FLUIDOS PLUGIN FOR DAZ STUDIO

v 1.1

ALVIN BEMAR

Sumario

INTRODUCTION

FLUIDOS for Daz Studio is a plugin that runs a modified version of an external fluids simulator, the **GridFluidSim3d** (Ryan L. Guy http://rlguy.com/gridfluidsim/) with some algorithms adapted from **FLIPViscosity3D** https://github.com/rlguy/FLIPViscosity3D and C. Batty VariationalViscosity3D (https://github.com/christopherbatty/VariationalViscosity3D): These programs are implementations of PIC/FLIP liquid fluid simulation written in C++11 based on the works of Robert Bridson and Christopher Batty. The fluid simulation engine outputs the surface of the fluid as a sequence of triangle meshes stored in the Stanford .PLY file format.

Features of the GridFluidSim3d and FLIPViscosity3D simulators incorporated to *Fluidos* plugin:

- Isotropic and anisotropic particle to mesh conversion
- Diffuse particle simulation
- Autosave and load state of a simulation
- GPU accelerated tricubic interpolation and fourth-order Runge-Kutta integration using OpenCL
- GPU accelerated velocity advection using OpenCL
- Accurate viscosity for buckling, coiling, and rotating liquids

It is necessary that OpenCL for your GPU vendor is installed in your computer.

For more technical explanations about fluid simulation, see the references.

BASIC OPERATION

A fluid simulation takes place in a spatial region called Fluidos domain. The objects with a geometry inside the domain will be considered as fluids or solid obstacles. There are other objects the simulator can manage: forces, fluid sources and fluid sinks. The forces affect the fluids dynamical behavior over time. The fluid sources are inputs of a fluid inside the domain while sinks are outputs. The simulator saves the results in files, at least one for each frame.

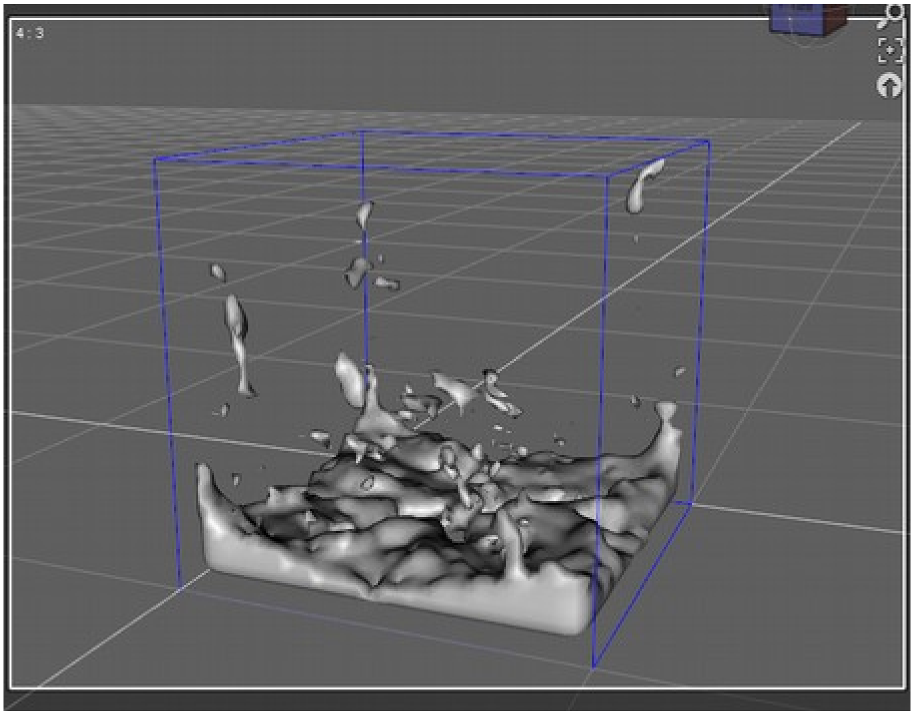
To visualize the simulated fluids, there is a Mesher, that can shows fluid surfaces or diffuse particles.

First example:

1. Prepare a folder where the simulator can save the fluid simulation files, anywhere you wish. A fast drive (as SSD) may enhance performance when rendering.
2. Create a new scene.
3. Insert a Fluidos Domain (menu Create – New Fluidos Domain and Accept).
4. Insert a sphere primitive (0.6 m diameter is a good option).
5. Parent the sphere to the Fluidos Domain.
6. Move the sphere to the center of the Fluidos Domain. Scale the sphere if you wish but avoid to fill all the Domain. Don't scale the domain for now.
7. In the Parameters tab of the sphere, as it is parented to the Fluidos domain, there are new parameters, in the FLUIDOS group. Set *ON* the *This is a Fluid mass* option.
8. Select the Fluidos Domain.
9. In Fluidos Domain Properties (Parameters tab, subgroup FLUIDOS/Main Settings), change the number of frames to 30 for a one second of simulation. Let the cell size in 2.50 (cm).
10. In the same Fluidos Domain Properties, subgroup FLUIDOS/Main Setting, click on Baked files folder and select *Browse* to locate the folder mentioned in the step 1.
11. Call the simulator (menu Edit – Run Fluid Simulation).
12. Click Accept in the dialog.
13. Wait to the simulator finish (a progress bar is displayed).
14. Set the inserted sphere to invisible.
15. Now insert a Fluidos Mesher (menu Create – New Fluidos Mesher and Accept). This must be in the same position of the Fluidos domain.
16. Choose the same Baked files folder (step 10) for the Mesher in Parameters tab, subgroup FLUIDOS Mesher/General, as in the step 10.
17. Put a key frame at frame 0 in the Timeline. Then, put a second one in frame 30 by setting *Completion* to 100 %.
18. Check *Enabled* in the parameters (subgroup FLUIDOS Mesher/General).
19. You will see a preview. This is intended to get a fast scroll around the Timeline.
20. The final simulation can be seen by setting *OFF* the Preview parameter of the Mesher (subgroup FLUIDOS Mesher/General).

You can stop the simulation by pressing escape key or click in Cancel in the progress bar. To resume, call again the simulator and set *ON* in Continue saved state before to click Accept button in the dialog.

The final results at frame 30 must be very similar to the following image:



Note:
If Daz Studio "freezes" during Fluidos running, or if the results are completely different than the above (e.g. all the fluid goes up), change the Preferred device to CPU (In Fluidos Domain Properties, Parameters tab, subgroup FLUIDOS/Advanced Settings) or to GPU 2 (if your system have two). Other option is set Off the property "Enable OpenCL", but this is some slower than CPU as preferred device. The plugin will remember these two settings for next time you create a New Fluid Domain[1].

1 The defaults are:
"Enable OpenCL" = On
"Preferred device" = GPU

RESOLUTION SCRIPTS

There are four resolution scripts included with Fluidos packages. These scripts are only for the convenience of the user; they are not required for the fluids simulation.

The scripts should be applied to the **Fluidos Domain** *after setting its size,* although could be reapplied in any moment before the running of the simulation.

The resolution scripts function is to select a suitable *cell size* and the *subdivision level* for the size of the Domain. However, after applying a script, the user can modify those properties to tune the simulation.

L Low resolution: fast, but low quality. The simulator could need until **18 MB** of RAM.

M Medium resolution. The simulator could need until **146 MB** of RAM.

H High resolution. Slow, but high quality. The simulator could need until **1.1 GB** of RAM.

VH Very high resolution. Could be very slow. The simulator could need until **9.2 GB** of RAM.

The actual memory RAM needed depends on the quantityof fluid inside the Domain.

Take into account the resources of your computer before running!

COMPONENTS OF THE SIMULATION

FLUIDOS DOMAIN:

The fluid simulator performs its computations on a 3D grid, and because of this, the simulation domain is shaped like a rectangular prism. You can modify the width, the depth and the height of the domain using the sizeX, sizeY and sizeZ properties. Or you can use the scale tool.

All the objects to be part of simulation, except by forces, must be inside the 3D space of the Fluidos Domain. Even more, all these objects, including forces, must be parented to the Fluidos Domain. There is no limit of level of depth in hierarchy inside the domain.

Multiple *Fluidos Domains* can be in a scene, but only one is calculated at a time.

OBSTACLES:

Any 3D object with a mesh can be used as a solid obstacle in the simulator. The meshes *must be closed surfaces*, to function properly.

Thin wall could leak fluids. This is fixed decreasing the cell size, increasing the *Thicken* parameter or adding a Geometry shell.

It is not necessary that the whole solid volume be inside the Fluid Domain, but the outside regions will be ignored by the simulator.

The solids could have a surface force in them. This can be used to stick the fluid to the solid or to repel it.

The engine *has no functionality to simulate the affect the motion of moving obstacles*, but you can animate the solids manually. The default behavior of the simulator is consider the obstacle as static, so if you keyframed its motion, the collision with fluid cells causes the obstacle substitute them and therefore. If there are fluids sources in scene, This can be mitigate by using surface forces to simulate a push and by increasing the FPS of the simulator. A little slower option is to *Enable the moving obstacles;* this functionality simulates the effect of the obstacle *on* the fluid.

Due to the intrinsic characteristics of the FLIP algorithms, the fluid volume could increase or decrease over time. The moving obstacles tends to enhance the loss, but enabling the moving obstacles lessen this behavior, although not entirely.

FLUID OBJECTS:

As in solids case, any 3D with a closed surface can be a fluid object. It will react as a fluid mass in the simulator.

It is not necessary that the whole fluid volume be inside the Fluid Domain, but the outside regions will be ignored by the simulator.

SOURCES AND SINKS:

The simulator can manage sources of fluid so to get a stream. There is a special FLUIDOS object for them: *Fluidos Source / Sink.* This sources can have a rectangular prism shape or a sizable spherical one (this is not visible as a sphere in the screen). They can be resized and moved around the domain. The velocity of the fluid is animatable too.

A similar object is the sink, but it absorbs the fluid. It no needs a velocity.

It is not necessary that the whole source or sink be inside the Fluid Domain, but the outside regions will be ignored by the simulator.

FLUIDOS MESHER:

The simulator saves the bakefiles in a folder you choose. But for rendering the simulation you need a Fluidos Mesher. This object reads the geometry generated by the fluid simulator. Do not scale the Mesher unless you want to scale or deform the simulation results.

If you want to run the simulator to extend or modify a previous simulation, you may want to de-enable the *Mesher* until the simulation is finished, because this can slow down the process.

Meshers can be in a scene, and all of them can be enabled.

VISCOSITY:

The simulator can manage viscous liquids too, as oils, honey or asphalt.

Sometimes, with low viscosity values, bits and pieces of fluid can get left hanging in the air. To solve this problem, increase the *Marker Particle Scale* value.

Viscosity simulation are slower than inviscid ones; be patient.

UNITS:

The units of length used in the simulator are cm unless is indicated otherwise.

FORCES:

The forces in *FLUIDOS* are nodes that can affect the fluid dynamics by attracting or repelling the fluid masses in space.

The *FLUIDOS* plugin is able to manage spatially constant forces; that is, forces than act with the same intensity in all space, as the usual gravity force in earth (thes forces are controled with the Fluid Domain Properties Body Forces options).

Other type of forces are the spatially variable forces. These act with variable intensity in space. FLUIDOS can manage three kind of spatially variable forces: Point, Torque, Linear and Flow force.

Point Force: This force acts from a point in space. The fluid is attracted to this point (strength positive values) or repelled from it (strength negative values). The intensity of the force depends on the distance of a fluid mass from the central point. The decay rate is quadratic, i.e., if the distance doubles, the intensity decreases to a quarter.

Linear Force: This force acts from a straight line in space. The fluid is attracted to this line (strength positive values) or repelled from it (strength negative values). The intensity decay rate is quadratic.

Torque Force: this is a rotational force. As in the case of point force, the position is relevant, but the rotation too. This force acts in all space exerting a tangential push over fluid, so this will rotate around the Torque force axis.

The last type of force are the surface forces. These act only *on* the surface of the fluid, not *in* its body. FLUIDOS has two surface forces:

Flow Force: this force acts randomly in points in the fluid surface not in contact with a solid obstacle. It can simulate wind, and for that, has a main direction of push.

Surface force in Obstacles: this force acts in the contact surface between a specific solid obstacle and the fluid. It's constant in all the contact surface, and its intensity is the same for all solids. Positive Intensity is for attraction and negative Intensity is for repulsion.

The forces units are in acceleration units (as if the mass is unitary), m/s^2.

All the forces are animatable.

DIFFUSE PARTICLES:

The diffuse particles, an additional option of a simulation, are equivalent a the Whitewater of other software. These represents foam, spray and bubbles. In fact, the Mesher uses triangles to represent them. But this triangles can be considered as billboards.

The diffuse particles add realism to a fluid simulation.

FILES:

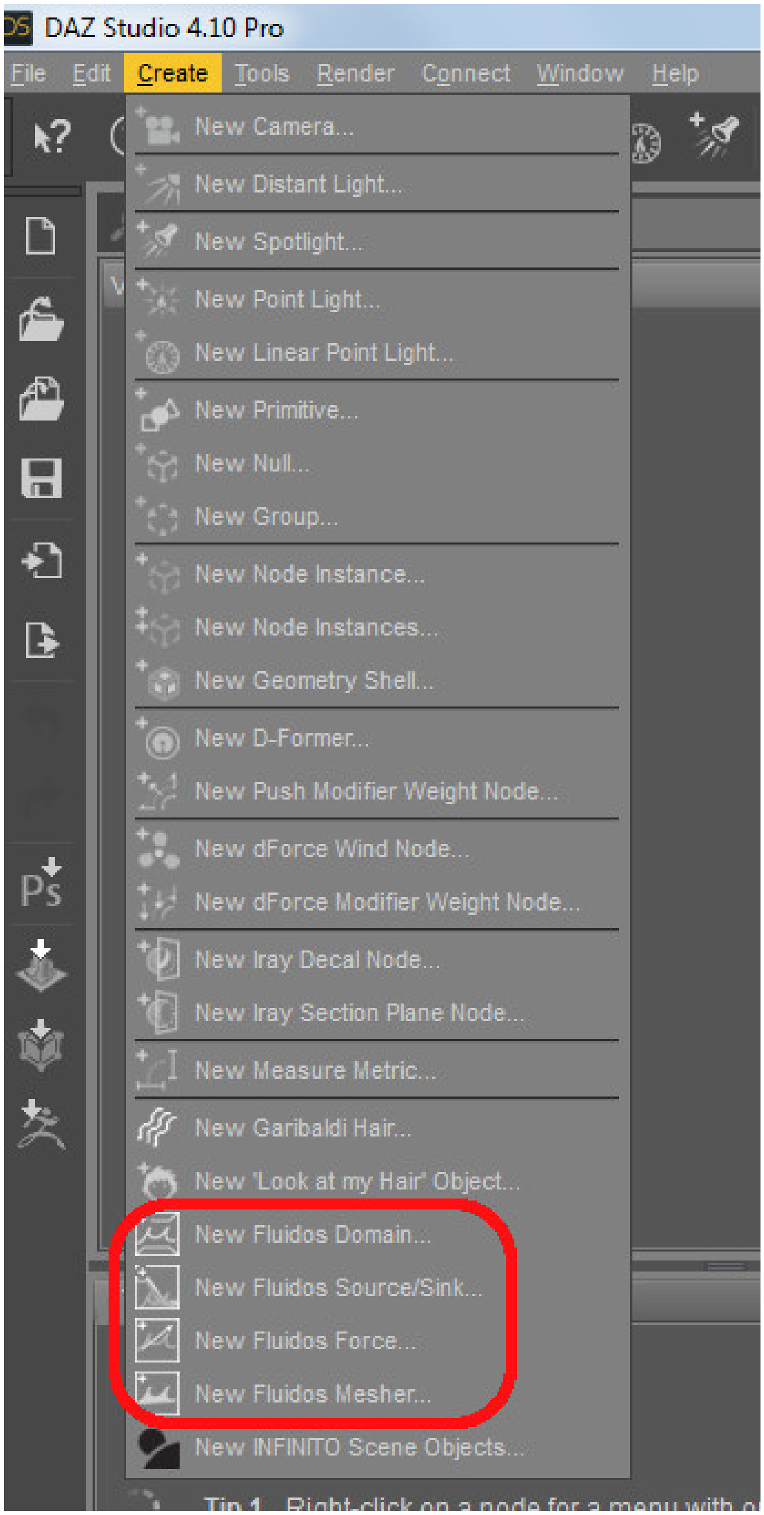
The plugin creates four subfolders inside the folder selected by the user: *bakefiles*, *logs*, *savestates* and *temp*. In *bakefiles* are saved the simulation in .ply format; *logs* contains the logs of each simulation; inside *savestates* is the file the save the last state of the simulation (it is used to resume a stopped simulation); *temp* is a reserved folder.

