choose first "Save All", then set the file path with the "Outfile" button and finally press "Save SD". The "Vertex All" option is only needed for deforming (soft) objects. Then close the dialog box after the SD-export is done. We will need the C4D-scene again afterwards, so save it and keep it unchanged for the moment.

Then we start up Realflow and create a new project (the project name does not matter). RealFlow now creates a folder structure on the harddisc for the different types of files that can be exported from simulations (cached particles, meshes, objects, animations, images). The SD-file from Cinema4D could now be moved into the "objects"-folder to keep the project files together. When we are about to import an object, we have to check first, if the scale factor and the axis convention is set correctly for our 3D-application, in this case Cinema4D, in the preferences dialog of the program. The Default scale should be at 0.01 and the Axis setup at YXZ:



Now let's import the cube using the SD-file you have just created in cinema4D. This command is in the File menu (File>Import>Import object) or in the "Edit" (Edit>Add>Objects>Import...) menu. The cube should now appear (usually as a wireframe) in the viewports. The cube has already basic settings for the interaction (i.e. collision) with fluid particles, it is not needed to turn on rigid body dynamics for the cube in this project.

The liquid emitter

The next thing is the setup of the particle/liquid emitter. Let's add one (Edit>Add>Emitters>Square). A rectangle appears. The position, rotation and scale of the emitter can be changed by moving one of the axis in the respective mode, which is enabled with these switches on the upper left border of the main program window:



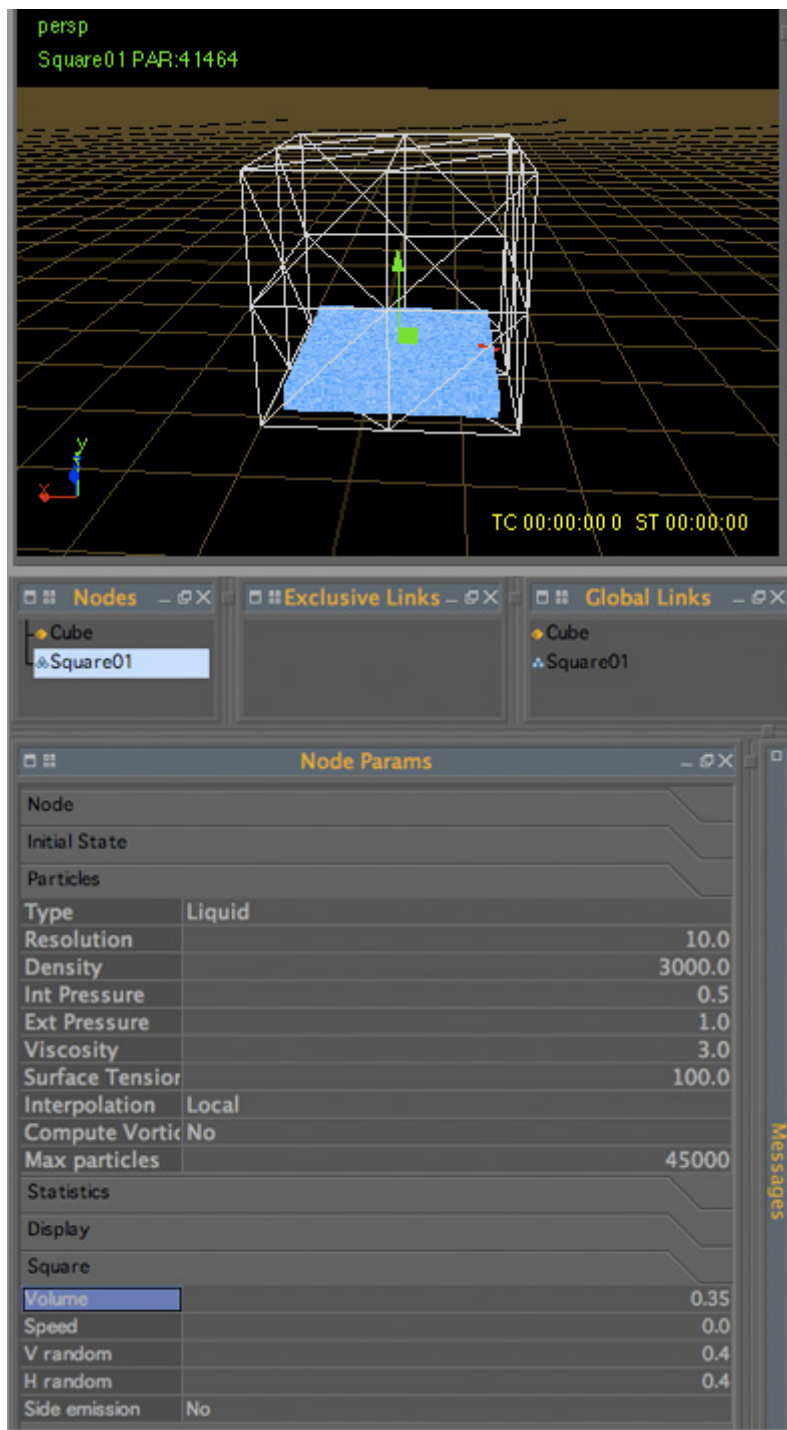
The new emitter appeared in the list of "Nodes" and in the "Global links" table. Each new element will appear in these two tables first.

