TurbulenceFD for CINEMA 4D

Introduction

Quick Start

The best way to get started quickly is to load any of the example projects in the Examples sub-folder of your TurbulenceFD installation and experiment with the settings. To create a scene from scratch, follow the following steps:

- Select TurbulenceFD Container from the Plugins/TurbulenceFD Menu to create a fluid container for the simulation
- Create one or several objects like spheres or such and move them inside of the container
- Scale the objects down (or scale the container up) such that there's enough empty space in the container for the smoke/fire the spread
- Attach the TurbulenceFD Emitter tag to all objects that will contribute to the simulation
- Set the tag parameter Channels/Temperature Value to 1.0
- Select Start Cached Simulation from the Plugins/TurbulenceFD menu.
- A dialog with a progress bar appears as the simulation is being computed.
- You can use Preview Fluid Container from the plugin menu to show a low quality preview of the current frame

Painting fluids

One important aspect of controlling a fluid simulation (if not the most important one) is to create the right emitter setup.
You can think of the emitter as a brush used to paint in a pixel image. In fact, the container works pretty much like a 3D pixel image. You paint into one or more of the fluid channels (similar to painting into the RGB and/or A channels of an image) using emitters.
The fluid simulation will then add velocities that let your "paint" flow, curl and/or expand. But, and that's the point here, essentially the result will be what you painted at the source. You'll want to try various settings for the emitter texture scale, octaves and contrast. Maybe use several small objects instead of one big one, use particles, animate the intensity of the emission, etc.
You can start by using only the temperature channel (disable the others) and see what different effects you can get just by varying the emission pattern and animation. You may not even need more than one channel for an effect - even in a high quality production.

Fire Simulation

Fire can be simulated in several ways. All you need is one or two fields as a basis to setup your fire shader, which is what you will most likely spend the most time with. You can even use the most straight forward density simulation and turn it into a nice flame only by using the right shading parameters. The density-based-flame example project shows how this works. See the next section for more details on shading fire.
There are several ways to get more control over your fire simulation. They are based on the Fuel channel. Objects can emit fuel that will burn if the temperature at a voxel is above the Ignition Temperature. When fuel burns, the air heats up and expands, as specified by the Expansion parameter. This is the essential control for explosions and large fire balls. Another effect of burning fuel is the Heat Creation and Density Creation. These parameters control how much is added to the temperature and density channels per unit of burnt fuel. Heat Creation is how a fire keeps itself burning, that is keeps the temperature above the ignition temperature after the initial ignition.
Another feature of Fuel is, that it may move slower than the rest of the fluid. This gives you some additional control over the shape of the flames. So does the Fuel Diffusion parameter, which will essentially blur the fuel field, letting fuel spread slowly into all directions regardless of the movement in the fluid.
The Fire channel provides an alternative channel to render fire. Fire values are large wherever fuel burns and cool off the farther away from the burning fuel they are. This creates a field that allows you to shade the flame based on the distance to the flame core. The next chapter will go into more detail on shading fire.

Shading Fire

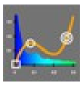
From a visual perspective, fire is essentially hot gas and soot particles that emit light. Most flames consist mostly of soot which is Carbon. Carbon is what is called a Black Body. A Black Body absorbs all light that hits it, hence it is black. However, when it's temperature rises to over about 600 Kelvin, it starts to emit light in a very characteristic way. This is where the familiar red/orange/yellow/white colors come from. Because fire mostly consists of soot particles, this is what dominates the color of flames.

Since soot particles emit the light of a flame, that means that the more soot particles there are, the brighter the flame will be. On the other hand, since Black Bodies also absorb light, more soot particles also make the flame more opaque. These two effects are exactly what is controlled by the Opacity group of the Fire Shader. This is what you use to control the shape of the flame.

To control the color you have two options. One is to specify a color gradient directly and the other is to use the physical model that describes the colors of a radiating Black Body. The later will give you an easy way to obtain realistic colors, while the former gives you full artistic freedom. When using the custom gradient, take care that you use a high dynamic range of colors (also make sure that the Clamp checkbox is not checked). Look at the intensity values of the default color gradient as an example. It actually has a constant orange color. Only the intensity runs from 0% to 2000%.

As mentioned above, you can shade fire using several different simulation channels. The density-based-flame example uses the temperature field to drive the color and the density channel for the opacity. While the fuel-based-flame shows pretty much the same using the Fuel and Fire channels. The Fuel channel gives you more freedom to simulate a reaction, but regarding shading it behaves the same. In these examples, the Mapping of the Opacity group creates the typical flame shape. A rising plume of fuel will only burn at the outer contour where there is enough oxygen. That means that most light is also emitted only from these areas. The Mapping function curve exploits the property of Density and Fire fields that they have their large values inside the plume/flame and smaller values outside. By creating a peak in the Mapping you specify where the surface of the flame should be that emits most of the light. See section 19 on the F-Curve editor for more details on how you can design the mapping curves.

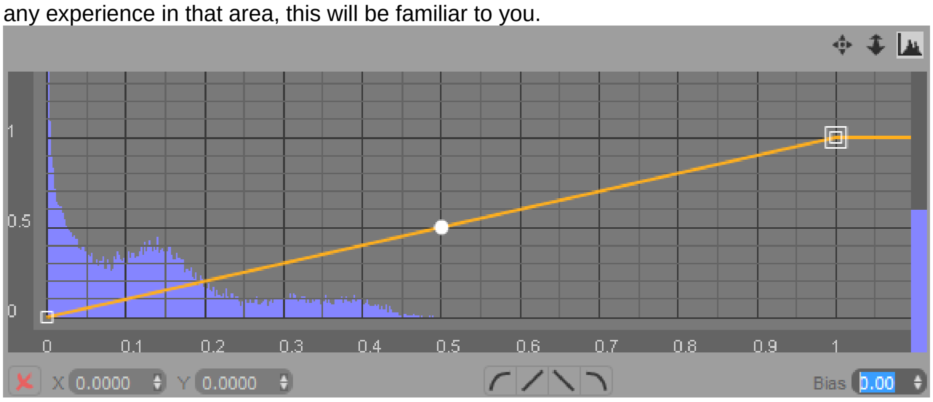
You can also use the Fire channel to drive the color or the opacity or both. You can even shade flames without using an opacity channel at all. Note that in this case you won't have alpha information to composite your flame later on, though. You still composite such a flame by using the brightness of the flame as a matte channel. This is justified by remembering that the brighter the flame is, the more soot particles there are and the more opaque the flame will be.



F-Curve Editor

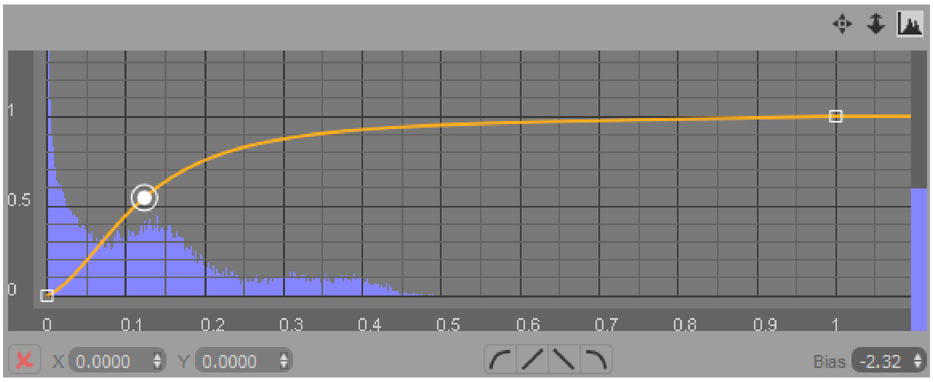
One of the most important aspects of volume shading is the choice of a transfer function. This function assigns a color and opacity to every input value. It is the shader's job to define such a transfer function. In order to do this, both shaders use an intensity mapping. While the way color is determined differs for smoke and fire, in both shaders the intensity mapping is basically the black-and-white version of the transfer function.