The geometry is not saved (except the present in screen) with the .duf, but in the folder ouput.

As the simulations overwrites their files, it is advisable to reserve a folder ouput for any relevant work.

MENUS AND PROPERTIES

To put the FLUIDOS nodes in scene, select the menu Create – New Fluidos Objects

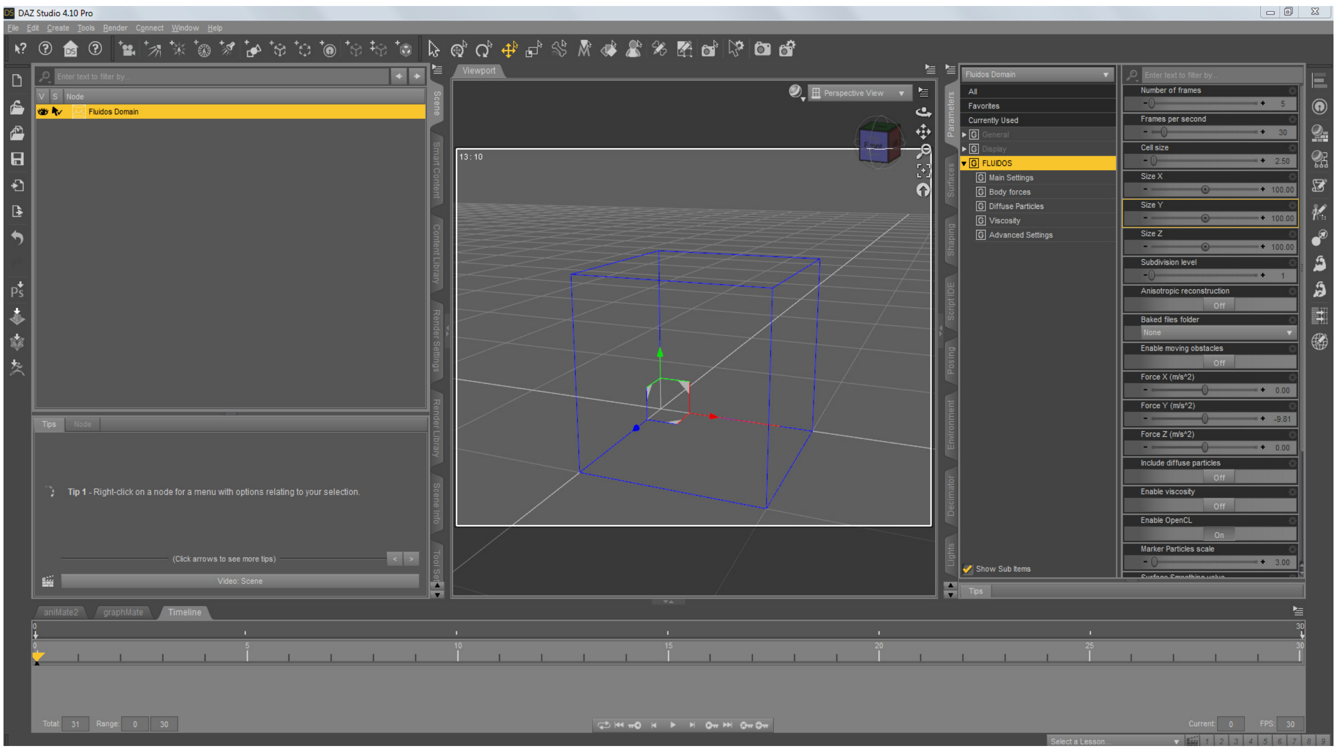


The user can now choose a new one of four nodes:

- Fluidos Domain
- Fluidos Source/Sink
- Fluidos Force
- Fluidos Mesher

Fluidos Domain

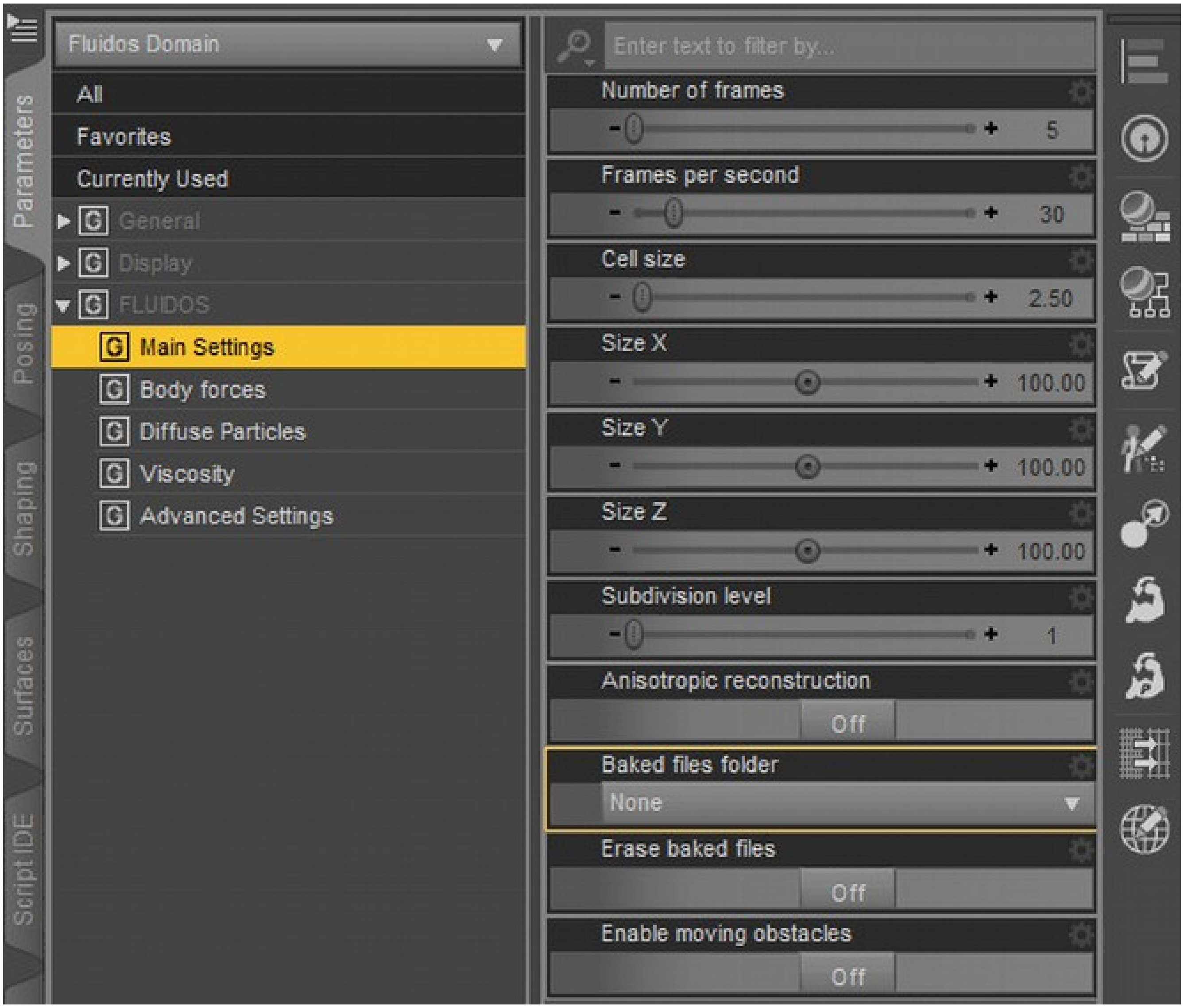
The Fluidos Domain is represented in screen as an empty cube. This cube can be resized as a rectangular prism.



In Parameters, there are five groups of them for the Fluid Domain:

- Main Settings
- Body forces
- Diffuse Particles
- Viscosity
- Advances Settings

Main settings:



Main Settings parameters are:

Number of frames: is the number of frames to simulate.

Frames per second: it is the frames per second of simulation result, not of the rendering ones.

Cell size: is the cell size of the fluid grid simulation in internal units of simulator. The smaller, the better simulation and the slower too. Is the cell size halves, the total number of cells for calculation grows cubically (8-fold increase). Beware, it can eats all physical memory.

Size X, *Size Y* and *Size Z* corresponds to the width, height and depth, as usual in Daz Studio. The units are centimeters, the default of Daz Studio.

Subdivision level: The surface subdivision level determines how many times the simulation grid is divided when converting marker particles to a triangle mesh. For example, a simulation with dimensions 256 × 128 × 80 and a subdivision level of 2 will polygonize the surface on a grid with dimensions 512 × 256 × 160. With a subdivision of level 3, the polygonization grid will have dimensions 768 × 384 × 240. A higher subdivision level will produce a higher quality surface at the cost of longer simulation times and greater memory usage.

Anisotropic:
Isotropic triangle meshes are the default form of program output. These triangle meshes represent the surface of the fluid and are constructed by the fluid simulator from a set of spheres with uniform radius. The meshes are written to the bakefiles/ subdirectory as a sequence of .PLY files in the form 000000.ply, 000001.ply, 000002.ply, ..., where the file numbers correspond to the frame number.

Anisotropic triangle meshes represent the surface of the fluid and are similar to isotropic meshes except that the meshes are constructed from a set of ellipsoids instead of a set of spheres. The benefit of constructing the surface from a set of ellipsoids rather than a set of spheres is that sharp/smooth features of the fluid surface are better preserved. This benefit comes at the cost of a longer surface mesh computation time.

The anisotropic meshes are written to the bakefiles/ subdirectory as a sequence of .PLY files prefixed with the keyword anisotropic in the form anisotropic000000.ply, anisotropic000001.ply, anisotropic000002.ply, ..., where the file numbers correspond to the frame number.

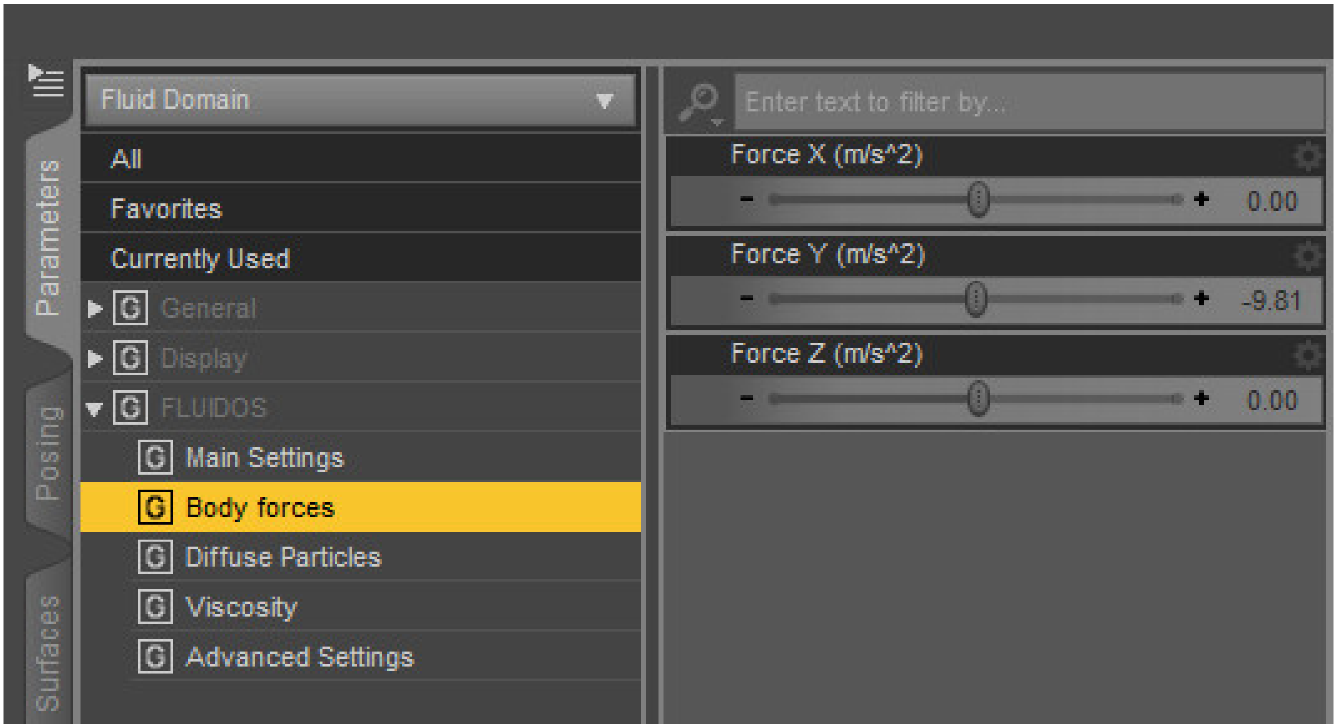
To get anisotropic meshes, the user must select the *Anisotropic* button.

Baked files folder: The user must set here the folder where the simulation files will be saved.

Erase baked files: enabling this property, the old files in the current folder will be erased the next time the simulator is running. However, if the user set ON “Continue saved state” option in running simulation, the *Erase baked files* property will be ignored.

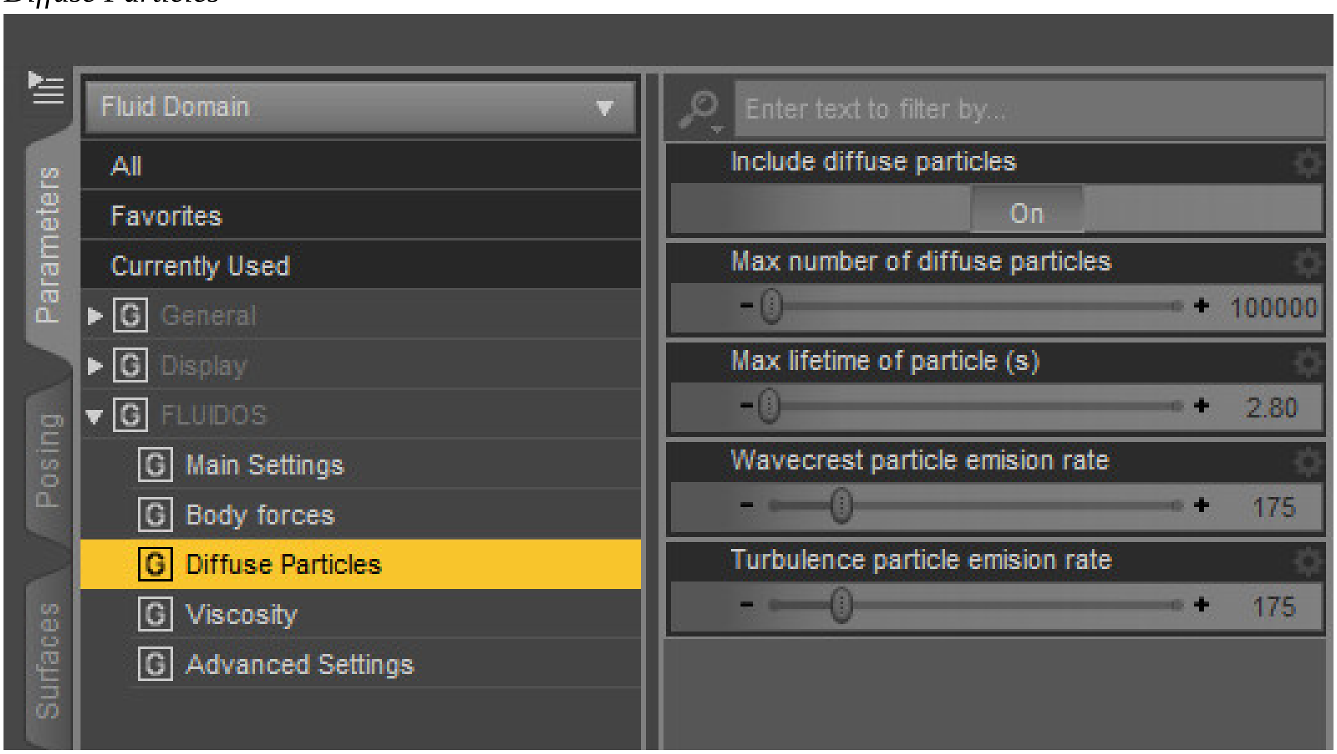
Enable moving obstacles: this option enables the simulation of the effect of the obstacle on the fluid.