The middle table (exclusive links) has the function of defining links only between specific elements. For example, a certain force should act only on selected objects in the project. Do so, the elements have to be moved inside the middle table and to be removed from the global table. This is not needed for our current project here: we will use only one object (the cube), two forces, one emitter and a mesh.

All the forces/daemons will act on the fluid and the fluid will react to the object (the cube). The daemons would also act on the cube in principle, but because we did not specify the cube as a rigid body object it won't move at all. We don't use exclusive links here, but we are going to animate the position and strength of the forces instead.

The next step is the setup of the emitter and the production of liquid particles. For our animation the square emitter (still empty at the moment) should fit inside the cube with some distance to the walls and to the floor of the cube.

The "volume" setting of an emitter is by default at 0 and that means the emitter will send out particles during simulation at a rate given in the speed setting, and until the "Max particles" are reached. However here, I have set the volume to a value >0 and RealFlow fills this volume at once with the full amount of particles (the number of particles depending on the resolution and the Max particles setting). Therefore we will start the simulation with a block of liquid already present in the scene.



There are some settings in the emitter panel which I have changed compared to the default values:

"Resolution" is set higher to provide more detail.

"Density" is higher to let the liquid react to forces in a more "heavy" and "flowing" way.

The balance of internal and external pressure has a big influence on the look of the liquids and these settings also determine the strength of interaction between particles. By applying more external than internal pressure, we can achieve a liquid which has the tendency to round up in form of droplets and to form smooth surfaces.

The "surface tension", which is set at 100 here instead of the default 0, makes the liquid ball up and form a spheroid shape.

"V random" and "H random" add randomness to the distribution of particles in the initial geometry of the liquid.

The mesh/liquid-surface

Then, the mesh/liquid surface has to be added. Select the Mesh01 in the Nodes list, use the right-mouse-button menu and the "Insert fluids" command which allows you to choose emitted particles that will be enclosed by the mesh.

The default settings for the mesh are good for a rather viscous-looking, thick liquid. A more realistic mesh for water can be generated with the filtering options and the other mesh settings.

From your 3D-software you are maybe already familiar with the concept of "metaballs", a technique to have a surface created around particles for bubbles and other organic-looking things. RealFlow uses this method, but you can achieve much better liquid surfaces than with the built-in metaball functions of Cinema4D or other 3D-packages.

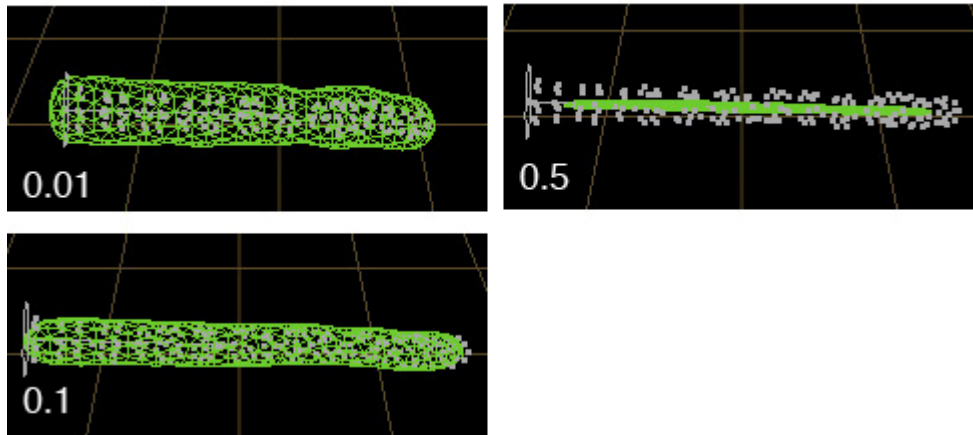


The "Polygon size" setting determines the polygon-resolution of the mesh. For this project I would use a relatively small value, because we are expecting some tiny droplets which should have a smooth surface. The finer the polygon mesh, the bigger in megabytes the mesh bin-files will be.