TurbulenceFD features a special purpose function curve (f-curve) editor that allows for convenient and accurate control of the transfer function by the artist. Working with this editor is very similar to post-processing images with color correction. If you have any experience in that area, this will be familiar to you.

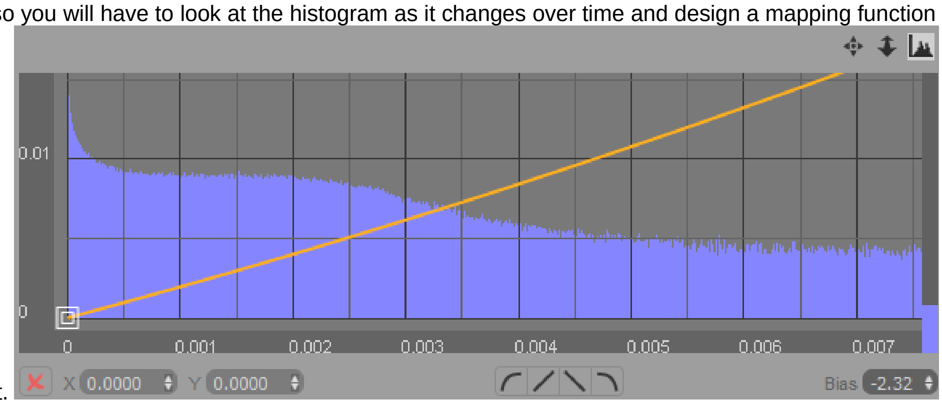


A linear mapping, as shown in the image above, makes color and opacity scale proportional to the input values. In many cases we want to cut off some parts or amplify others, though.

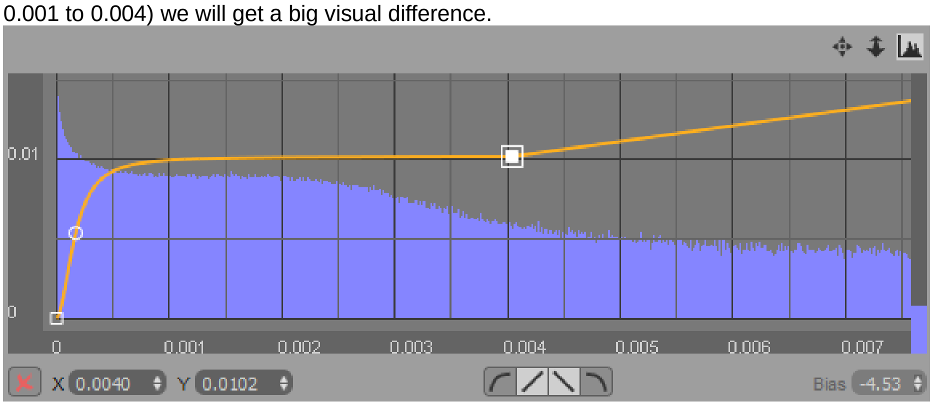
For example, for smoke shading we might want to have a smaller range of thickness (or opacity) that the smoke can take on. That is, the thickness should not fall off linearly with the density, but sustain a high value and then fall of quickly and the low end of the density range as shown in the image below.



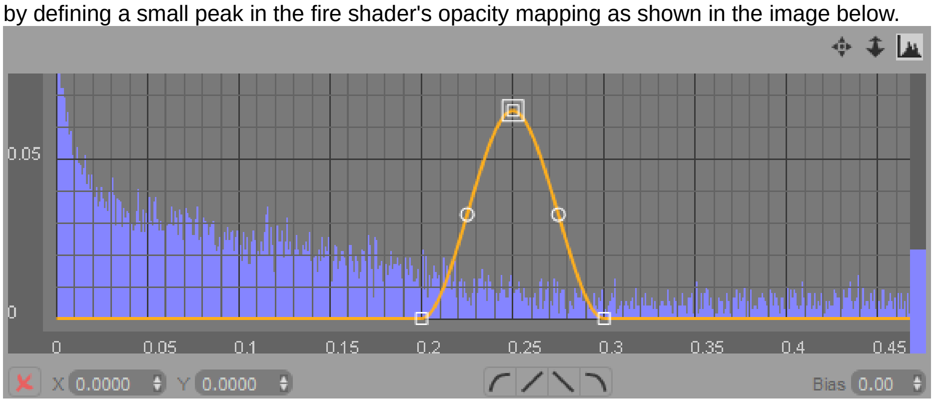
To assist with the design of the mapping function, the editor shows a histogram of the input values as a backdrop. For example, in the image above we can see that there are no input values above 0.5, so we should focus our mapping function on the values that are actually in the container. Of course, the value distribution will change from frame to frame depending on your simulation, so you will have to look at the histogram as it changes over time and design a mapping function accordingly or even animate it.



Often it is necessary to create very subtle changes to the mapping function to obtain a certain look. Note the value ranges on the X and Y axes in the image above. The editor has been zoomed in to show the fine details at the low end of the input value range. The histogram shows us that there are still significant changes at this scale and by increasing the intensity for this range (about 0.001 to 0.004) we will get a big visual difference.

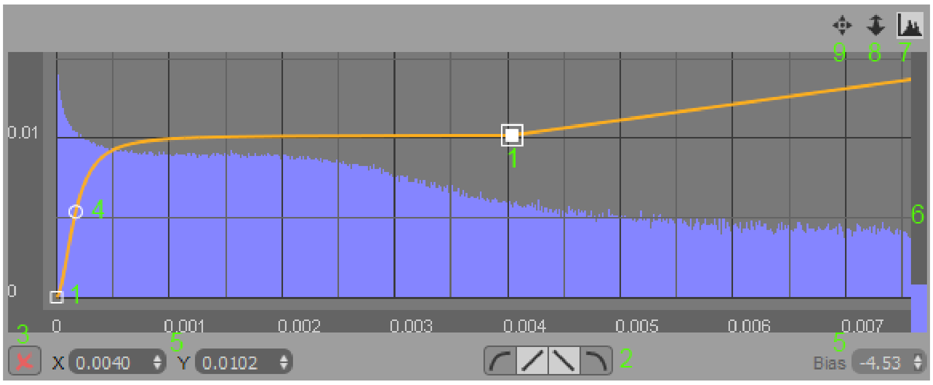


For fire shading, we usually want only a narrow band around the surface of the flame to be visible and luminous. We can do this by defining a small peak in the fire shader's opacity mapping as shown in the image below.



With different f-curve designs, we can achieve a wide range of looks. Make smoke contours sharper by suppressing low density values, fill the inside of a flame by extending the above peak to the right, etc.