Body Forces



Body forces: Add a spatially constant force such as gravity to the simulation. Nevertheless, this force is an animatable parameter. The unit are m/s^2.

Diffuse Particles



Diffuse particles:
The diffuse particle feature is a post-processing simulation run ontop of the PIC/FLIP fluid simulation that generates spray, bubble, and foam particles. These diffuse particles can be combined with a surface mesh in a render to give the fluid highly detailed small-scale aeration effects. The diffuse particles are stored as vertex only .PlY meshes where each vertex represents a single diffuse particle.

The diffuse particle meshes are written to the bakefiles/ subdirectory as a sequence of .PLY files prefixed with the keyword diffuse in the form diffuse000000.ply, diffuse000001.ply, diffuse000002.ply, ..., where the file numbers correspond to the frame number.

To enable diffuse particles select the *Include diffuse particles* check box. Beware, although the simulator can manage easily a great number of particles, its representation in Daz Studio can be very costly and give very slow rendering or locking of the program. You can limit the *maximal number* of diffuse particles with the slider.

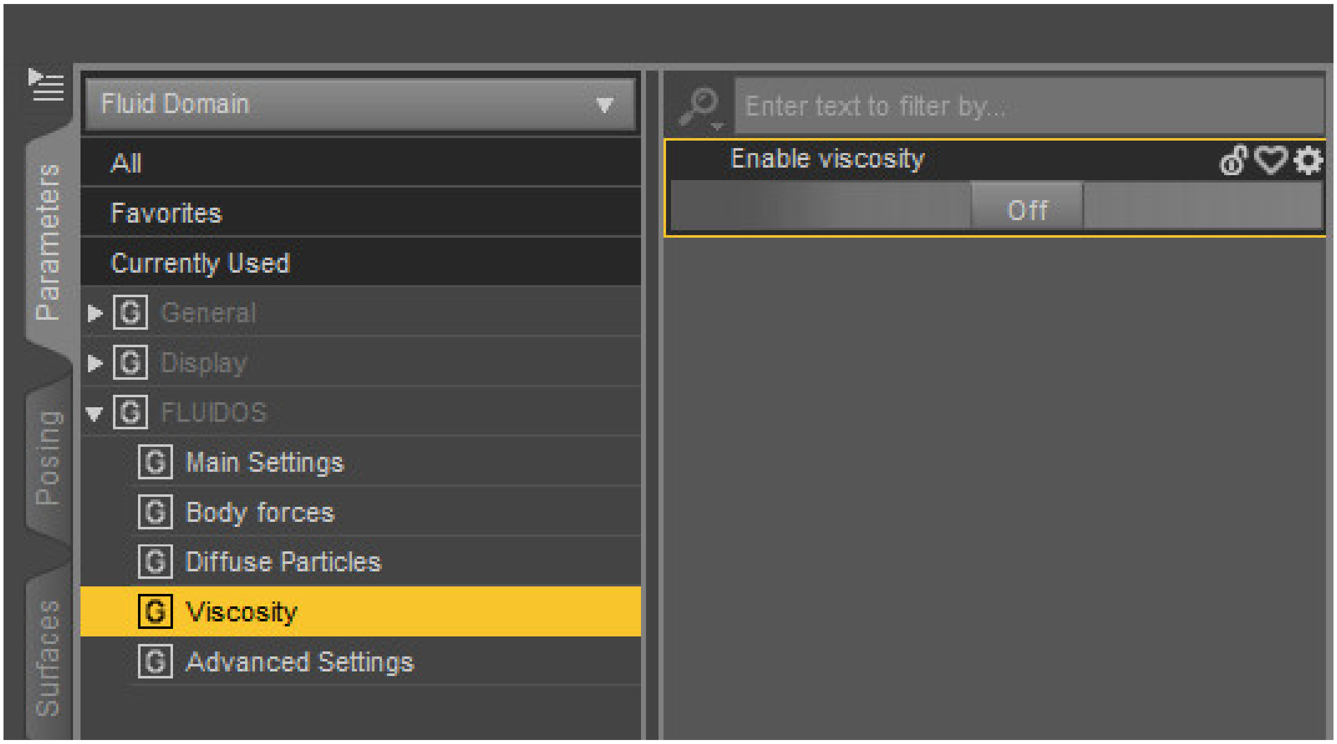
The maximum lifetime of a diffuse particle is spawned for in seconds. Set this value to control how quickly/slowly diffuse particles fade from the simulation.

Diffuse particle emission rates:
The diffuse particle simulator spawns particle emitters in areas where the fluid is likely to be aerated such as at wavecrests and in areas of high turbulence. The number of emitters spawned in an area is proportional to how sharp a wavecrest is and how turbulent the fluid is at a location.

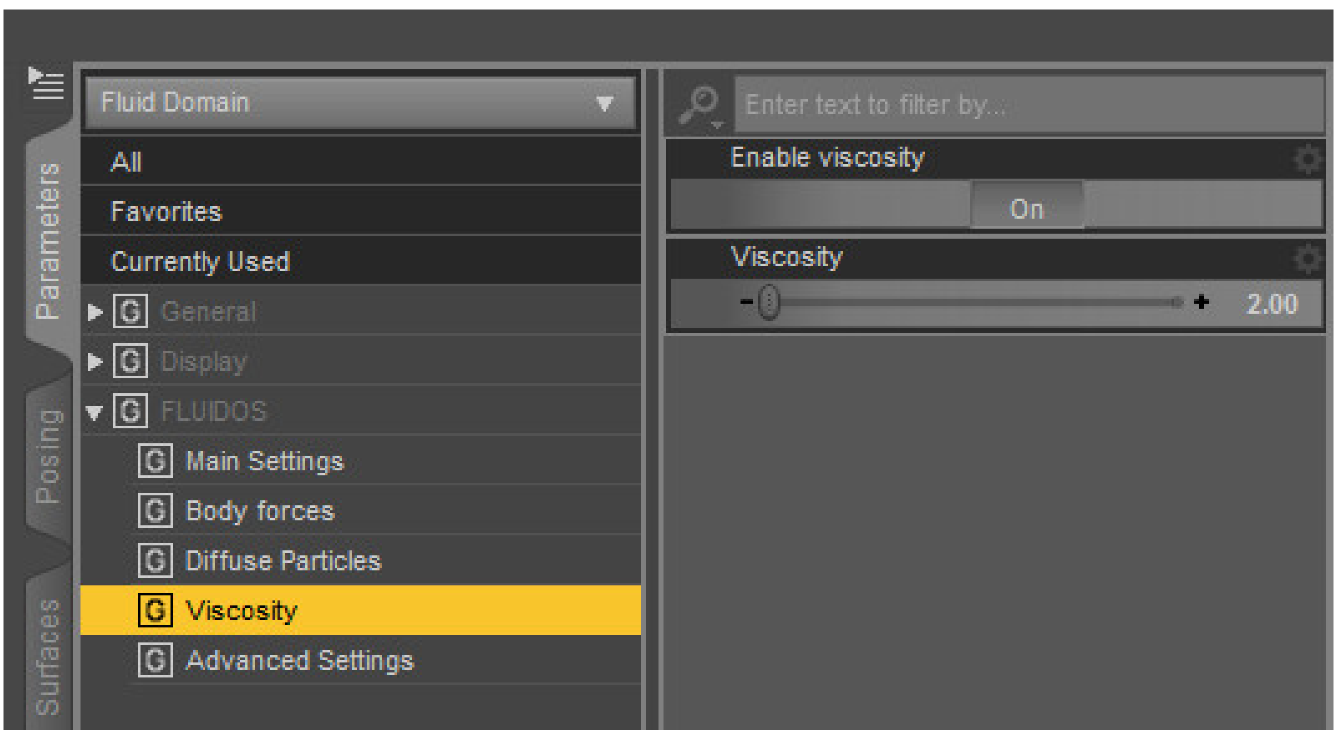
The number of particles generated by an emitter is controlled by two rates: *Wavecrest emission rate*, and *Turbulence emission rate*. An emission rate is the number of particles generated by an emitter per second. The wavecrest emission rate controls how many particles are generated by wavecrest emitters. The turbulence emission rate controls how many particles are generated by turbulence emitters.

An important note to make about emission rates is that the number of particles generated scales as the simulator dimensions scale. This means that a simulation with dimensions 128 × 128 × 128 will generate about eight times as many diffuse particles than a simulation with a dimension of 64 × 64 × 64 when using the same rate values.

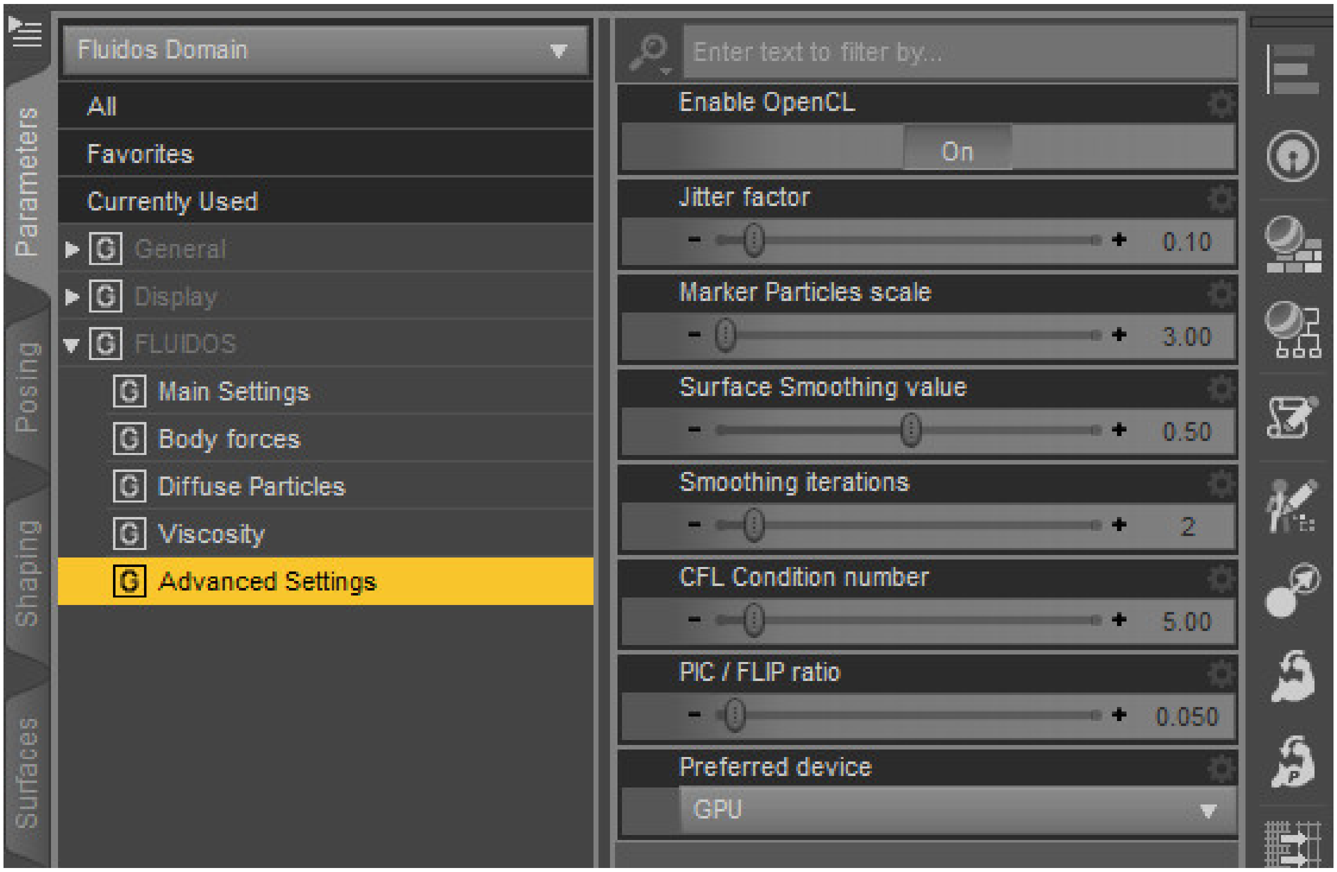
Viscosity



Viscosity: To simulate high viscosity fluids (as oils, honey, etc.), set *Enable viscosity* on and put a viscosity value. The value is animatable.



Advanced Settings



Enable OpenCL: this allows to switch off or on the OpenCL. The simulation without OpenCL is somewhat slower and the results are slightly different.

Jitter factor: This property controls the amount of random jitter added to newly spawned fluid particles. Sometimes, low values may produce symmetric artefacts into the resulting fluid simulation. They can be avoided increasing jitter factor.

Marker particle scale: Marker particles track where the fluid is and carry velocity data. Marker particle scale determines how large a particle is when converting a set of particles to a triangle mesh. A marker particle with a scale of 1.0 will have the radius of a sphere that has a volume 1/8th of the volume of a grid cell. It is very useful for viscosity simulations because sometimes the fluids freezes in air. Increasing the marker particle scale fixes this.

Surface smoothing value: The amount of mesh smoothing when generating the surface mesh.

Smoothing iterations: The number of smoothing iterations of the fluid geometry reconstruction.