"Texture" we don't need for this project.

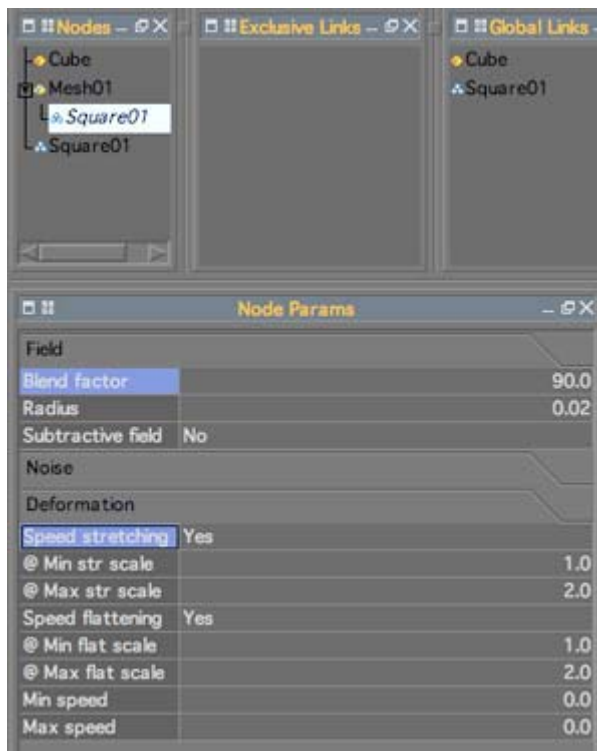
The next setting in the list is more important: I always recommend to turn on "Filter method", which takes away most of the "blobbiness" from the liquids.

Mesh-Filtering: Relaxation

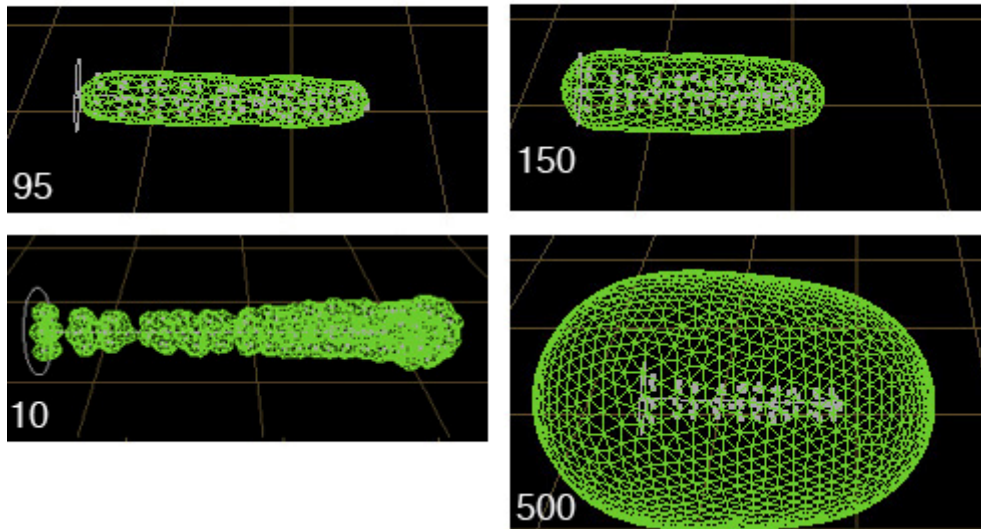


The "Relaxation" Filter seems to have a very similar effect as the mesh "blend factor" (see below), but the outcome is different. A higher relaxation value makes the mesh shrink to the core of the particle cloud so that the mesh appears thinned or even spikey. The blend factor on the other hand can be increased to produce a larger shape.

The blend factor and additional mesh-settings are shown in the node params, when we select not the mesh but the emitter *inside* the mesh:



Mesh-settings: Blend factor



"Noise" applies a certain amount of randomization on the mesh, similar to the displacement that is often used in 3D-applications in the rendering step.

Deformation settings: "Speed stretching" and "flattening" apply some distortion to the mesh depending on the direction and speed of the particle movement. The maximum stretch scale is 2.

The daemons

Now, the most important "players" in this animation have to be prepared: the forces or *daemons* as they are called in RealFlow.

Not all of the daemons are actually forces. Those with a "k" in front of the name can kill particles depending on a certain conditions and/or within boundaries.