The following image shows the various controls of the editor:



- select and drag knots (1) using the left mouse button
- add new knots using the right button
- select multiple knots by holding down shift while left-clicking and dragging the mouse to span a selection-rectangle
- move selected knots (1) by left-clicking on any of them and dragging them to their new position
- define smooth or linear tangents for the knots using the tangent buttons (2)
- delete selected knots by pressing the delete button on the toolbar (3) or the delete key
- bend a curve segment (change the bias) by left-clicking a segment handle (4) and moving the mouse horizontally
- change all these values using the value input boxes (5)
- adjust the scale of the histogram using the blue slider on the right (6)
- toggle the histogram on and off using the histogram button (7)
- use the mouse wheel or zoom button (8) to zoom horizontally or vertically when holding Shift at the same time
- use the middle mouse button or the translate button (9) to translate the view

Advanced

GPU Simulation

Fluid Simulation needs quite a bit of processing power. Mostly because there is a huge amount of data to be pushed around. This makes memory bandwidth the most important factor for simulation speed. Today's fastest memory interfaces are found in GPUs - about 10 times faster than those of CPUs. Coupled with the appropriate amount of parallel compute power, GPUs are the ideal type of processor for fluid simulation.
TurbulenceFD makes use of GPUs for it's simulation pipeline. Unlike with some GPU accelerated tools, this is not just a stripped down version of the CPU pipeline. All features are supported at the same quality. In fact, you can switch between CPU and GPU simulation on-the-fly (see Simulation Window). This is also what TurbulenceFD will do automatically, should it run out of GPU memory. It will then continue the simulation on the CPU.

Supported GPUs	Nvidia GPUs with Compute Capability 2.0 or newer, listed at http://developer.nvidia.com/cuda-gpus While TurbulenceFD works with less than 1GB of GPU memory, a GPU with 4GB or more memory is recommended. Please make sure to use the latest driver for your graphics card.
Tips	• When choosing a GPU, prefer the one with the most per-GPU memory • Ideally use two GPUs: one (possibly smaller one) as primary display GPU and one as a secondary GPU for simulation only. • When your system has only one GPU and you run large simulations, disable the viewport preview to speed up the simulation. • If you have a supported GPU but don't have anything to select but "Use CPUs", update your driver (see Supported GPUs above). • The larger the resolution the better the speedup (GPU vs. CPU) will be. At very low resolutions, the GPU sim may not be much faster.

Network Render

Note that network rendering is only available in the licensed version.
In order to render TFD scenes on a render farm, the plugin has to be installed on each render node.
Note that the simulation has always to be run on a single machine (usually your workstation) before nodes can render the result.

Particle Support

TFD integrates with CINEMA 4D's built-in particles as well as Thinking Particles. Particles can act as emitters, adding values to a fluid channel or velocities to the fluid velocity field. They can also be affected by the fluid velocity field, such that they move with the flow.
In order to use particles as emitters, add a TurbulenceFD Emitter tag to the Emitter object (built-in particles) or a Particle Geometry object (Thinking Particles). The settings in the Particle emission intensity tab allow you to modulate the emission values based on one of the particle properties like Age, Mass, etc.
All built-in particles are affected by the velocity of a fluid container if the containers Simulation/Velocity/Particle Velocity Scale parameter is above 0%.
Thinking Particles can be affected by any value or velocity in a fluid container via the two XPresso nodes that TFD provides. The GetFluidData node (found in System Operators > XPresso > TurbulenceFD) allows you to read values from any fluid channel at a given position. It has two mandatory input ports. The first is the Volume Object and the second is the position at which you want to read the values from. For each fluid channel there is an output port that will hold the corresponding value.
The PFluid node (found in System Operators > Thinking Particles > TurbulenceFD) is a modifier node, similar to PGravity, that moves particles along with the flow. Like the GetFluidData node it requires you to specify a Volume Object that will provide the velocity field of the flow. Send your particles to the Particle input port, for example with a PPass node. When the particles pass through the Volume Object, their trajectories will be affected by the fluid flow. Optionally you can use the Velocity Scale input port to adjust how strong effect on the particles' velocities will be.
See the projects temperature-particles.c4d and flame-thrower.c4d in the Examples folder for examples of particle setups.

Tweaking Performance

General tips:

- Store/Load the sim caches to/from the fastest drive you have (e.g. an SSD). Avoid network drives unless you're on a high-end network and file server (10GbE Isilon, etc.)
- Make sure your virus scanner does not scan .bcf files on read/write
- Close all other applications during sim/render

Simulation Performance:

- Don't run simulations that exceed your available memory
- Make sure you don't sim and/or cache channels you don't need for rendering
- Use only 1 or 2 sim steps unless you see artefacts
- Don't use obstacles unless they're essential to your effect.
- Don't use very high-poly objects as emitters. Use low-poly proxies instead.
- Don't use large numbers of octaves for noise on emitters, turbulence, etc. unless you're sure you need them
- Disable the preview(s) and/or timeline update during sim
- When using several emitter objects, parenting them under a null with one emitter tag is faster than adding a separate tag to each object.

To optimize render times, consider this:

- Avoid section of mapping curves that have values close to zero. This often happens when strongly bending a curve segment, such that one part of it seems to touch zero, but it actually doesn't.
- Use Sub-Grid Detail only of absolutely necessary. Use a low or no distance between Largest and Smallest Scale.
- Lower the Illumination Resolution until you see artifacts.
- For multiple scattering lower the Depth and Directional Resolution until you see artifacts.
- Increase the Step Size and Shadow Step Size until you see artifacts.
- For Motion Blur use a low number of Sub-Steps
- Make sure Adaptive Step Size is enabled

Reference

TurbulenceFD menu

Items in the Plugins/TurbulenceFD menu:

TurbulenceFD Container	Adds a new fluid container object to the scene.
Simulation Window	Opens the Simulation Window.
Start Simulation	Starts the simulation and opens the Simulation Window.
Continue Simulation	Continues the simulation and opens the Simulation Window. This item is disabled unless the currently selected simulation cache has a LastSimState.bcf file which was stored when the last simulation stopped.
Up-Res Simulation	Runs an up-res pass on the current simulation cache opens the Simulation Window.
Preview Fluid Container	Open the preview window that show a fast preview render of the fluid container.
TurbulenceFD Help	Open this manual in your default browser.
Check for Updates	Check online for updates of TurbulenceFD.
Purchase License	Opens the Jawset shop in your default browser.

Emitter Parameters

Emitter Active	Activate or deactivate the emitter tag entirely. This option can be used for convenience when working with multiple emitters.

General

Radius	If Fill Object is not checked, the values will be set in a region around the surface of the tagged object(s). Below is a comparison of two different thickness values for a sphere object.
Collision Object	If checked, the object will act as an obstacle to the flow. The fluid will have to flow around it. A moving collision object pushes the fluid around. There is a special requirement for the geometry to work as an obstacle. To sample the object geometry on the voxel grid, the simulation has to determine whether a voxel is inside the object and or outside of it. A simple plane does not work for that reason - it doesn't have an "inside". Instead, you would have to create a wall using a scaled box. All parts of the object have to be large enough to cover at least one voxel. Structures thinner than the voxel size will not be resolved properly on the voxel grid.
Fill Object	If checked, the tagged object(s) will be filled with the values specified in this tag. Else, the values will be set only in a region around the surface of the object(s). There is a special requirement for the geometry to work as a filled emitter. To sample the object geometry on the voxel grid, the simulation has to determine whether a voxel is inside the object and or outside of it. A simple plane does not work for that reason - it doesn't have an "inside". Instead, you would have to create a wall using a scaled box. All parts of the object have to be large enough to cover at least one voxel. Structures thinner than the voxel size will not be resolved properly on the voxel grid.
Restrict to polygon selections	You can drop polygon selections here to restrict the tag to these polygons. If the list is empty, the tag applies to the whole object. See CINEMA's documentation for details on working with selections.