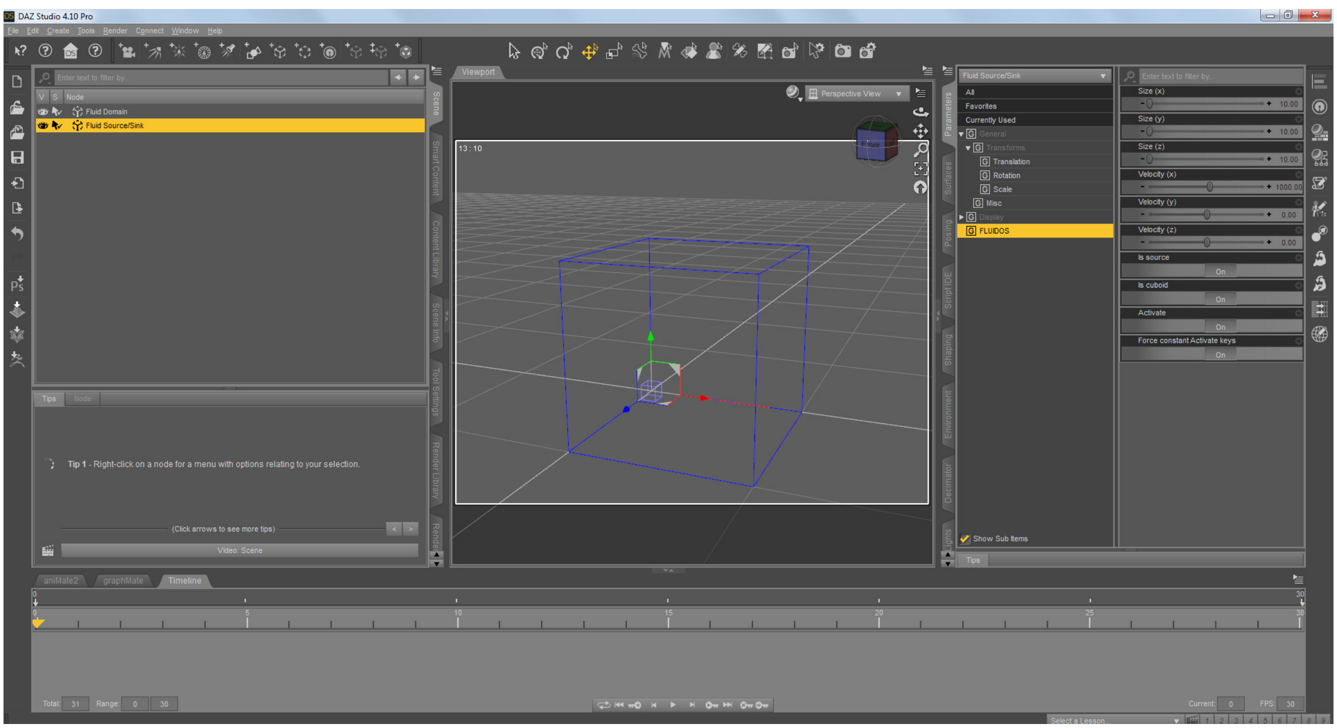
CFL Condition number: This is the maximum number of grid cells that a particle may travel in a single time step. **The larger is the number, the faster; the smaller, the more accurate.**
BEWARE: combining *low values* of CFL number with *small* cell size could cause the simulations take a lot of time to finish.

If you experiment long-time simulations, you may want to increase the CFL number.
The low limit of CFL in plugin is *0.1*. But **a *0.0* value disables the CFL condition number, so the simulations are the faster, but the less accurate**.

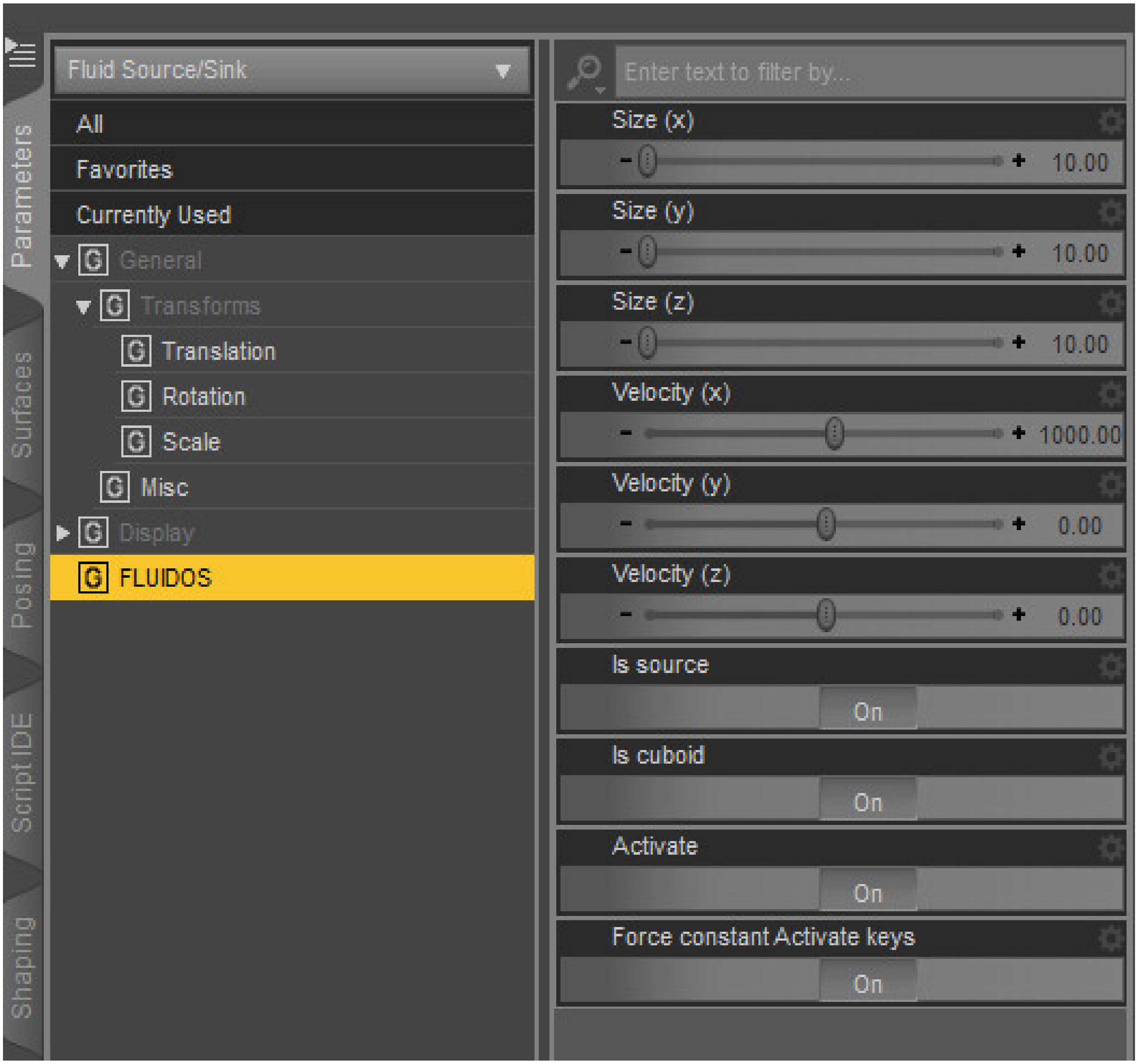
PIC/FLIP Ratio: The ratio of PIC velocity to FLIP velocity to use when updating particle velocities. Particle-in-Cell (PIC) and its variant Fluid-Implicit-Particle (FLIP) are the engine simulation methods. The PIC method is not very accurate, but stable. The FLIP velocity method is very accurate, but less stable. Using a value of *0.0* results in a completely FLIP simulation, while using a value of *1.0* results in a completely PIC simulation.

Preferred OpenCL device: The primary OpenCL device (GPU or CPU). For some computer systems, CPU is better. If you have a second GPU, you can select it (“GPU 2”).
Note 1: If your system has only one GPU, and you select GPU 2, the plugin will use the unique GPU.
Note 2: Some graphics cards (as some AMD and Intel) can yield incorrect simulations (the fluid goes to the Domain top in every simulation, no matter the settings). In this case, select CPU as preferred device to solve the problem (you could, instead, disable OpenCL, but the calculations will be something slower).

Fluidos Source / Sink



The source (or the sink) are represented as an empty cube, but can be resized the same the fluid domain.
Parameters of Fluidos Source / Sink



The size can be controlled with the parameters *Size (x)*, *Size (y)* and *Size (z)*, the units are centimeters. The velocity of flux is in cm/s and is controlled with *Velocity (x)*, *Velocity (y)* and *Velocity (z).*
Size and velocity are animatable parameters

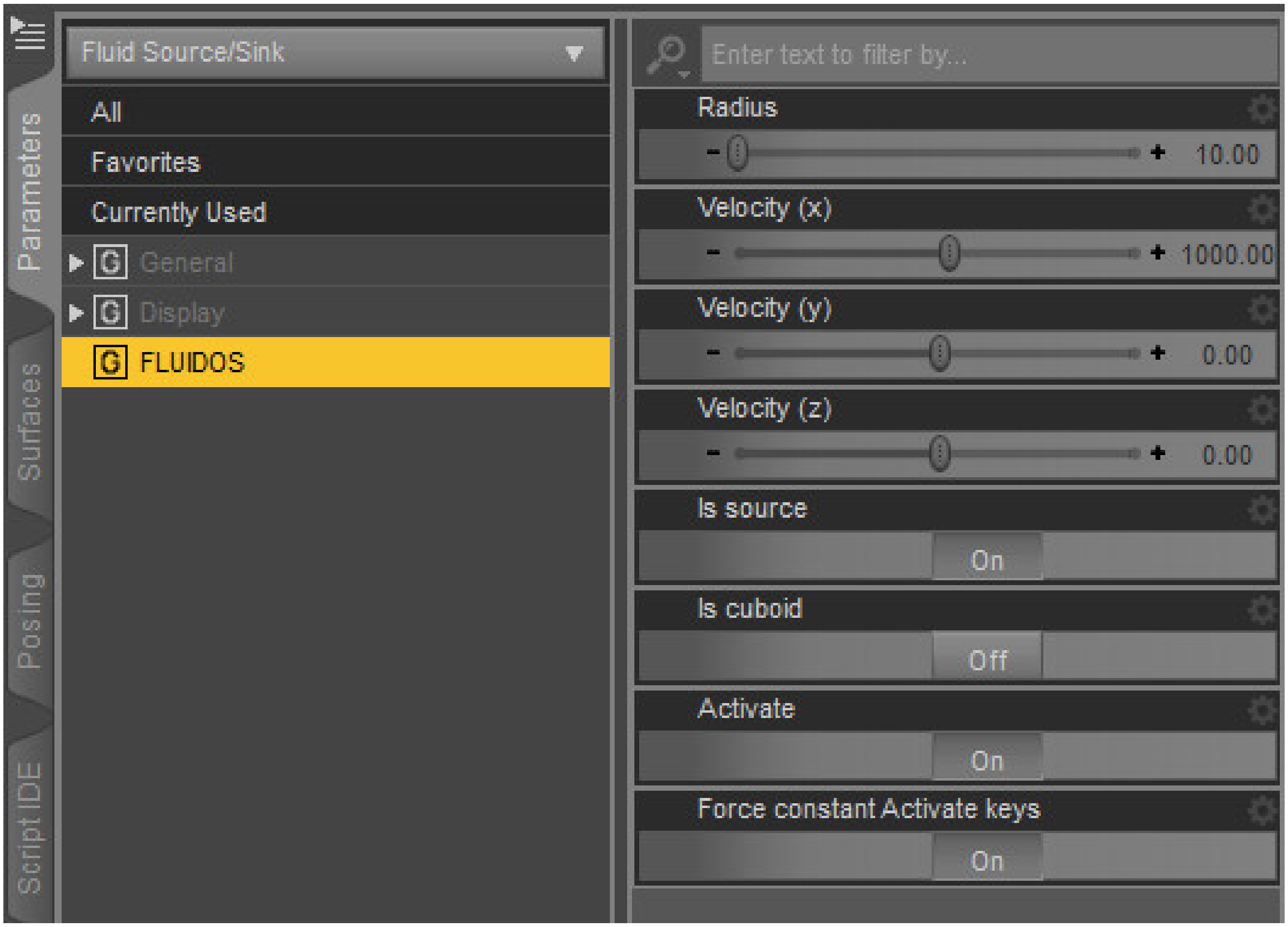
Is source: Let *ON* to get a fluid source, *OFF* to get a fluid sink.

Is cuboid: If is *ON*, the geometry of the source (or sink) is a rectangular prism; *OFF* produces a spherical geometry (not visible in the screen representation of the object).

Activate: *ON* to get the source (or sink) active, *OFF* to deactivate; this is animatable (e.g. you can interrupt a flux at any time).

Force constant Activate keys: if this parameter is turned *ON*, the keyframes of *Activate* are treated as constant interpolation: if the user add a first keyframe as *ON* and the next as *OFF* for *Activate* parameter, the plugin will add a *OFF* key just a frame before the last one. This way, the source is active until the user set *OFF.*

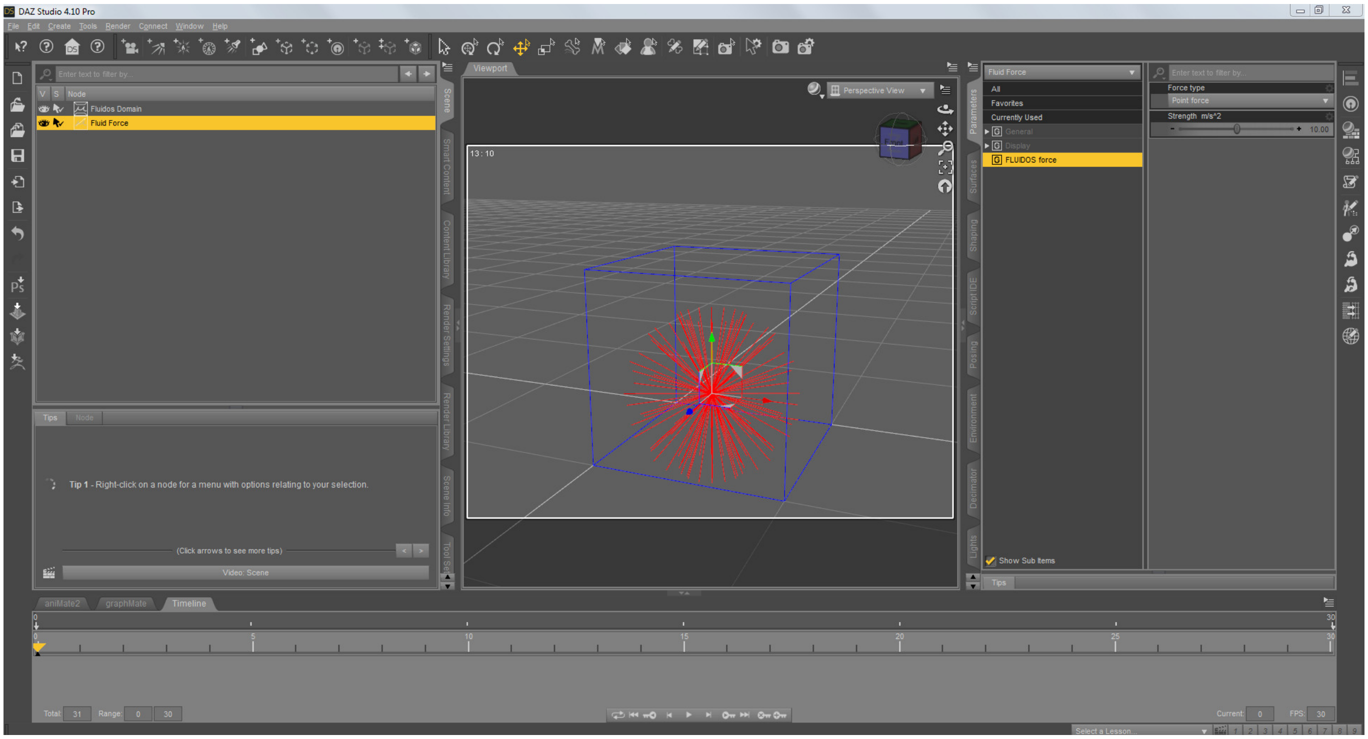
When *Is cuboid* is *OFF*, the size of the object is controlled by only one number: the *radius* (cm), as it behaves as a sphere in the simulation.