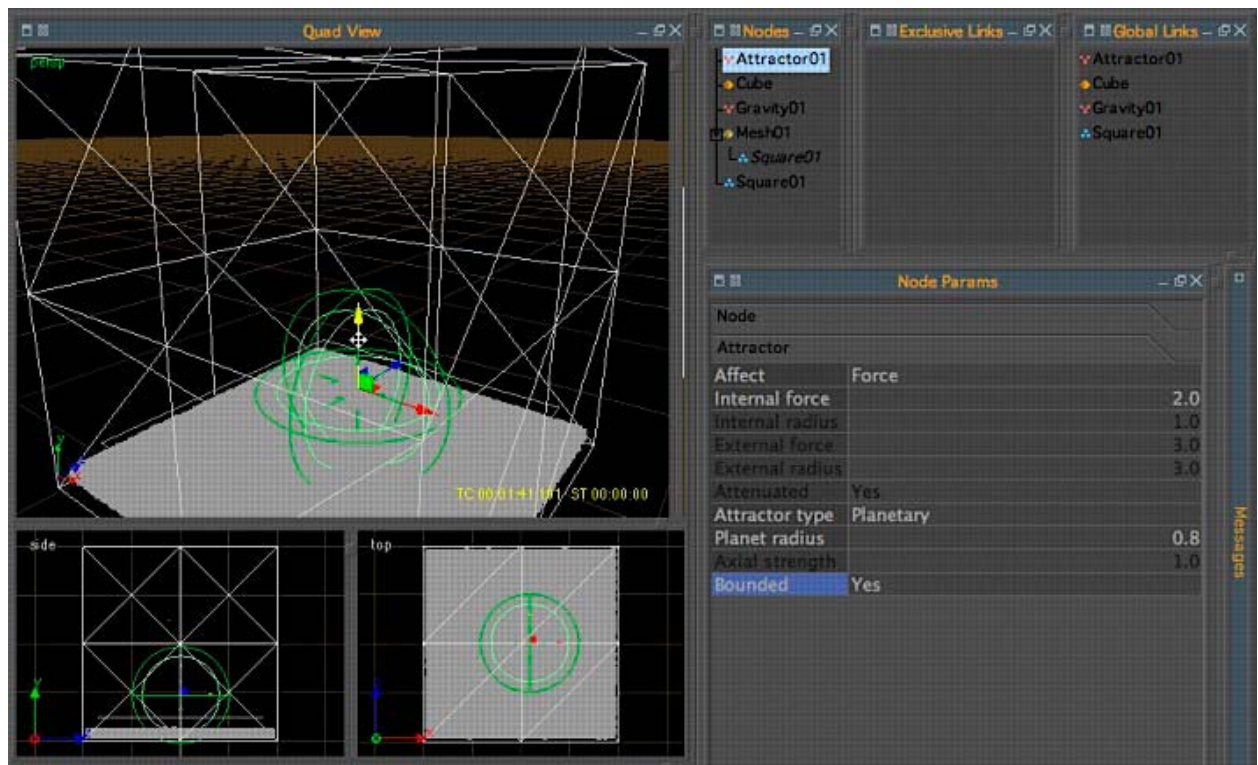
A gravity daemon is always useful to have. It provides natural gravity in the scene. But it can be used as a general unidirectional force, for example, if the liquid should rise into the air, its axis could be turned around or you could apply negative values for the strength setting. Just use the add command (Edit>Add>Daemons>Gravity) to get a gravity daemon. The strength of gravity can be animated. The position of the daemon does not matter as long as we don't use specific limits as boundary. Just leave it at default values for the moment.

The attractor daemon provides a centre of attraction in space. Without the influence of any other force, a sphere of liquid will form around an attractor. If the attraction force is strong, the liquids will gain speed when rushing to the attractor and overshoot it, until coming back and forming a sphere around it. When the attractor is moved slowly, the water "droplet" moves as well. If the attractor loses its force, the droplet may fall down under the influence of the gravity daemon. If the force of the attractor changes too quickly, "pressurized" liquid can explode. And finally, particles will be pushed away, when the force of the attractor is negative.

For this animation the spherical or the planetary type is useful.

The setting "Bounded" yes/no has a big impact on the animation as well: a not bounded attractor affects all particles, a bounded attractor just acts inside the region indicated by the green circles.

"Attenuated" means that the force loses strength depending on the distance to the centre (which is the normal behaviour of forces in nature).



Animating the daemons: keyframes and curves

There are basically two methods to set a keyframes inside RealFlow: The simple keyframes (Position Key, Rotation Key, Scale Key) and the curve editor. With the curve editor you can set all accessible parameters of a specific daemon or object, but it is also possible to set keyframes first and then modify the animation by shaping the animation curve in the respective curve-window. Also the strength of the daemon, not only its motion can be keyframed. Animated parameters will show a brownish background in the settings panel.