Texture

Volume Texture	This CINEMA 4D shader allows to add texture to the emission. Usually the Noise shader is used. It will texture the volume that is covered by the band around the surface of the emitter. It does not use UVW coordinates or work with 2D surface textures. See CINEMA's documentation for details on working with shaders.
Surface Texture Channel	When set to any channel other than None, the right most texture tag on the emitter object (or any of it's children) is used to control the emission. The values from the selected texture channel will be used to scale the emitted fluid values. This allows you to "paint" where on the object fluid values are emitted. At the moment, this works only for UVW mapping of textures. Other spaces (like Object, World, Texture, etc.) don't work, yet.
Use Vertex Maps	If you emitter object (or any of it's children) have vertex weights set, check this option to use the weights to control the emission. You can scale the intensity of the emitter fluid values by the weight assigned to each vertex. This allows you to "paint" where on the object fluid values are emitted, similar to surface textures. See CIENMA 4D documentation for more information on how to use vertex maps.

Particle emission intensity

Particle property Intensity	When the emitter tag is used on a particle emitter or Particle Geometry object, you can modulate the emission intensity by any of the particle properties like Age, Mass, etc. This f-curve allows you to re-map the emission intensity based on the value of the property chosen from the drop-down box above.

Force

Velocity Weight	When using particles as emitters, the particles can drag the fluid along their trajectory as they move. The larger this value, the more the particles will affect the fluid velocity.

: Note that vice versa, you can also let the fluid drag along the particles by using the Particle Velocity Scale parameter (see Velocity).

Normal Force	The force that the object exerts on the fluid into direction of it's surface normal. Normal Force requires more simulation time than the Direction Force below. It should be preferred for planar emitters that have the same normal everywhere like a plane or disc.
Directional Force	Emit a force that points into the direction specified here. While the Normal Force may change it's direction across the emitter's surface, the Directional Force is not affected by the emitter's surface normals.
Pressure	Pressure can make the fluid expand or contract as it would in an explosion or black hole. Positive values cause expansion, negative values cause contraction.

Channels

Temp./Dens./Fuel /Burn Emission Mode	There are two ways how channel values (temperature, density, fuel) are added to the fluid **Add** the object will emit the value set below per second **Set** the channel values for the object will be held constant the value set below To understand the difference, think about temperature. Using Set mode, you specify the temperature the object has. Using Add mode, you specify by how much the object will heat up the fluid every second. In the later case, if the fluid wouldn't carry away the heat around the object (e.g. Buoyancy is 0), it will continue to heat up the fluid infinitely. While in Set mode the value stays constant.
Temp./Dens./Fuel /Burn	The value emitted per second (in Add mode) or to be set constant at the object (in Set mode).

Container Parameters

Container

Resolution

Voxel Size	Specifies the size of a voxel. Voxels are cubes. This value specifies their side length. When changing this value, the resolution will change accordingly.
Grid Size	specifies the size of the fluid container. The simulation will clip everything outside of this box. In most cases, the simulation will clip even more than that, trying to minimize the box that acutually needs to be simulated. See the Clip Below parameters of the fluid channel tabs for more details. The info field Max Memory Usage shows the memory that the simulation would use if the simulation used all the space in this container.
Grid Offset	There are two ways of moving the fluid container in space during simulation. You can move the container object like any other object in the scene or you can this Grid Offset parameter. Moving the container as an object will move the container and the fluid in it. If you're moving the container using the Grid Offset parameter, only the container will move. The fluid will stay in place or be clipped at the new container boundaries.
Max Memory Usage	This field shows the maximum resolution (in voxels) that the container can use as well as the amounts of memory that it would use in this case. How much the simulation will actually need depends on how the simulation grows over time. Here is an example: *386x649x395 99.0MV - CPU 7.9GB GPU 3.7GB UpRes 62.4GB - Cache/F: 1.9GB UpRes 14.9GB* The first section shows the maximum dimensions of the container in voxels and the total number of MegaVoxels or Million Voxels (MV). The middle section shows the amount of memory the simulation will at most require when run on the CPU, the GPU or in Up-Res mode. The last section shows the maximum size a single cache frame would have on disk without compression for a normal simulation and an Up-Res pass. When starting the simulation, TurbulenceFD will warn you if the available memory would not suffice would the simulation take up the whole container. You can ignore the warning if you know that your simulation will stay small enough or if you're keeping an eye on the simulation progress, so you can abort the simulation if necessary. When running simulations un-supervised, it's a good idea to make sure the container dimensions and simulation settings are chosen such that the avilable memory is sufficient. If the available memory is exceeded, the machine will most likely become unresponsive and you may have to reboot your system.

Cache

Just like high-quality rendering, fluid simulation is a computationally very intensive task. Moreso than rendering however, detailed voxel grids require a lot of memory. A 128x128x128 grid (2 MegaVoxel) basically corresponds to 128 bitmap images with 128x128 32bit (=4byte) pixels for each channel we cache. The velocity channel is actually a vector channel, so it uses 3 times as much space. That's 2MegaVoxel x 4byte x 3 = 24MB for each frame - only for the velocity. Every additional channel uses another 8MB per frame.

Each frame is written to a separate .bcf file in the cache directory. Each .bcf file contains all active channels (see Simulation parameters) and the velocity, if you enable it (see Cache Velocity below). The filenames contain the frame number, so you can easily identify every single frame, for example to continue the simulation from that frame at a later time (see General parameters above).

While all frames are written to disk, TurbulenceFD will only keep a small number of frames in working memory and dynamically load frames from disk as necessary for rendering and in-editor previewing.

Simulation Caches	specifies a base directory for all simulation caches of this TurbulenceFD object. In the list below, you can use the add/rename/remove buttons to manage the caches in the base directory. Select the cache in the list that will be used for simulation and rendering. Each cache is saved to a sub-directory of the base directory. If you start a simulation without specifying a base directory or cache, the default base directory from the TurbulenceFD preference tab will be used and a new cache will be created automatically. You can specify an alternative cache base path in the Alt. Base Path field. This is useful for render farms that use a mix of MacOS and Windows. Since the paths will differ between the

	operating systems. TFD will use the alternative base path if the main base path cannot be found or accessed.
Lock Cache	If checked the current cache directory is locked against accidental overwrite. In order to run a simulation on a locked cache, you have to uncheck this box first.
Compress Cache	If you are using very high resolution grids and disk space becomes an issue, you can enable compression. Note that compressing the cache also requires processing time which will slow down the simulation. However, especially when saving large caches to a network drive, it can balance out with the time it takes to transmit the data over the network, because compression can reduce the file size up to 60%.
Cache Temp./Dens./Fuel /Burn	Select whether to store either of these fluid channels to disk. You only need to store the channels you use as shader inputs. You can save memory by not caching channels that are not shaded. However, if you want to continue a simulation from any previously simulated frame without re-simulating al prior frames, all active channels need to be cached.
Cache Velocity	Caching the velocity of the fluid allows you to use Velocity Displacement during rendering, continue or restart simulations from any frame of the cache and move particles through the fluid. However, it requires considerable amounts of disk space. Because velocity is a 3-dimensional vector, caching the velocity takes 3 times as much memory as for example the temperature field.
Cache Collision	Cache the collision field used by the simulation. You can use the viewport preview with this channel to check how voxelization of the collision objects has turned out. See the Collision Object parameter in the emitter's General tab for more information about collision object voxelization. The collision field is a partial signed distance field. For values inside collision objects, each voxel value represents the negative distance to the closest point on the object's surface. Voxels that are not inside any collision object have a very large positive value.
Current Frame Info	Shows details about the simulation cache for the current frame.