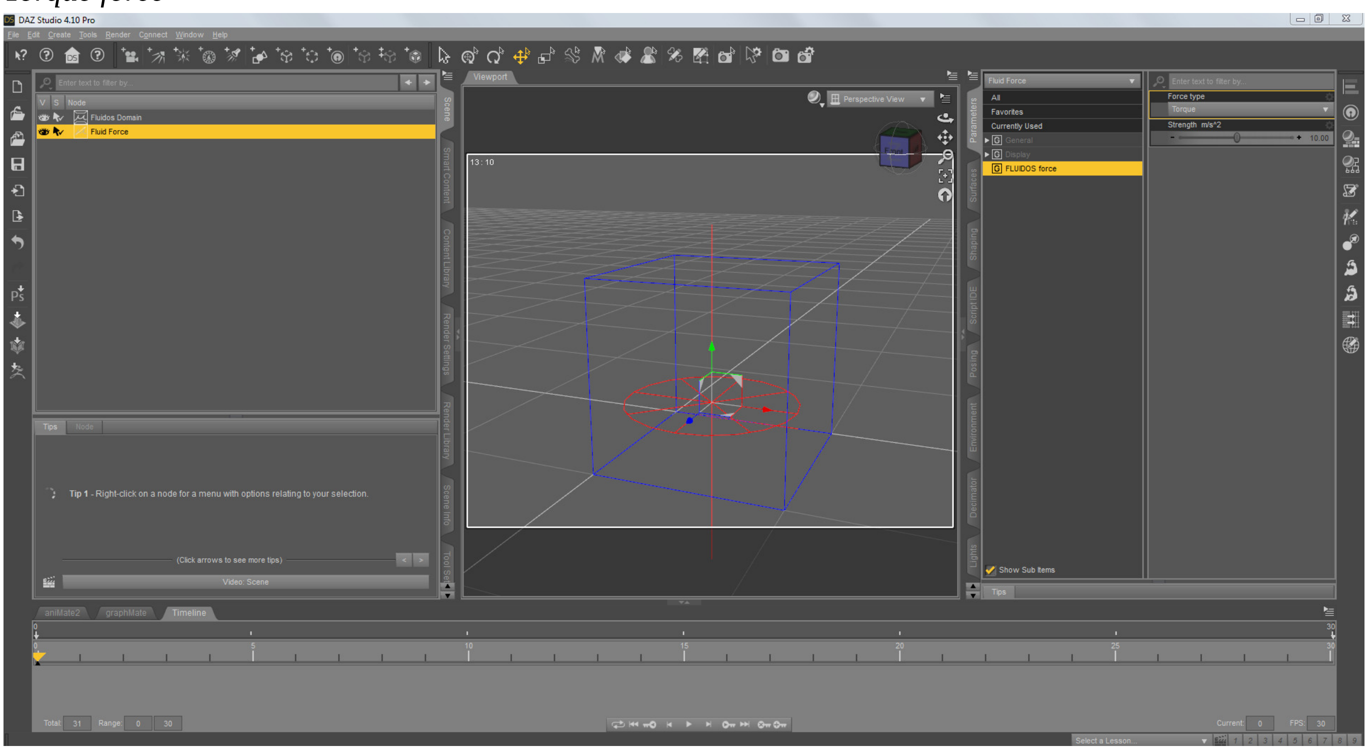
Fluidos Forces

The Fluidos force is represented this form:

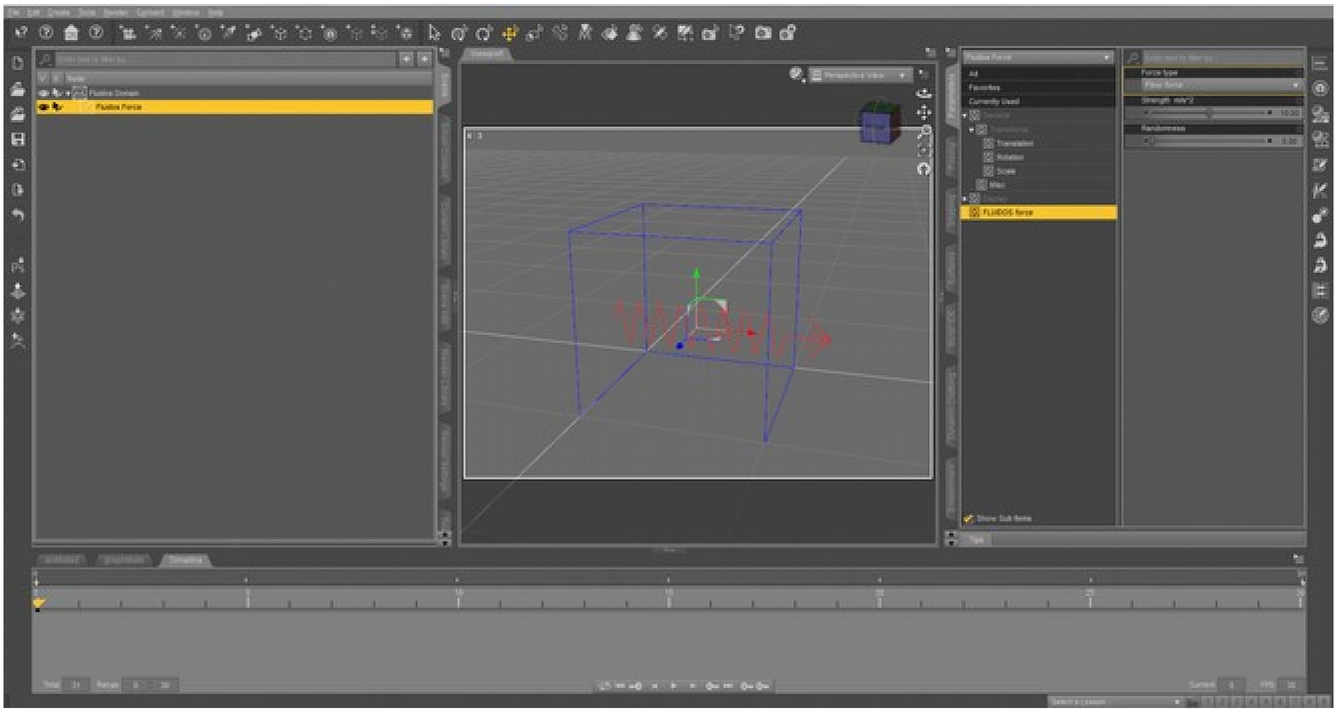
Point force



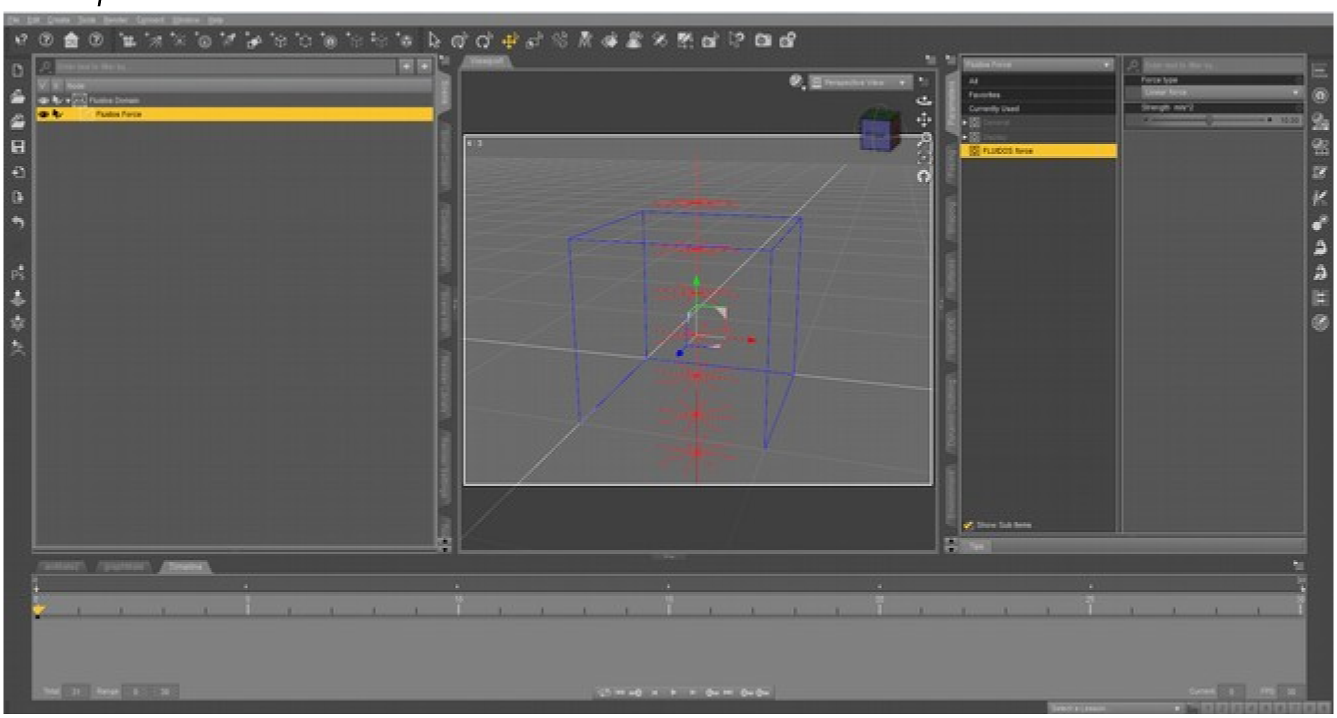
Torque force

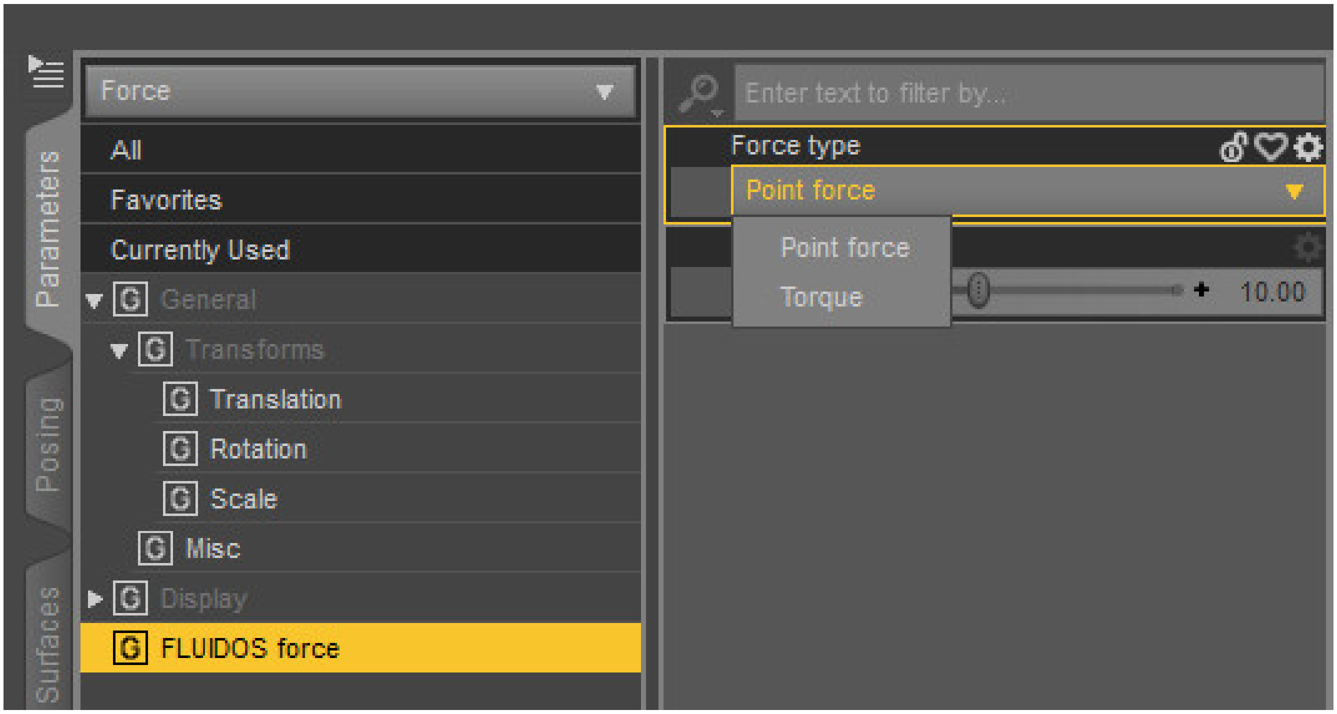


Flow force

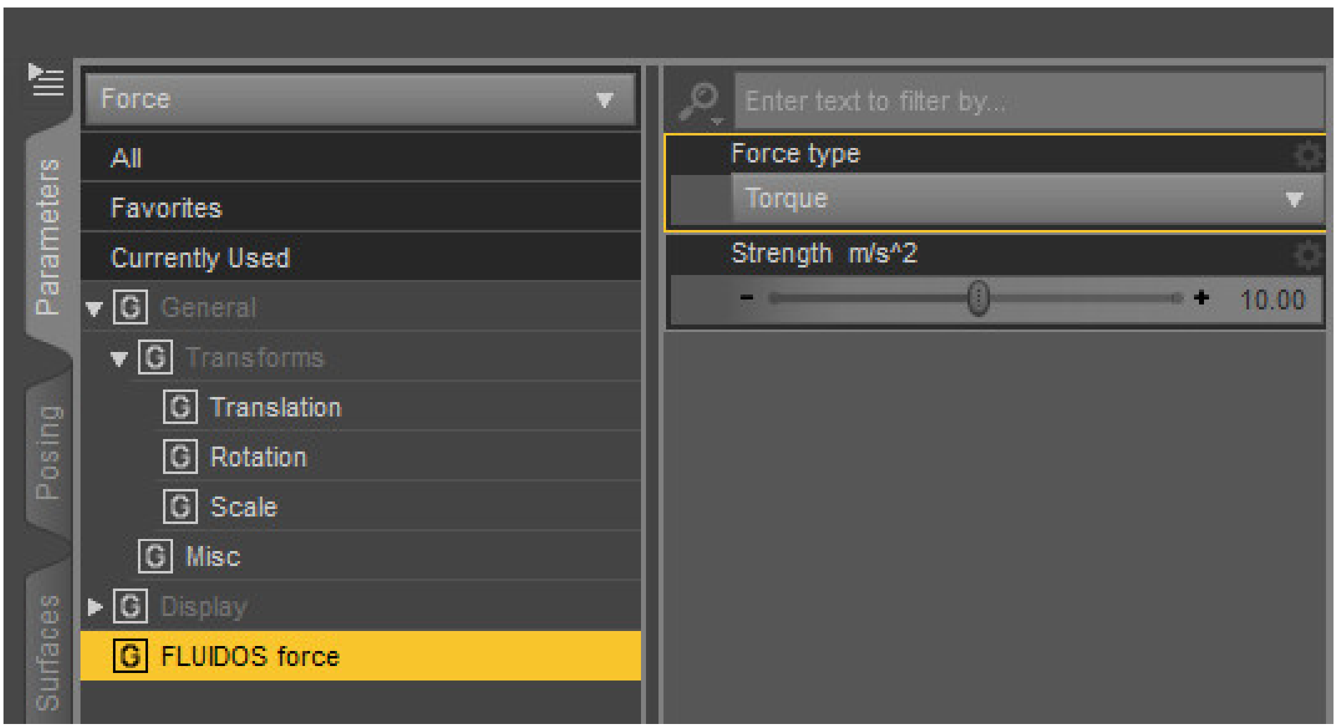


Linear force

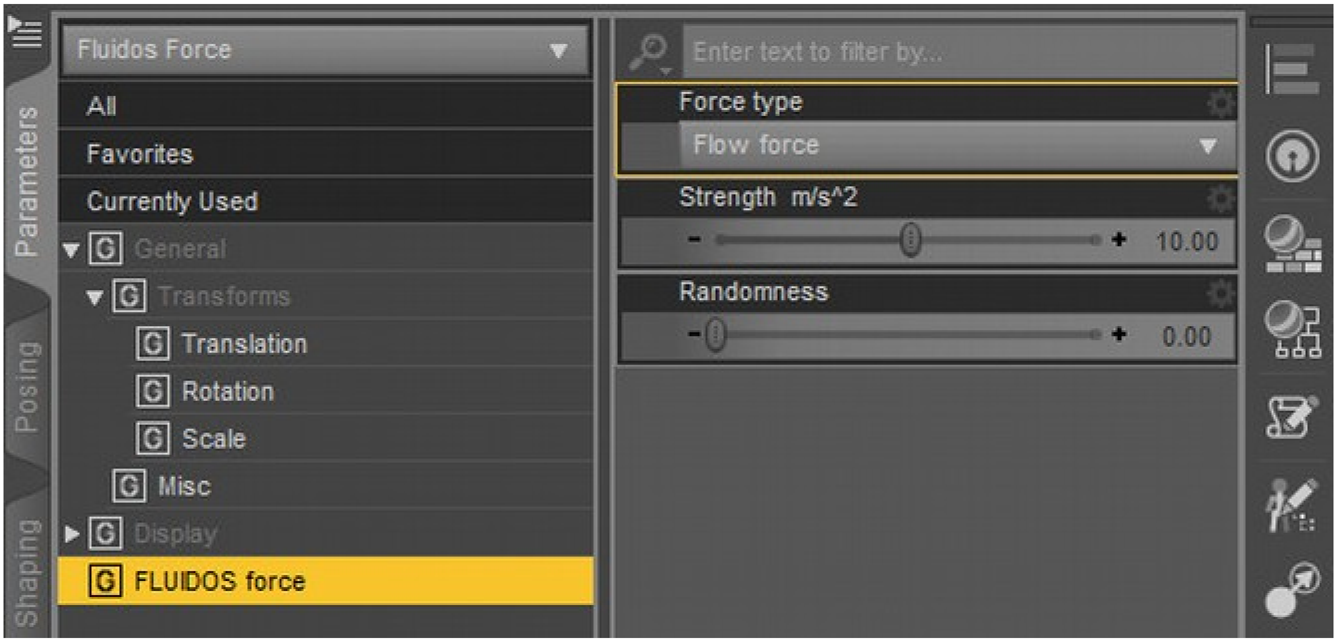




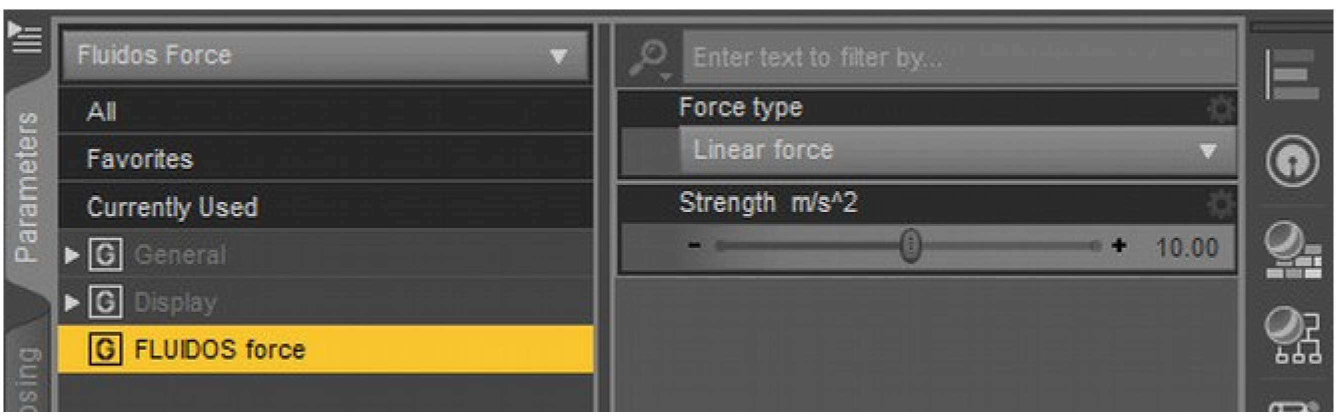
Point Force: This force extends its action form a point in space, the decay rate is quadratic. Its unis are in *m* /s^2 (acceleration).



Torque Force: this is a rotational force. As in the case of point force, the position is relevant, but rotation too. Its unis are in *m* /s^2 (acceleration).



Flow Force: This force extends its action over all the fluid free surface (not in solid contact). Its unis are in *m* /s^2 (acceleration). The *randomness* property controls the waves density on the fluid surface. The position of this force in not relevant, only the rotation.



Linear force: This force extends its action form a straight line in space, the decay rate is quadratic. Its unis are in *m* /s^2 (acceleration). The position and rotation are relevant variables.

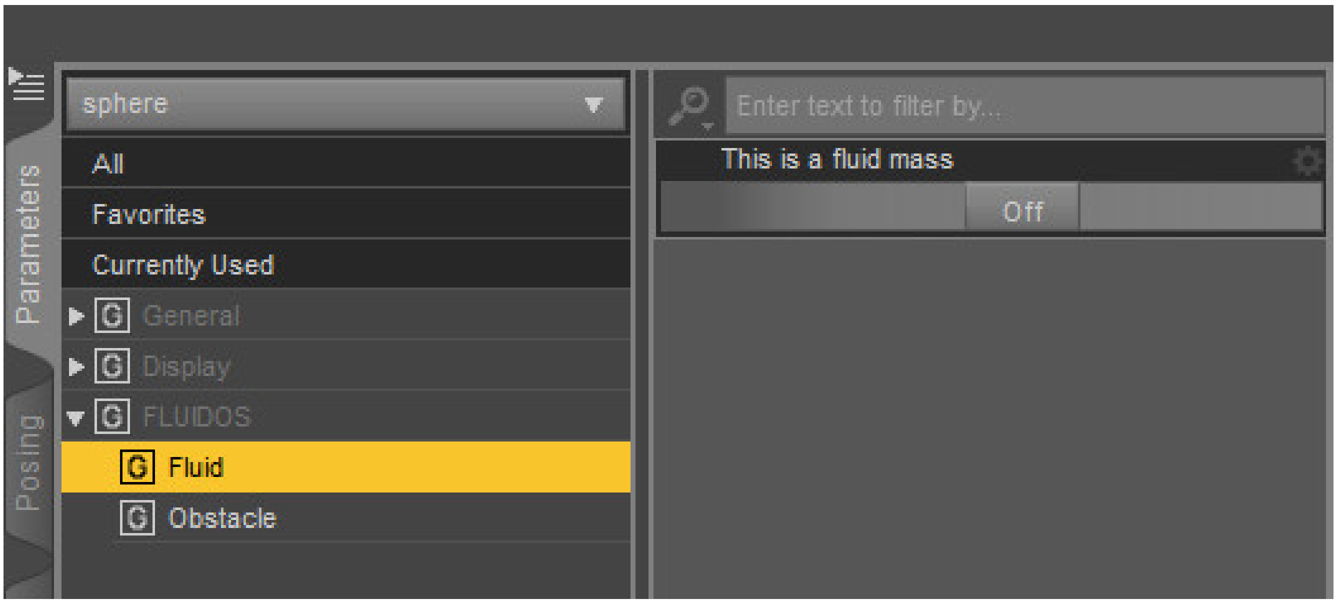
Solid or Fluid Properties:

When any geometric object is parented directly or indirectly to a Fluid Domain, will acquire some FLUIDOS parameters

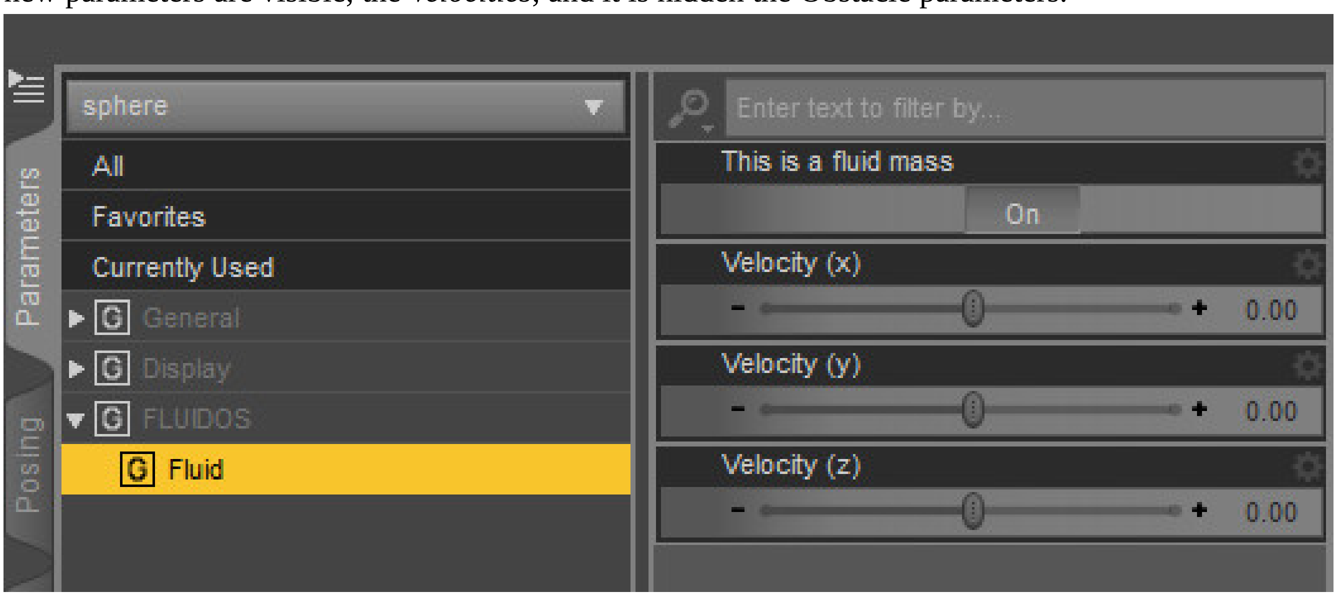
There are two groups of parameters:

- Fluid
- Obstacle

Fluid

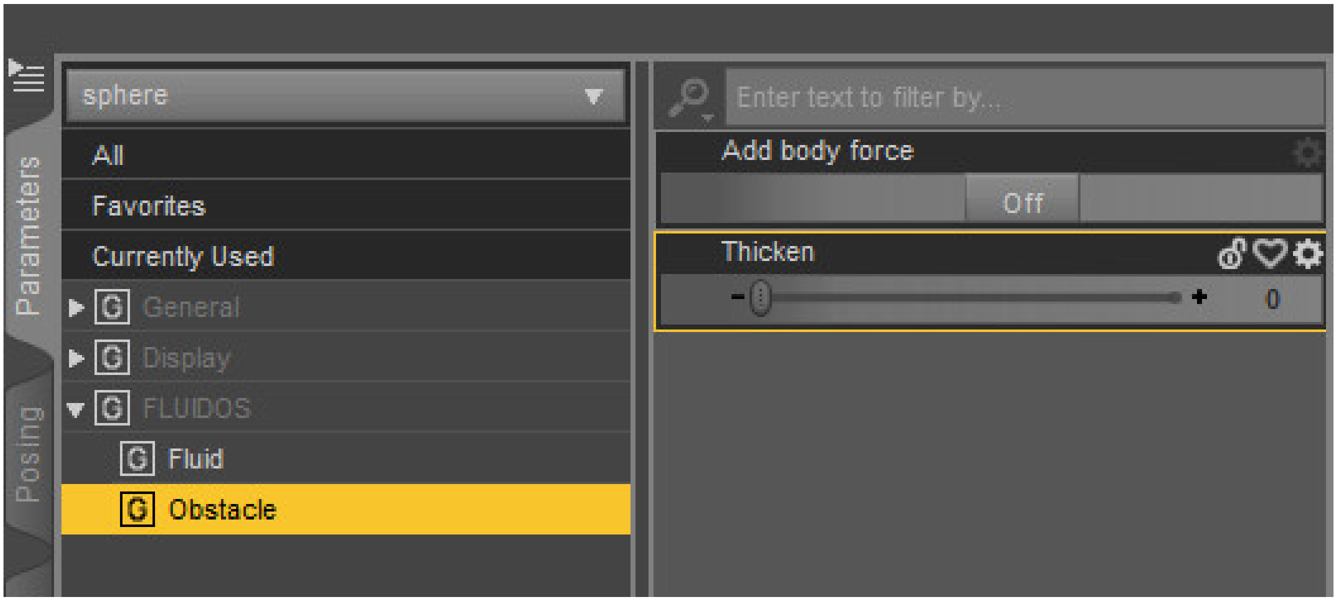


If *This is a fluid mass* is *OFF*, the object are considered a solid obstacle. But if this parameter is set *ON*, new parameters are visible, the *Velocities*, and it is hidden the Obstacle parameters.



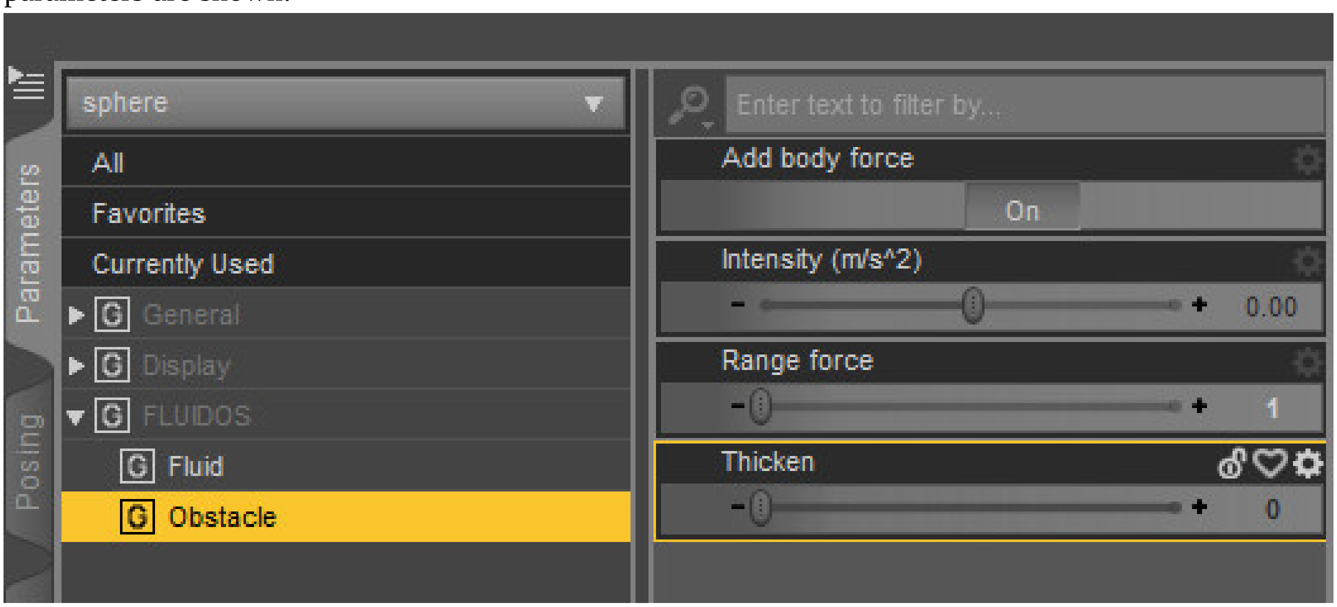
The initial velocity of the fluid mass is in cm/s and is controlled with *Velocity (x)*, *Velocity (y)* and *Velocity (z).*

The *Obstacle* parameters is shown when *This is a fluid mass* is *OFF.*



Thicken: this option add invisible layers in the solid grid. These thicken the solid walls so the fluids cannot spill out. Each extra layer is a cell size width. It is animatable.