All these preparations happen before the liquid simulation starts, just to make that clear. When you have created a key you can right-click on the field for that parameter, select "Open curve" and you should see the curve of the selected parameter in the curve editor. On the left you see a list of possible parameters. If you would like to have the forces changing more abruptly or smoothly, just move the handles of the curve. Sometimes the curve is not completely visible. In this case you can use the "fit" commands in the pull-down menu bar. Finally, the entire animation can be tested by playing with the timeline at the bottom of the main RealFlow window (without simulation, just using the arrow buttons).

I have summarized these animation functions in this little movie taken from the screen: [animation](#)

It is actually the real fun of this type of simulations, that you animate the position and force of a daemon and the outcome is always a surprise !

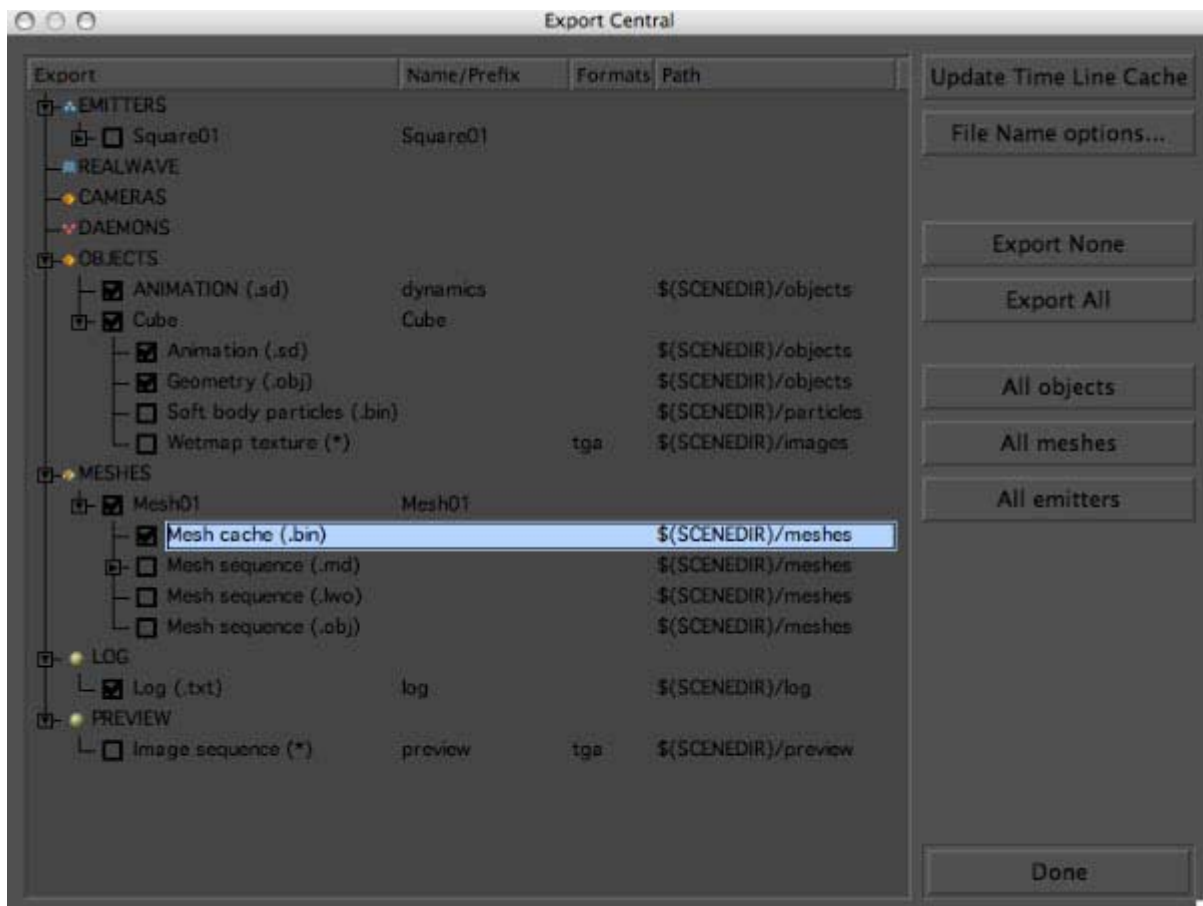
Export options

These settings are in the dialog "Export Central" in the Export menu.

Do not forget to select the export options before running the simulation to have at least the meshes (Mesh01) exported to the harddisk.