Simulation

Simulate while rendering	If this box is checked, the simulation will be done frame by frame during the rendering of the scene to the picture viewer. This is useful for batch simulation and rendering, but can also be used as a way of previewing the simulation while it runs at a higher quality as the other previewing options provide.

General

Solver

Frame Sub-Steps Limit	If your simulation has high velocities and you're seeing noise artefacts in the flow, allow the simulation to take more sub-frame time slices if necessary by increasing this value. This will allow the simulation to resolve the fluid motion better. The simulation will also take more time, though.
Pressure Iteration Limit	Wherever velocity deforms the fluid, pressure is build up. It is a fundamental part of the simulation to equalize this pressure. For simulations that have high velocities and/or collision objects or closed container boundaries, it is recommended to use values between 2 and 10. Values larger than 10 should rarely be necessary. For simulations that don't have collisions and mostly moderate velocities, the simulation time can be kept lower by using the minimum of one iteration.
Adaptive Container	If enabled, TFD simulates the smallest possible part of the container in order to save time and memory. The Clip Below parameters in the Velocity, Temperature, Density, Fuel and Burn tabs control when TFD may shrink the container. Unchecking this box is only useful for benchmarking and testing purposes.

Advection

Velocity/Channel Advection	The advection accuracy affects the sharpness of the details in the fluid. **1st order** produces a somewhat blurrier result where vortices get smoothed away quicker, but it's also faster and uses less memory. **2nd order** produces sharper details, keeps vortices alive longer but uses more time and memory.
Adaptive Tracer	When advecting the fluid, the simulation traces a trajectory through the velocity field for each voxel. The fluid moves along these trajectories. The default tracer approximates the trajectories with a single line segment. The Adaptive Tracer uses many line segments, such that none of them are longer than the voxel size.
Cubic Interpolation	Moving the channels along the flow requires repeated interpolation. This interpolation slightly blurs the channel each time. Using cubic interpolation reduces this blur at the cost of somewhat higher simulation times.

Misc

Use less memory but more time	Since fluid simulation can be very memory intensive, you are likely to exhaust your system's memory with large simulations. In order to allow you to run larger simulations on machines with less memory, checking this option will let TurbulenceFD use less internal caches which results in less memory being used but more simulation time being required. Render times are not affected.
Collision Objects Enlarge Container	The adaptive container will be enlarged to contain all collision objects by default. This may not be necessary, especially if the objects aren't moving. In such a setup, you can uncheck this option to keep the container smaller.
Smooth Collision Surface Rendering	When rendering solid obstacles emerged in the fluid, extrapolation will avoid artefacts on the surface of the obstacle. If you have invisible or transparent obstacles, disable this option.

Closed container boundaries	By default the sides of the fluid container are open. The fluid can just move out of the container and disappear. If you want to simulate a closed or partially closed space, you can select the sides of the container that will be closed. The checkboxes are labeled according to the half-spaces of the coordinate system. That is, +X is the side of the container in the positive X direction and so on. For example, a ground explosion can be simulated in a container with the -Y side closed to avoid having fluid leave the container through the ground.

Up-Res'ing

When running a simulation at a higher resolution (lower voxel size) it will take longer to simulate and may produce different results. Not only will there be more detail, but even the coarse shape of the fluid motion may be different. While the differences won't be huge, they may still be undesirable after you got the overall look right in a lo-res simulation.

Up-res'ing is a simulation post-process that allows you to increase the resolution of a simulation cache without running a full simulation at the higher resolution. The result will look pretty much exactly like the lo-res version but with additional small detail.

In order to be able to up-res an existing simulation cache, it needs velocity. Make sure you enable velocity caching when running the lo-res simulation. When up-res'ing a cache you don't need to cache the velocity again unless you need it for rendering (for sub-grid detail or motion blur).

Note that you can also change the Max. Simulation Steps for the up-res post-process as well as the channel parameters (i.e. the parameters in Simulation/Temperature, Density, Fuel and Burn).

Upres Scale	When running an Up-Res'ing post-process by clicking the Up-Res button, the resolution of an existing simulation will be increased. This parameter specifies how much larger the resolution will be.
Fine Turb. Intensity	The Up-Res'ing post-process also adds small scale turbulence to the fluid that will create additional detail. This parameter specifies the strength of this small scale turbulence.
Fine Turb. Small Power	Specifies how stong small turbulence detail is compared to the next larger size. This works just like the Simulation/Turbulence/Small Power parameter but it's applied only for the additional detail added during Up-Res'ing.

Timing

Turbulence uses the frame-rate (FPS) set in the project settings (Edit > Project Settings). Take care when using a different frame-rate for the renderer (Render Settings > Output > Frame Rate). The fluid will not be in sync with the rest of the animation when using different frame-rates.

Time Scale	This value controls how fast time will pass for the simulation. At 1.0, the simulation will run at the same speed as the animation. Values below 1.0 slow the simulation down, values above 1.0 speed it up. You can for example animate this value down to 0.0 to create bullet-time-like animations.
Frame Range	specifies what part of the time-line to simulate. The settings work just like those in CINEMA's Render Settings in the Output section. All Frames simulates the whole timeline. Preview Range simulates the range selected in the editor's preview slider. Manual allows you to enter different start and end frames for the simulation.
From	Specifies the start frame for the simulation.
To	Specifies the end frame for the simulation.
Start Clean Simulation	Clear all channels before starting the simulation at the frame specified above.
Load Simulation State From File	If Start Clean Simulation is unchecked, channels will be initialized from the file specified here. The file may be any .bcf file from a TurbulenceFD cache directory. Channels that are not available in the .bcf file will be cleared as if Start Clean Simulation was checked. See the parameters in the Cache tab for information about how to control which channels are saved to the disk cache.
Continue Simulation From File	If checked, the next simulation run will load the state from the file specified above and jump to the frame following the one cached to the file. Use this to continue a simulation at any cached frame, possibly after changing some parameters.

Velocity Field

Velocity

Clip Below	Specifies the minimum velocity in voxels per second that will prevent the container from adapting it's size. Velocities below that will be cut off when the container adapts it's size. The default value is chosen such that it works with most simulations, including explosions which need lower values than other simulations. However, at the default value the container may be lept larger than necessary. If you want to speed up the simulation by letting the container adapt more tightly to the channels, increase this threshold. A value of e.g. 1000 will effectively ignore the velocity when adapting the container. See also the threshold parameters of the channels below.
Damp Velocity	Specifies the percentage by which the velocity is reduced in each frame. This simulates drag or friction in the air. This is for example useful for explosions where you may want the fluid to leave the source quickly but then form a cloud that moves slower.
Particle Velocity Scale	If this value is larger than 0%, particles will be moved by the fluid. In this case, the fluid container works as a particle modifier. When using particles, 100% would be the most common value here.