Add body force: here, it is a surface force linked to the solid obstacle. If this button is *ON*, the next parameters are shown:

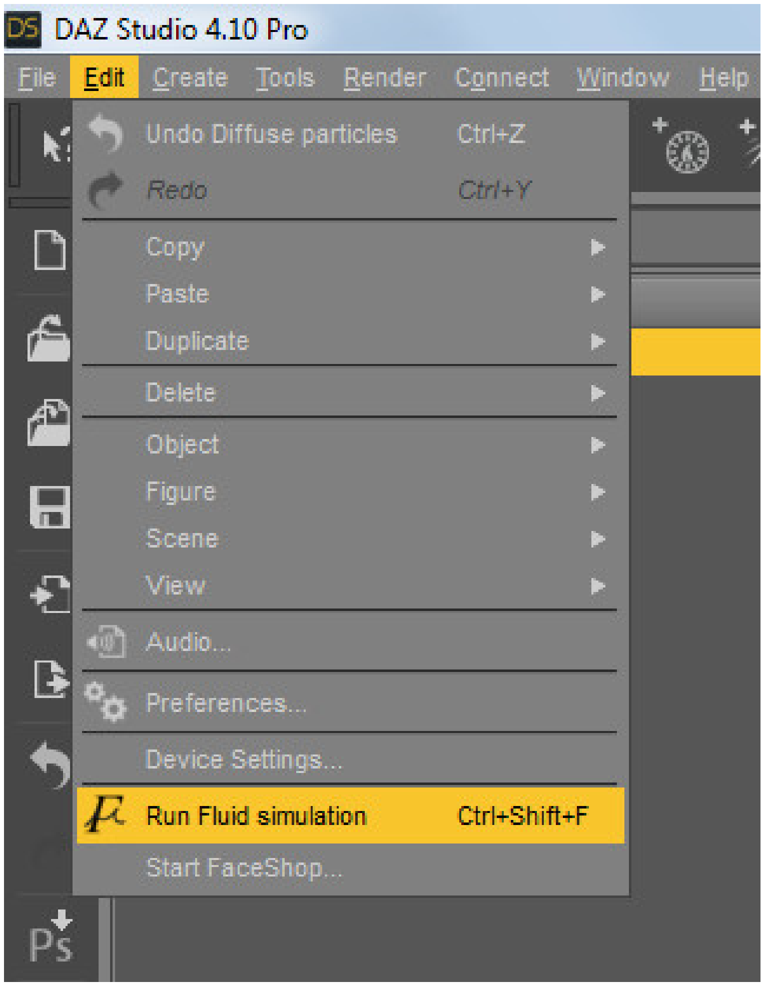


Intensity: this set the intensity of the force, it can be positive or negative. It is animatable.

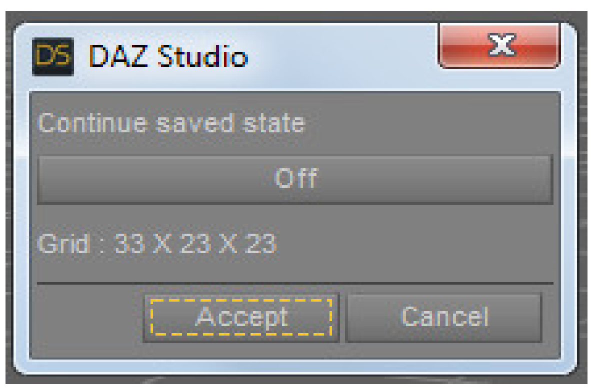
Range force slider allows to “thicken” the extent of this force to one or two cells away the solid.

Running a simulation

To run a simulation, all the objects involved in it, must be parented to a Fluid Domain. Then select the Domain and go to the menu *Edit – Run Fluid simulation* (Ctrl+Shift+F is a shortcut).



The next box then appears.



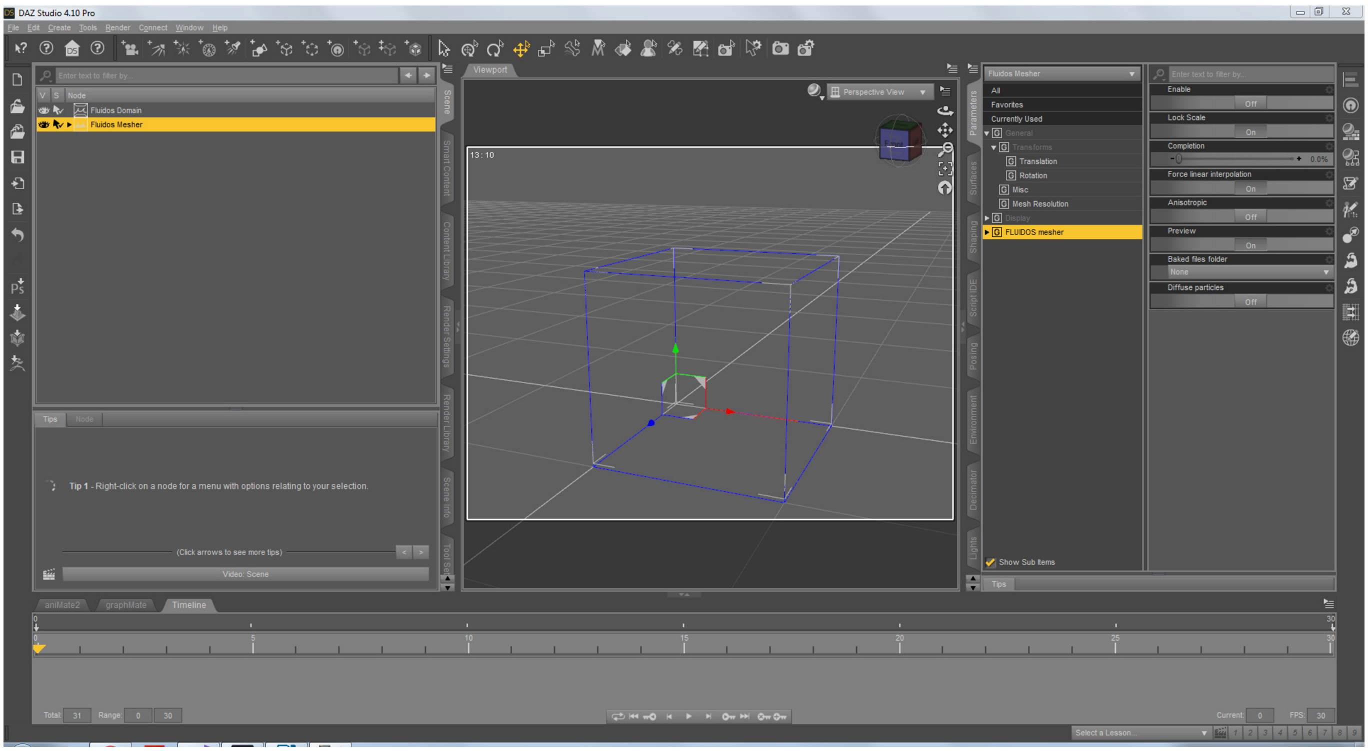
If there is a previous simulation baked in the selected folder, it can be resumed or extended for more frames by setting *ON* the *Continue saved state* button.

The box tells the Grid size. The larger the grid, the slower the simulation and the bigger the fluids geometries.

Push *Accept* to run the simulation. The simulation can be canceled by clicking in the *Cancel* button of the Progress bar (or pushing the Escape key). As the engine is unresponsive when is calculating a frame, sometimes seems to be locked (in complex simulations), be patient.

Visualizing the results of simulation

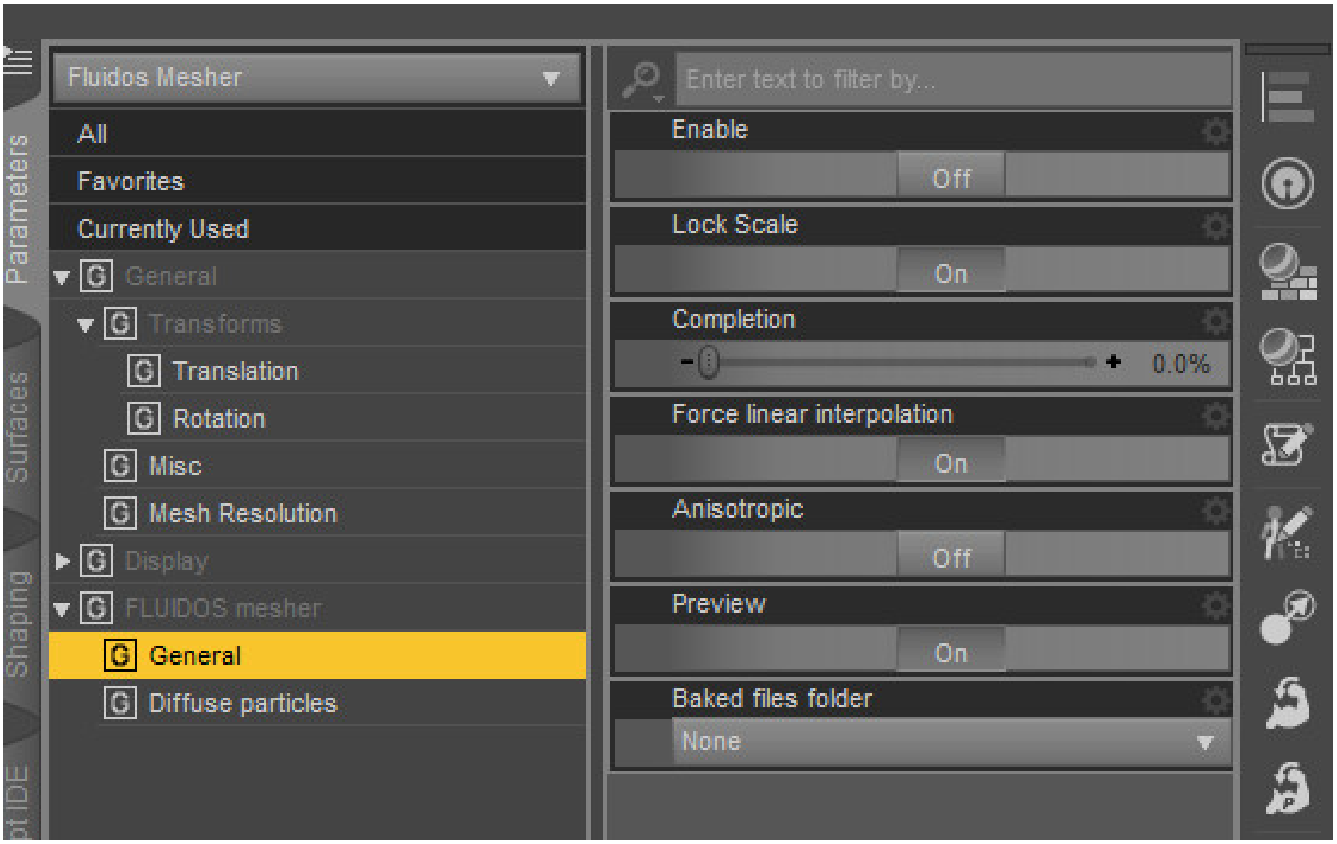
The Fluidos Mesher is the object that can show the results of the simulation:



The user must insert the Mesher at exactly the same position of the Fluidos Domain.
Its parameters are in two groups:

- General
- Diffuse particles

General parameters



Enable: Set *ON* this button to enable the modifier. If the simulation is extended (new frames calculated) and the Mesher is enabled, it must be disabled and re-enabled to update.

Lock Scale: set *ON* to prevent accidentally resize the Mesher. The Mesher must not be resized if the user wants to see the results of the simulation matching the original set up of the scene. But, in other case, setting this parameter *OFF* allows the access to scaling parameters

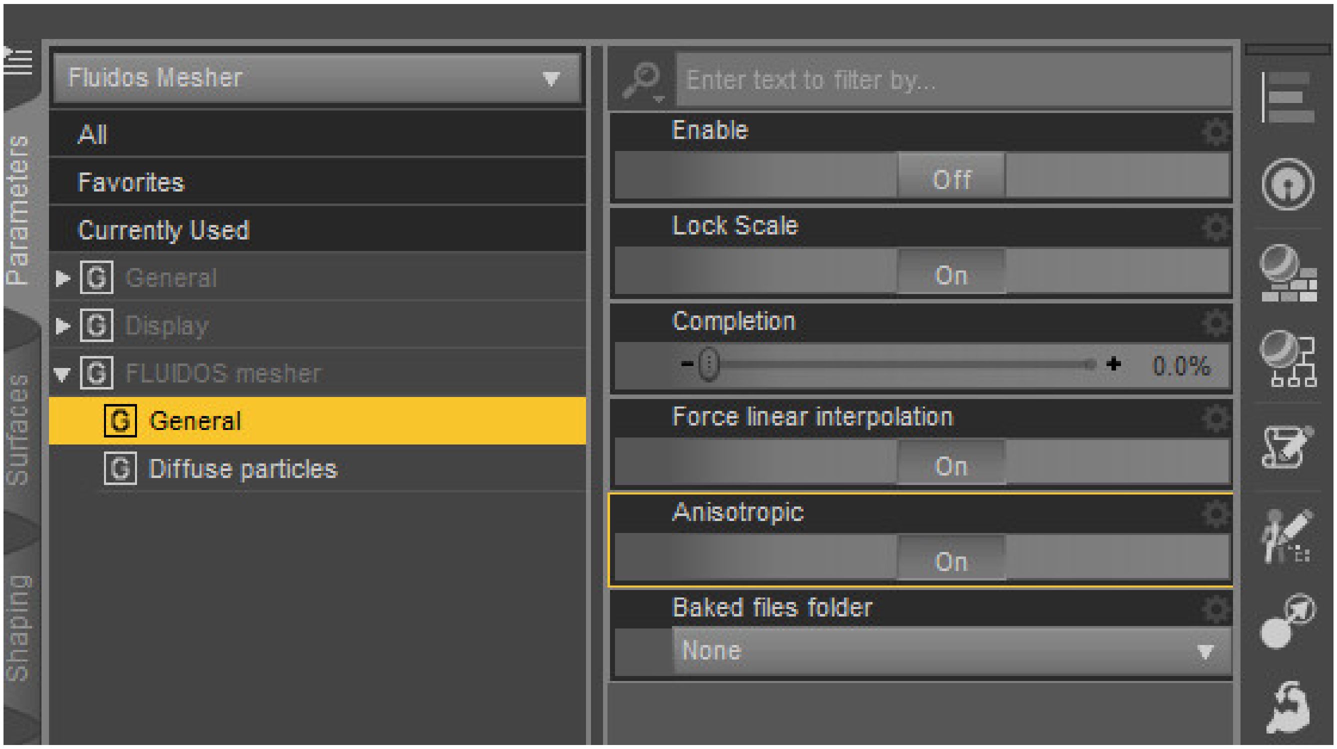
Completion: controls the animation the fluid simulation. Set it to 0% at the start of your animation, and at 100% at the end. This is the only Mesher parameter that is animatable.

Force linear interpolation: The default interpolation of the key frames in Daz Studio is non linear, but to get a correct synchronizing of the simulation in Mesher with the scene, a linear interpolation is needed. Setting ON this button forces a linear interpolation of the key frames of the Mesher (other option is to use a plugin as GraphMate).