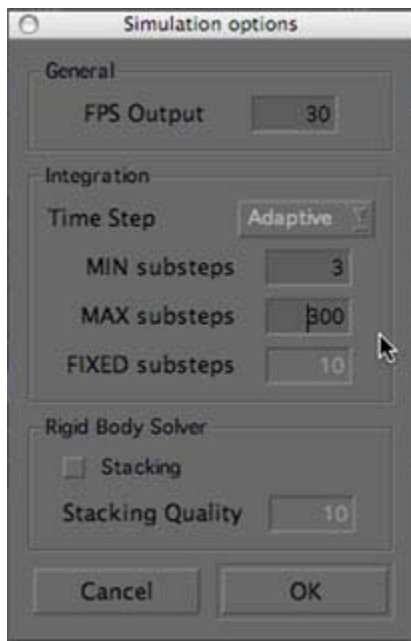
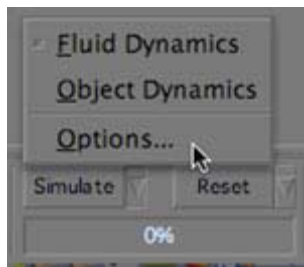
You could also store the particles themselves (EMITTERS>Square01). This would allow to reload them again and apply a new mesh with different settings and save calculating time in the simulation step. However, the particles produce rather large files on the harddisk and for a first liquid test, it is not necessary.

The export central is also the place to export object animation. The simulated dynamic motion of objects is stored in the animation.sd (the name in RF version 3 was dynamics.sd). In addition you can export the animation in the format of sd-files for single objects, and you can export the geometry of objects in the OBJ-format. This is necessary, of course, if the objects were created inside RealFlow and not imported from a 3D-application.



Let the simulation begin

You can find some simulation settings in the preferences (those are mainly thought for new scenes) or by using the fly-out menu next to the "Simulate"-button. In this project we don't animate objects but fluid dynamics.



The simulation settings have a big impact on the speed and accuracy of the simulation. The settings shown above are for high quality. If the simulation is too slow for your taste, the MAX setting can be reduced to 50, for example.

Simulations can become less stable with fewer substeps, collision detection might be less accurate and the program may eventually crash.

An alternative to reducing the substeps is to test first with fewer particles and do a simulation at higher resolution later (which will have a different result to a certain extent). Move the timeline to the start and let the simulation begin!

Rendering the animation

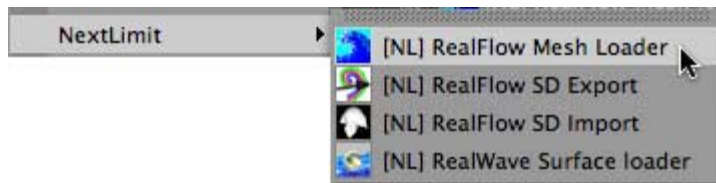
Importing the liquid into Cinema4D

Once the simulation is completed and the liquid dance looks good, it is time to render. So we can go back to Cinema4D and finish the scene setup.

The scene I used for the movie is available for Cinema4DR10 [LINK](#) (right-click).

For the front-side of the cube I just selected the polygons and deleted them, the liquid stays inside anyway... The mesh loader plugin from NextLimit will import the meshes created during the simulation. Set the file path for the mesh-bins in the ".bin File Sequence"-setting of the plugin

options by selecting the first mesh-file on the hddisc. Maybe you have to adjust the length of the animation in the timeline according to the amount of frames created and saved in RealFlow. You can always add more steps in the simulation (simply increase the "Frame End" and continue the simulation) and those new bin-files will be exported with a file name corresponding to the frame number.