Wind

This wind operator allows you to define a global wind that is blowing in the fluid container. It can be used to add wind as it would appear in an outdoor scene. To create small, local winds like from an exhaust pipe, use the Normal Force parameter on an emitter object (e.g. a plane).

Direction	Specifies the direction of a global wind in the container.
Speed	Specifies the speed of the global wind in the container.

Vorticity

Amplifies existing curls to keep them alive longer. When using strong amplification, the curls will get stronger and stronger. After some time the flow may become very noisy. To avoid that, the Intensity Channel and Mapping allow you to restrict the effect of Vorticity to a region that you control using another fluid channel.

Vorticity	Sets the strength of amplification for small curls.
Intensity Channel	You can control the vorticity in space using any of the fluid channels. Use this to amplify the curls only where the temperature is high enough, for example.
Intensity Mapping	For improved control over the intensities, the values of the Intensity Channel are re-mapped by this f-curve before scaling the vorticity.

Turbulence

Adds random velocity noise to the container, making it swirl and a chaotic way as you would expect from turbulent fire or similar. It is driven by a procedural noise function that changes over time. The following parameters control the intensity, scale, etc. of this texture.

Turb. Intensity	Specifies the strength of the turbulence.
Intensity Channel	You can control the turbulence intensity in space using any of the fluid channels. Use this to add turbulence only where the temperature is high enough, for example.
Intensity Mapping	For improved control over the intensities, the values of the Intensity Channel are re-mapped by this f-curve before scaling the turbulence intensity.
Smallest Size	Specifies the size of the smallest curls added to the fluid.
Largest Size	Specifies the size of the largest curls added to the fluid.
Small Power	Specifies how strong small curls are with respect to the next larger ones. A small power of 1.0 will make curls of all sizes equally strong. A small power of 0.5 will make small curls half as strong as curls of twice the size.
Period	The turbulence field changes over time. This value specifies the number of frames between it's phases.

Channels

Temperature

Active	Check this to enable simulation and caching of the temperature channel.
Clip Below	If the Adaptive Container option is enabled, this value defines the minimum temperature that will cause the container will consider non-empty. If this value is zero, no voxels containing temperature will be lost. However, often shading settings are such that low values aren't visible anyway. In this case, increasing this threshold allows the sim to keep the container smaller.
Temp. Diffusion	Air mixes even if it does not move on the large scale. This effect is called Brownian Motion. Diffusion accounts for this effect by basically blurring the temperature. As a result of high diffusion, the buoyancy force becomes smoother.
Cooling	Specifies the percentage by which the temperature is cooled down in every frame. Cooling works like exponential decay in the Burn channel.
Half-life	Instead of specifying the cooling intensity as percentage per frame, you can specify the time it takes for the temperature to cool down to half it's value. This may be more intuitive in many situations. Note that "no cooling" would mean infinite half-life, which CINEMA 4D cannot display in a value spinbox. In this case the half-life will display 0F, just like it does if the cooling is 100%. You will have to use the percentage value to distinguish these two cases.
Buoyancy	Buoyancy is the force that makes warm air rise and cold air sink. This value specifies the force per unit of temperature exerted on a particle.
Buoyancy Direction	specifies the direction of the buoyancy force.

Density

Active	Check this to enable simulation and caching of the density channel.
Clip Below	If the Adaptive Container option is enabled, this value defines the minimum temperature that will cause the container will consider non-empty. If this value is zero, no voxels containing temperature will be lost. However, often shading settings are such that low values aren't visible anyway. In this case, increasing this threshold allows the sim to keep the container smaller.
Dens. Diffusion	Air mixes even if it does not move on the large scale. This effect is called Brownian Motion. Diffusion accounts for this effect by basically blurring the density.
Dissipation	Specifies the percentage by which density is dissipated in every frame. Dissipation works like exponential decay in the Burn channel.
Half-life	Instead of specifying the dissipation intensity as percentage per frame, you can specify the time it takes for the density to dissipate to half it's value. This may be more intuitive in many situations. Note that "no cooling" would mean infinite half-life, which CINEMA 4D cannot display in a value spinbox. In this case the half-life will display 0F, just like it does if the cooling is 100%. You will have to use the percentage value to distinguish these two cases.
Gravity	The more dense the fluid is, the more mass it has. Therefore, it it gets affected by gravity force. This parameter specifies the strength of that force.
Gravity Direction	specifies the direction of the gravity force.

Fuel

Fuel works like a cloud of combustible gas like propane or gasoline mist. When it burns, it...

- emits Burn which then represents the burning flame
- expands the cloud (i.e. it emits pressure)
- emits heat (see Temp. Emission)
- emits soot (see Density Emission)
- reduces the amount of fuel left

Fuel Masking allows you to control where fuel burns. The Fuel Mask is a cloud taken from any other channel, smoothed and re-mapped. Only within that cloud the fuel will burn. This allows you to control ignition and suffocation of burning fuel.

Active	Check this to enable simulation and caching of the fuel channel.
Clip Below	If the Adaptive Container option is enabled, this value defines the minimum temperature that will cause the container will consider non-empty. If this value is zero, no voxels containing temperature will be lost. However, often shading settings are such that low values aren't visible anyway. In this case, increasing this threshold allows the sim to keep the container smaller.
Fuel Diffusion	Specifies how fast fuel diffuses (see Temperature/Diffusion above for more details).
Burn Rate	Specifies the amount of the fuel that is burnt in every frame. The faster it burns, the more Burn and Temperature will be created. Burn Rate works like linear decay in the Burn channel
Fuel Mask Channel	If a channel is selected, fuel masking will be active. See the simple-ignition example for a setup using the Burn channel.
Fuel Mask Smoothing	Before using the selected channel as a fuel mask, it can be smoothed or blurred. This will also make the cloud a bit larger, thereby propagating the ignition front like you would expect it to.
Fuel Mask Mapping	Re-mapping the smoothed fuel mask gives you control over the shape of the fuel mask and how steep the ignition front is. A flat increase will let the burn start slowly the larger the fuel mask values become. A steep increase will makwe the fuel burn at the full burn rate as soon as it touches the fuel mask.
Expansion	A factor that specifies how much the gas expands per burnt voxel of fuel.
Temp. Emission	Specifies how much heat is generated per unit of burnt fuel.
Density Emission	Specifies how much density is generated per unit of burnt fuel.