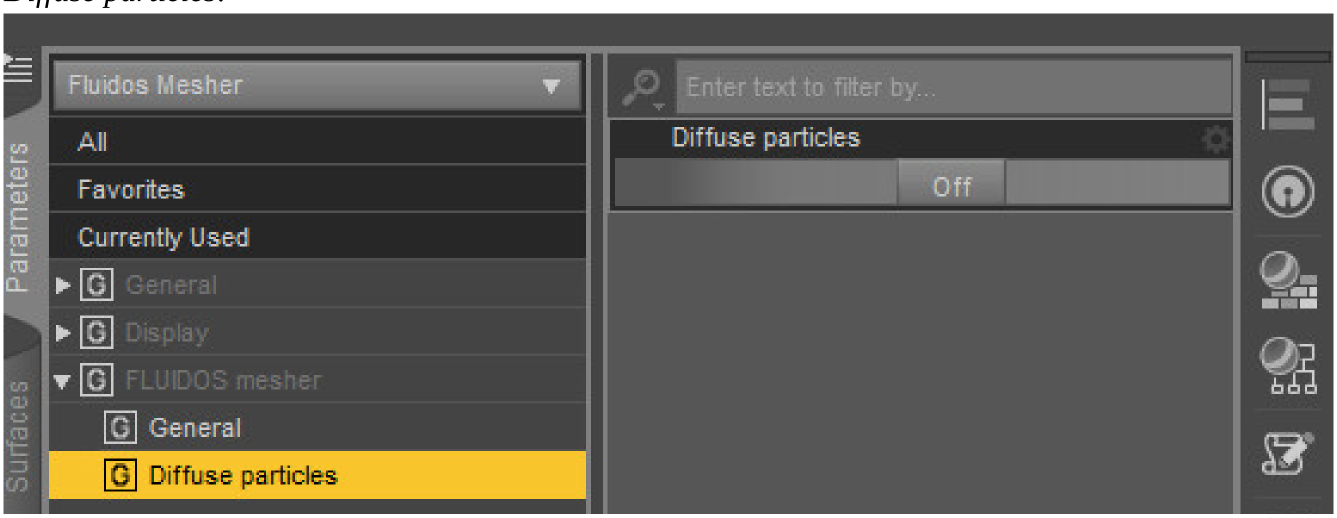
Anisotropic: Set *ON* in order to the Mesher can read the anisotropic files if they exist (the isotropic and anisotropic files are saved in the bakefiles/ subdirectory). By default, the Mesher reads only the isotropic files. Of course, the isotropic or anisotropic files must be created previously by running a simulation.

Preview: by default is set *ON*, so the Mesher shows a preview, low resolution version of the fluid simulation. It is recommended to avoid as much as possible to set *OFF* except during final rendering because extreme simulations can be very slow to get rendered in the viewport of Daz Studio.

Baked files folder: use this to select a folder of fluid simulation.
If *Anisotropic* is set *ON*, *Preview* is hided because there are no preview generated for anisotropic reconstruction:

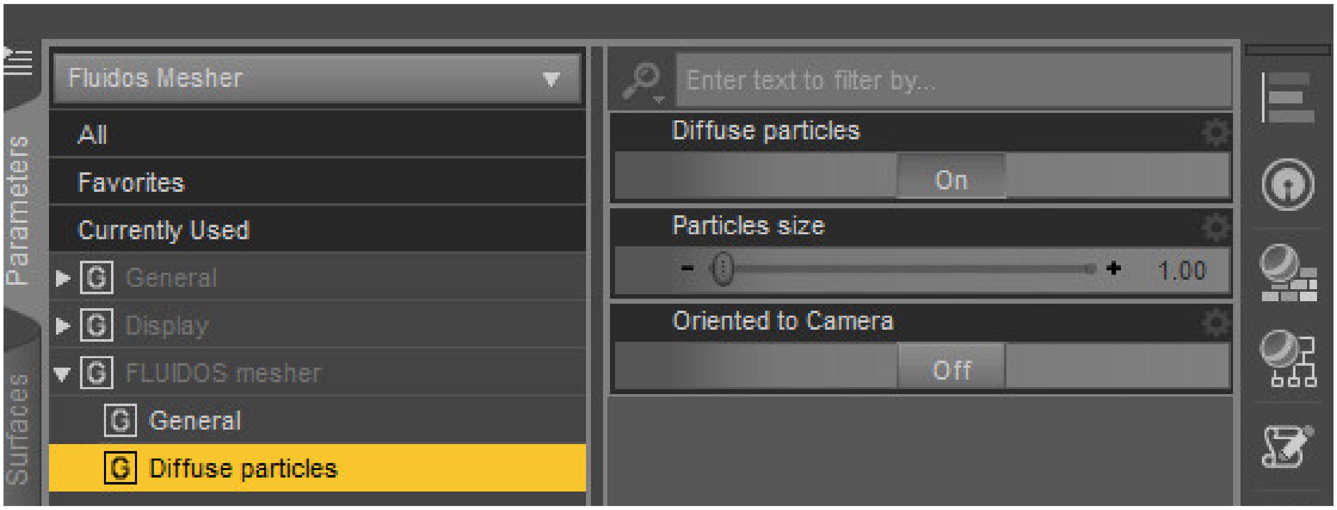


Diffuse particles:



Diffuse: Set *ON* this option to get the modifier reads **only** diffuse files if they exist. The Mesher can not read diffuse and not diffuse files at the same time. If you want to render both simulations, use one

Mesher for diffuse and another one for not diffuse. There is no other limit for the number of Meshers in scene than the memory system.



Particles size: the diffuse particles can be resized (the default value is 1.0), this is an animatable parameter. The particles are triangles than can be used as billboards.

Oriented to camera: this orients the particles to face the opposite direction the camera is aimed.

References:

1. Guy L. Ryan. 2016. PIC/FLIP Fluid Simulation. http://rlguy.com/gridfluidsim/

2. Guy L. Ryan. 2017. FLIPViscosity3D https://github.com/rlguy/FLIPViscosity3D

3. Batty C and Bridson R. 2008. Accurate Viscous Free Surfaces for Buckling, Coiling and Rotating Liquids. In Proceedings of the 2008 ACM/Eurographics Symposium on Computer Animation.

4. Batty C. 2008. VariationalViscosity3D https://github.com/christopherbatty/VariationalViscosity3D

5. Bridson, R., & Müller-Fischer, M. 2007. Fluid simulation: SIGGRAPH 2007 course notes. In ACM SIGGRAPH 2007 courses. https://www.researchgate.net/publication/234801901_Fluid_simulation_SIGGRAPH_2007_course_notes_Video_files_associated_with_this_course_are_available_from_the_citation_page

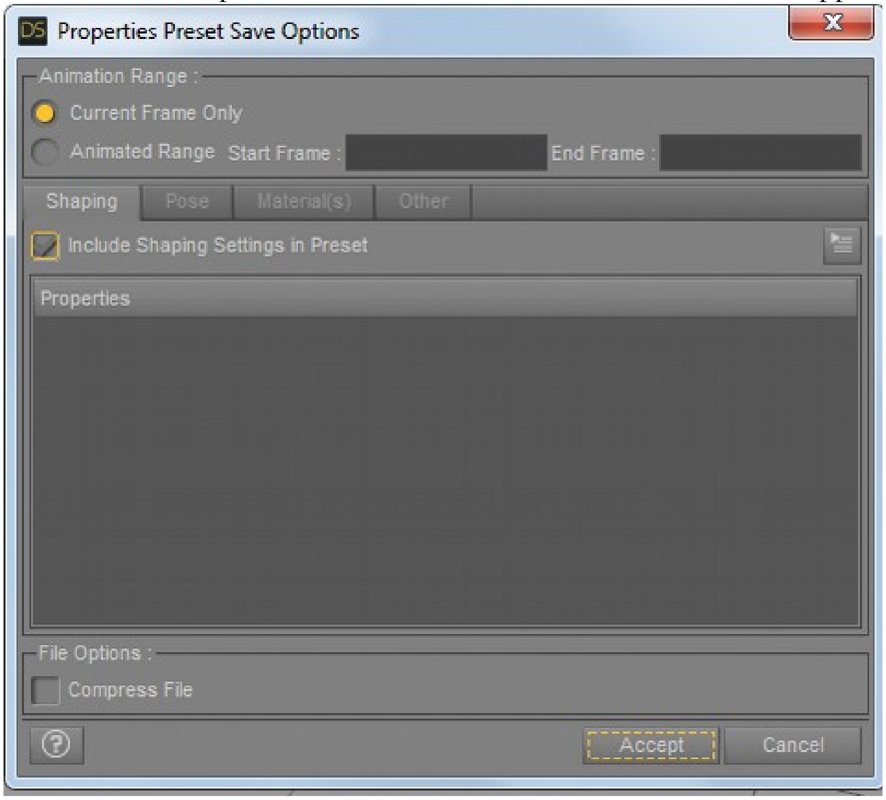
6. Bridson, R. (2016). Fluid simulation for computer graphics. 2nd ed. CRC Press.

About the engine simulation fluidsim.dll:

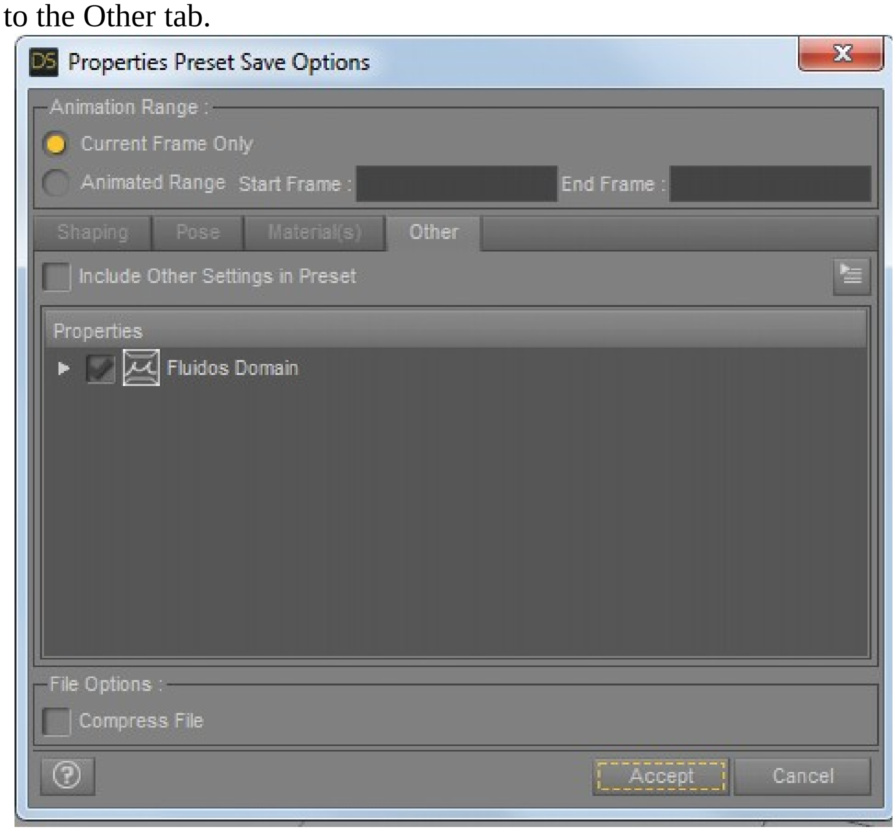
The engine simulation (fluidsim.dll) is a modified version of GridFluidSim3d licensed under zlib license, by Ryan L. Guy. The engine uses some modified codes of FLIPViscosity3D licensed under MIT license as:

APENDIX: How to create a Properties preset for the Fluidos Domain.

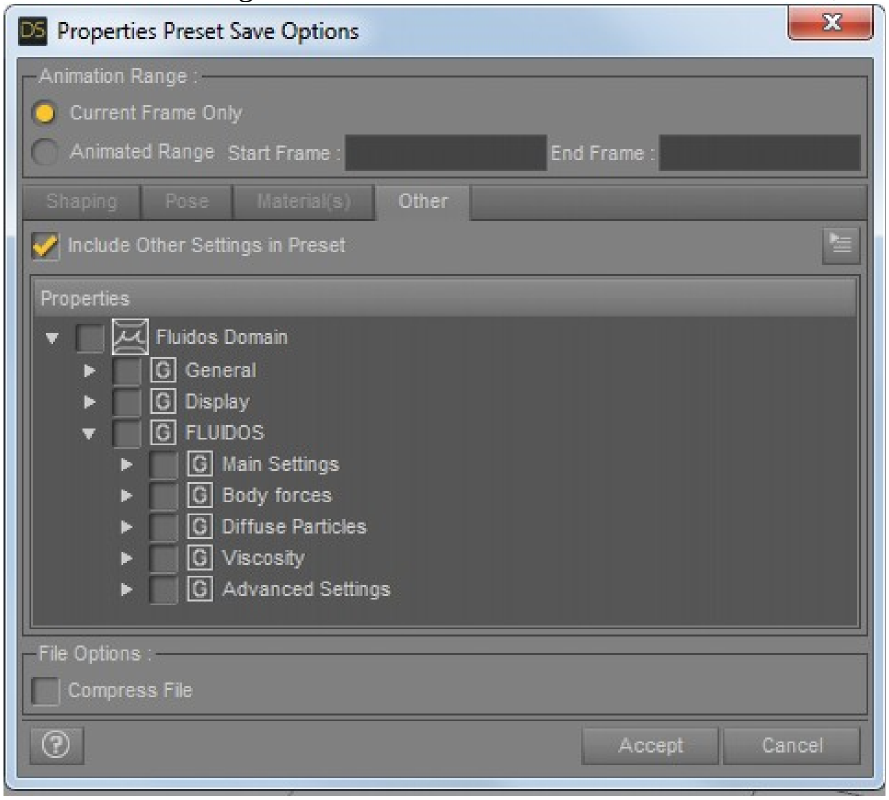
1. Select the Domain. Set the Baked Folder you always use.
2. Go to "File - Save As - Properties Preset" menu. A window like this will appear:



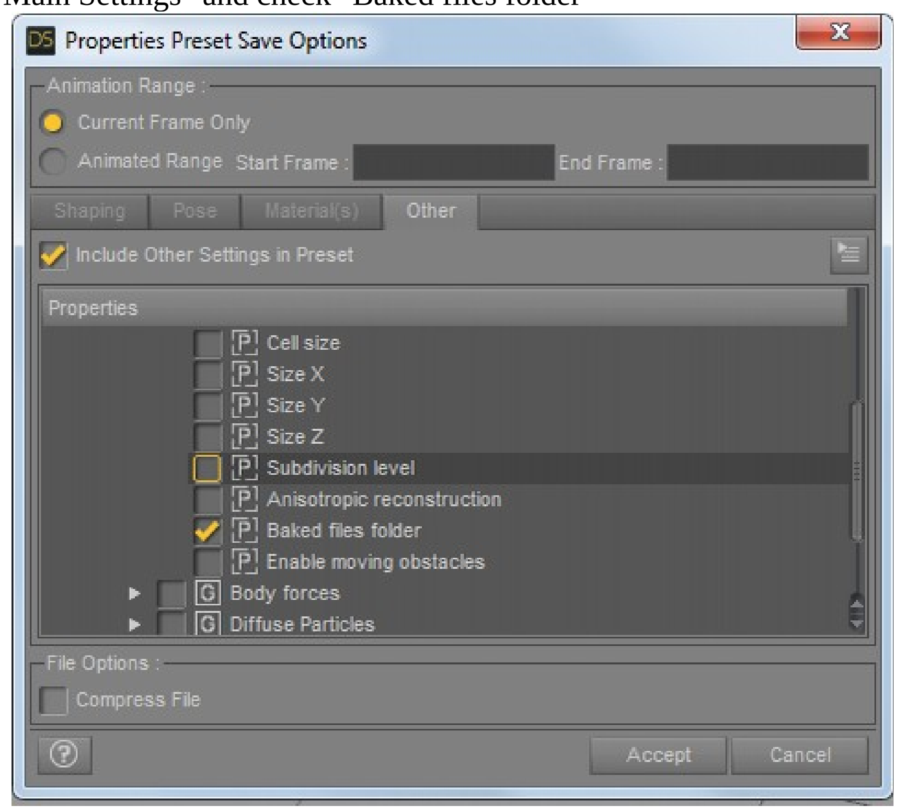
3. Uncheck "Include Shaping Setting in Presets". Do the same in Pose and Materials(s) tabs.
4. Next, go to the Other tab.



5. Check "Include Other Settings in Preset". Next uncheck "Fluidos Domain", and expand.



6. Expand "Main Settings" and check "Baked files folder"



7. Click Accept.

8. You have now a Properties preset only for the Baked folder. Aply to any domain or mesh you wish.

You can choose other properties besides the Baked files folder. The same property preset can be used for Domains and Meshes. If you add an only-Domain-property to your preset, when you apply this to a Mesher, the only-Domain-property will be ignored.