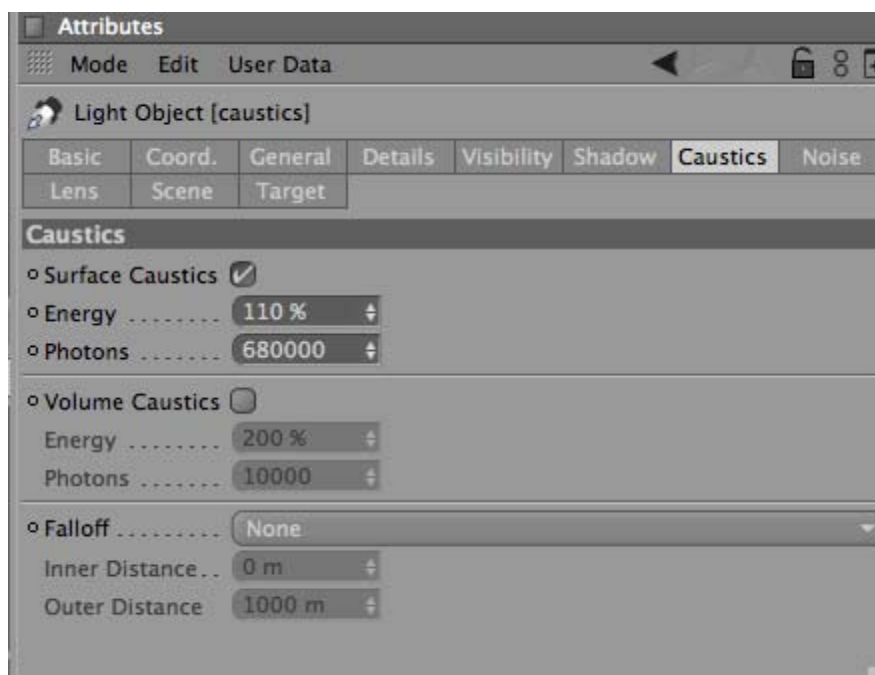
The light setup in Cinema4D

I have used two lights in this scene.

One is an omni-light called "filling" which is sitting close to the ceiling of the box.

And the other one is "caustics" which is a spot light targeted at the center of the box. This light should provide the caustic reflections caused by the surface of the liquid droplets. Actually, you could exclude the Cube in the "scene" settings of this light source, so only the caustics appear but not the reflection of the spot itself at the walls of the box.

Caustics require the advanced render module of the Cinema4D-package.

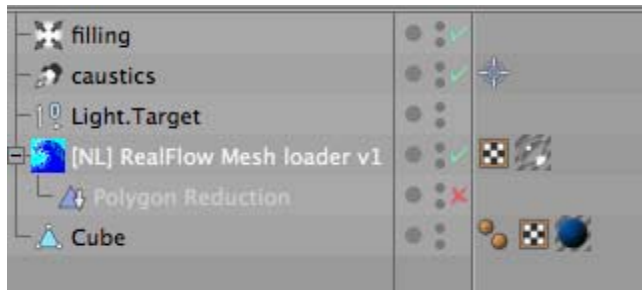


Personally, I like Caustics very much, but they increase rendertime and can create problems, if the settings of materials and lights are not balanced.

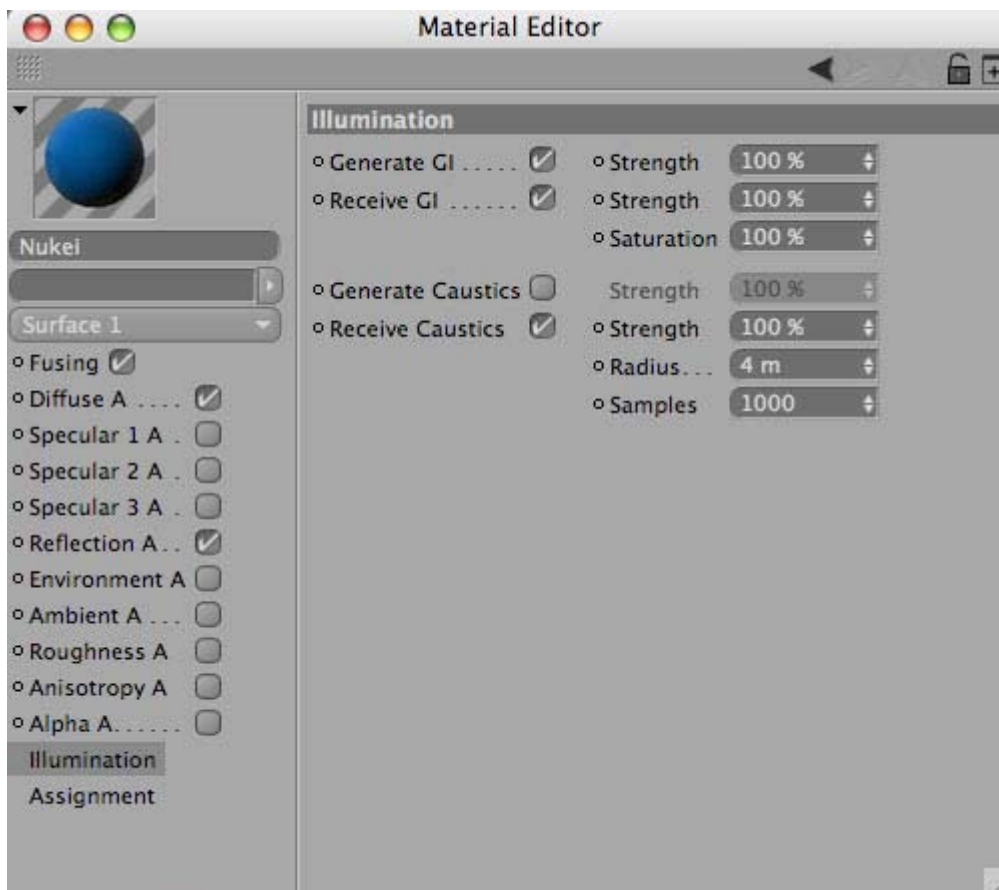
It can happen, that you see a lot of single spots instead of sparkling lights. In my experience, the most important parameter for good caustics is the number of photons from the light source and the "Radius" setting in the illumination channel of the material that should receive the caustic light. The