Burn

Active	Check this to enable simulation and caching of the burn channel.
Clip Below	If the Adaptive Container option is enabled, this value defines the minimum temperature that will cause the container will consider non-empty. If this value is zero, no voxels containing temperature will be lost. However, often shading settings are such that low values aren't visible anyway. In this case, increasing this threshold allows the sim to keep the container smaller.
Burn Diffusion	Specifies how fast Burn diffuses (see Temperature/Diffusion above for more details).
Decay Mode	Decaying a channel can be done by a constant amount or by a percentage of the current value. The Decay Mode specifies which will be used. **Linear**: Decay the burn channel by subtracting a constant value in each frame. If you emit 1.0 into the burn channel, setting Decay to 0.1 in this mode will let the emission live for 10 frames before it drops to zero. This mode works well for flames that have a sharp contour **Exponential**: Decay the burn channel by subtracting a percentage of the current value (of a voxel) in each frame. As the current value becomes smaller and smaller, so does the value that is subtracted. Therefore the emission never drops back to zero - only very close. And the speed at which is drops will get slower and slower. This is the type of decay that temperature and density have. It models the way dust slowly dissipates in air. The decision of which mode to use is mostly driven by your shading settings. To get an idea of how the modes differ, compare the histograms in the shader's mapping curve editors. The exponentially decayed channel will have lots of low values and less high values, because the decay slows down the smaller the values get. The linear decay more uniform distribution of high

	to low values. This makes shading sharp contours easier with the linear mode. If you want the channel values to live long and build up a more smoke-like cloud, the exponential decay works better.
Decay	In linear mode, Decay specifies the constant value that will be subtracted from the current channel value in each frame. In exponential mode, it's the percentage of the current value that will be subtracted.
Half-life	Instead of specifying the decay intensity as percentage per frame, you can specify the time it takes for the channel value to decay to half it's value. This may be more intuitive in many situations. For example if Half-Life is one frame, the channel value drops to 50% within one frame. That corresponds to a Decay of 50%.

Viewport Preview

The viewport preview shows a fluid channel using one of the shaders or a generic color scale for a more analytic result.

Show Bounding Boxes	Check to display the bounding boxes. The clipping box is displayed in white, the simulated region-of-interest is diplayed in green.
Show Grid	Check in order to display the container's grid in the viewport. This allows you to visualize the resolution of the fluid container.
Channel	Select the fluid channel that you want displayed in the viewport.
Shader	Select the shader you want applied to the viewport preview. If this selection is None, the generic color scale and range fitting is used to shade the values. Note that for performance reasons some of the shading features are not applied to the viewport preview. **The preview is not applying** • Smoothing • Separate Opacity • Sub-Grid Detail • Illumination • Motion Blur • Distance/Opacity Mapping
Use Shader Color	If checked, the shader's color settings are used for the preview. Else, only it's mapping is used along with the generic color scale.
Preview Mode	Full Slicing mode renders the whole 3D volume semi-transparent with opacity and color depending on the channel values. Single Slice mode renders a 2D slice of the volume and allows for close inspection of the values.
Interpolate Voxels	Uncheck this if you want to see the separate voxels more clearly.
Slice Orientation	Available in Single Slice Mode. Specifies the orientation of the slice for perspective viewports. Orthogonal viewports will always display the slice oriented such that it is visible and ignore this parameter.
Slice Position	Available in Single Slice Mode. Use this slider to move the slice along it's normal direction.
Slices per Voxel	Available in Full Slicing Mode. The higher this value is, the smoother the preview will look. Values above 2 may not improve the result significantly because of the limited color resolution of the OpenGL framebuffer.
Opacity	Available in Full Slicing Mode. Set the overall opacity of the preview to high values in order to amplify the faint parts of the fluid. Bring it down to smaller values in order to see the inner core of the fluid cloud.
Fit Range	Find the minimum and maximum value in the current grid and set the display range accordingly. While the Density and Fuel channels only have values from 0 to 1, the other channels have larger value ranges. For these channels this function can be used to fit the displayed range to the values that are actually present in the current frame. This is also a quick way to check how strong velocities or how high temperatures are.
Auto-Fit	If checked, the range will be fit to the current frame automatically.
Range Start	The start value for the displayed range. All values in the channel that are below or equal to the Range Start will be displayed with the color at the low end of the gradient.
Range End	The end value for the displayed range. All values in the channel that are above or equal to

	the Range End will be displayed with the color at the high end of the gradient.
Display Color	This gradient defines the color and opacity that will be used for the viewport preview.

Rendering

Note	In CINEMA 4D R12 and newer it is **highly** recommended to use Linear Workflow with TurbulenceFD. If this isn't your default already, activate Linear Workflow in the Project Settings.

General

Frame Offset	This offset is added to the current frame number to determine the frame used for rendering. You can use this to create instances of the container that render with a different frame offset each.
Frame Step	By default the render sequence will be the same as the simulation sequence. If you want to render a simulation in backward order, set this value to -1 and the Frame Offset to the end of the frame range. You can also skip frames by using values larger than 1 (or smaller than -1). Using a values of 0 allows you to freeze a fluid frame while animating the rest of the scene.
Step Size	Specifies the step size as a percentage of a voxel that is used to sample the fluid container. Large values make renders faster but can introduce noise. Smaller values reduce noise but need more time. You should reduce this value only if you see noise artifacts in your renders to avoid increasing the render times unnecessarily.
Shadow Step Size	Shadow rays are usually more robust wrt. noise artifacts. Setting a higher step size for these rays allows you to save render time. This is particularly effective for lit smoke renders.
Interpolation	Specifies the way values are interpolated from the simulation grids. Fast can be somewhat blurry and have banding artifacts. Smooth is the blurriest but avoids banding completely. Sharp avoids blurriness but can have banding artifacts. Note that you can always reduce or even remove blurriness by using steep Mapping curves in the shader settings, even when using Smooth interpolation. Also note that the illumination method (see below) can be the reason for banding as well.
Use Distance/Opacity Mapping	Enable the mapping of distance from camera to fluid opacity. See below for details.
Distance/Opacity Mapping	Maps the distance from the camera to an opacity for fluid rendering. This can be used when flying the camera through a cloud. Voxels that are very close to the camera may occlude too much of the cloud or they may simply be too big and look blobby. You can fade away their opacity using this mapping

Smoke Shader

Mapping

Channel	Select the simulation channel that you want the shader to use as input.
Smoothing	Blur the field before using it as shader input.
Mapping	This function curve (f-curve) allows you to map the input values to an intensity curve before they are used to determine the smoke's color and opacity. A linear 0 to 1 gradient will leave the values unchanged. To edit the Mapping, click on the curve to open the f-curve editor. See the section on using the f-curve editor for details.
Separate Opacity	These parameters let you use a separate input channel for smoke opacity. If you use it, the first channel, smoothing and mapping parameters will only control the smoke color.

Color & Opacity

Thickness	Low values can be used for steam or vapor, high values for thick smoke. Unlike rigid bodies, smoke particles usually do not fill space up completely. Therefore, smoke is always partially transparent. However, the larger the distance that light travels through smoke and the thicker it is, the higher the probability that light rays hit a smoke particle and get scattered or absorbed. That is, while you may be able to look through 1 inch of smoke at thickness 1, 10 inches may be virtually opaque. At thickness 10 however, you may not even be able to look through 1 inch of smoke.
Brightness	The brighter the smoke, the more it will reflect incoming light. You can also adjust the smoke's brightness using the Smoke Color below, but this parameter allows for HDR brightness ranges and is easier to adjust - especially if you have a gradient set for the Smoke Color.
Smoke Color	Defines the smoke color gradient. It assigns colors to re-mapped channel values. The default is to use a single color for all input values (0 to 1) and let only opacity and illumination shade the smoke.

Sub-Grid Detail

There are two type of render-time sub-grid detail. Both work at render time without requiring a re-simulation.