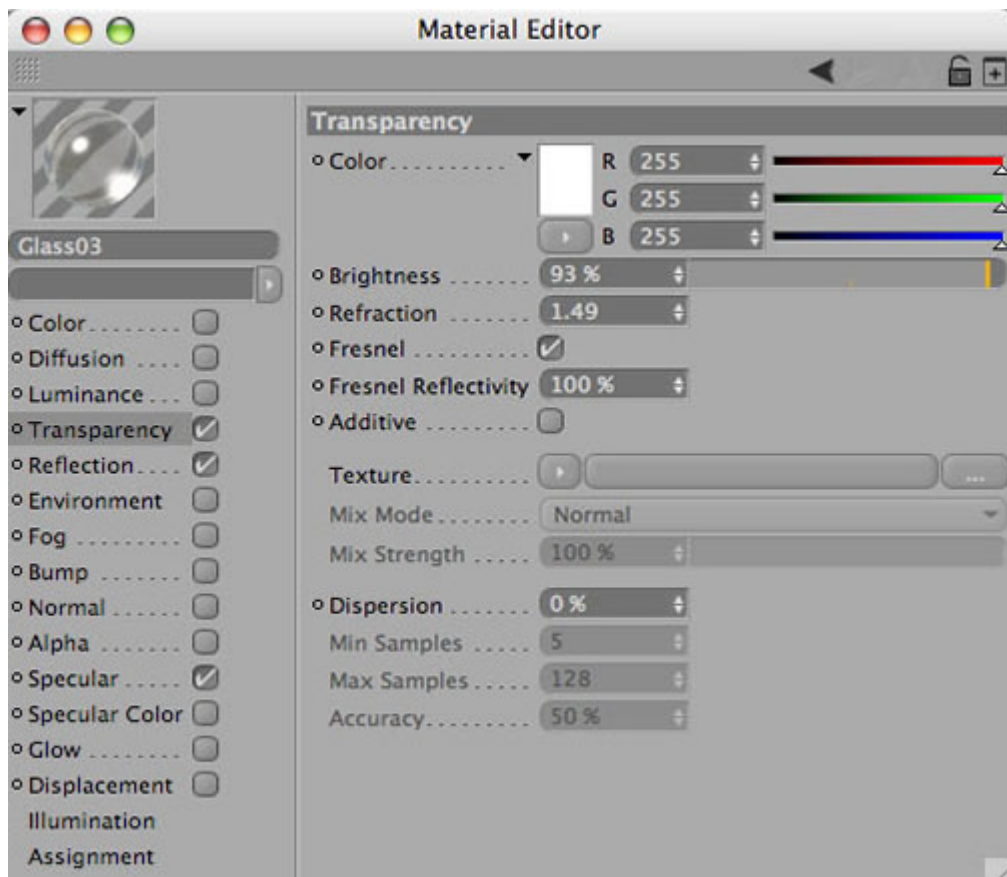
sampling parameters on the materials are less important. Of course, you have to turn on caustics in the render settings in order to see them (100% surface caustics) and they need to be recalculated freshly for every frame.

The *volume* caustics are meant for visible light. This is not needed here and should be deactivated, otherwise the rendering times explode.



I often apply a "Polygon Reduction" deformer from the objects menu to the imported mesh. This reduces rendering time but it is a pain during the playback of the meshes in the editor. Therefore, I turn it off when working on the scene (as on the picture) and just turn on before rendering. The cube has a standard nukei shader on it. There are some smaller changes, including the illumination setting to make the caustics sharper (smaller caustic radius and more samples):





This is my water material for project (above). I have tried to keep it as simple as possible in order to reduce rendering times. There are many variants of water materials possible, for example with additional wavy-roughness, some percent of fog, HDR-environment, translucent materials or ambient occlusion. It is a wide field of experimentation. In the render settings you can specify a folder where completed images will be stored, or render directly to a movie.

The scene file for C4DR10: [scene](#)

An example of many possible project setups for RealFlow4 (without exported data): [project](#)

For questions: [Email](#)

Have Fun !

Christian Z.

first version January 2006, updated for RF4 June 2007