- Velocity displacement deforms the fluid during rendering and can produce very sharp details.
- Sub-Grid Noise adds detail to the fluid that

Velocity Displacement works only if a Velocity Cache is available (see Container). Sub-Grid noise works without a Velocity Cache, but for smooth animation you should also have the Velocity Cache available.
Both options increase the render time.

Velocity Displacement	Velocity Displacement warps the field forward or backward in time, depending on the value you set. Rendering times will be higher, so this is disabled in fast preview mode. Low values can add more detail while introducing very little noticeable deformation. Higher values can change the appearance of the field very much, but produce interesting effects. Note that you will have to enable Cache Velocity in the Simulation/Cache tab in order to be able to use Velocity Displacement. 
Noise Intensity	Specifies how strong the noise will affect the shaded fluid.
Smallest Size	Specifies the size of the smallest noise.
Largest Size	Specifies the size of the largest noise. Note that the larger the difference between Smallest and Largest scale, the more render time will ne needed.
Small Power	Specifies how strong small detail is with respect to the next larger ones. A small power of 1.0 will make curls of all sizes equally strong. A small power of 0.5 will make small curls half as strong as curls of twice the size.
Period	The turbulence field changes over time. This value specifies the number of frames between it's phases.

Illumination

<table>
<tr><td>Illumination</td><td>Illuminating smoke is the most time consuming part of a rendering gaseous fluids. TurbulenceFD offers several illumination modes with different speed and quality to choose from. In the list, the speed decreases (first mode is fastest, last mode is slowest) and quality increases (first mode has lowest quality, last mode has best quality).
Note that the default Fast mode may be all you need in many scenes even at high quality requirements. However, the thicker the smoke gets, the more likely you're going to see stairstep artifacts. In these situations, choose the fastest mode that delivers the quality you need.
<table><tr><td>None</td><td>No illumination at all</td></tr><tr><td>Fast</td><td>Low Quality</td></tr><tr><td>Smooth</td><td>Medium Quality</td></tr><tr><td>Optimal</td><td>Slowest, Lowest Memory Usage</td></tr></table>Both Corrected modes use more memory the more light sources illuminate the fluid container.</td></tr>
<tr><td>Scattering Anisotropy</td><td>When light hits a gas it either gets absorbed or bounces off into a different direction. Different types of gases have different preferences of directions into which they scatter light. Water vapor for example tends to scatter light in forward direction. That means that most of the light that hits the vapor does not change it's direction too much and continues to travel forward. As a result, the vapor will be brighter when the camera is on the other side of the vapor cloud as the light source.
The Scattering Anisotropy parameter specifies the preferred direction of light scattering. A positive value means forward scattering, a negative value means backward scattering.
For example, a value of 1.0 means that the light does not change it's direction at all. You will only see lit smoke if the camera is looking at the light source at the exact angle of the light. Vice versa, a value of -1.0 means that the camera has to look into the same direction as the light to catch any light scattered off the smoke. All light is scattered in exactly the opposite of it's incoming direction.
The neutral value of 0.0 means that light gets scattered into all directions equally.
In practice you would usually only use small values like 0.2 to get brighter backlit smoke for example.</td></tr>
<tr><td>Illumination Resolution</td><td>Illuminating smoke takes most of the render time. You might want to reduce the resolution at which the illumination is computed to save render time.
On the other hand, if you have extremely thick smoke you may see grid artifacts unless you increase the illumination resolution.</td></tr>
<tr><td>Multiple Scattering</td><td>Enabling Multiple Scattering allows for two effects.
It works like Global Illumination for smoke. That is, it computes additional bounces of light inside the smoke, effectively making it brighter. This is the preferred way of creating ambient light. Using the ambient light parameter on the light source will create a low quality approximation of this effect.
It allows for the Fire Shader to illuminate the smoke</td></tr>
<tr><td>Max. Depth</td><td>The higher the Dpeth value, the more bounces will be computed. More bounces will make the smoke brighter but also cost more render time.
You can create a cheap approximation of a higher Depth value by lowering the Falloff (see below) instead.</td></tr>
<tr><td>Directional Resolution</td><td>Increase this value if you see ray artefacts in your smoke. A higher Directional Resolution will increase the render time.</td></tr>
<tr><td>Falloff</td><td>The higher this value, the faster the light will fall off as it travels through smoke. You can get brighter smoke by using a lower value here.</td></tr>
<tr><td>Light Brightness</td><td>This allows you to separately adjust how much external lights contribute to Mutiple Scattering.</td></tr>
<tr><td>Fire Brightness</td><td>This allows you to separately adjust how much brightness the Fire Shader contributes to Mutiple Scattering.</td></tr>
</table>

Fire Shader

Mapping

Channel	Select the simulation channel that you want the shader to use as input.
Fire Smoothing	Blur the field before using it as shader input.
Clear Smoke Above	You may want to remove all smoke from the region where fire is shaded to allow for a clearer flame. This value specifies the mapping intensity above which fire will erase smoke for shading.
Mapping	This function curve (f-curve) allows you to map the input values to an intensity curve before they are used to determine the fire color and opacity. A linear 0 to 1 gradient will leave the values unchanged. To edit the Mapping, click on the curve to open the f-curve editor. See the section on using the f-curve editor for details.
Separate Opacity	These parameters let you use a separate input channel for fire opacity. If used, the above input, smoothing and mapping parameters will only affect the fire color.

Color & Opacity

Opacity	Specifies the opacity of fire. Fire opacity also affects the brightness of the flame. The more glowing particles there are, the more they will obscure the background and the more light will be emitted.
Color Mode	Selects the way the color gradient is defined. Manual Color allows you to specifiy a color gradient directly, while Black Body Color will use a physical model to compute the color.
Fire Color	Available in Manual Color mode. If Manual Color Mode is selected, you specify the color gradient here. Make sure that you use a high dynamic range of colors (intensities larger than 100%) and that the Clamp checkbox is not checked.
Luminance	Available in Manual Color mode. Defines the overall color intensity in Manual Mode.
Edit	Available in Black Body mode. Copy the color gradient from Black Body Mode to the gradient editor in Manual Mode.
Low Temp.	Available in Black Body mode. In Black Body Color Mode, this value gives the color temperature corresponding to an input value of 0 from the simulation channel. It's most intuitive to observe the gradient above to see this parameter's effect.
High Temp.	Available in Black Body mode. In Black Body Color Mode, this value gives the color temperature corresponding to an input value of 1 from the simulation channel. It's most intuitive to observe the gradient above to see this parameter's effect.
White Point	Available in Black Body mode. Fire has an enormous dynamic range. The white point is used to map it to a lower dynamic range used for you renders. It's most intuitive to observe the gradient above to see this parameter's effect.
Damping	Available in Black Body mode. Instead or in addition to adjusting the white point, you can damp the dynamic range using this parameter. The higher the value the darker the colors will be.
Red/Green/Blue	Available in Black Body mode. The Black Body color model is an idealisation of carbon based light emission. In real fires plenty of other chemicals are burnt or created during the reaction. These controls let you change the tint of the fire color to allow for a broader spectrum of colors. For example, you may want to add more red (or remove green and blue instead) to get closer to the look of oilier fires.

Sub-grid Detail

There are two type of render-time sub-grid detail. Both work at render time without requiring a re-simulation.

- Velocity displacement deforms the fluid during rendering and can produce very sharp details.
- Sub-Grid Noise adds detail to the fluid that

Velocity Displacement works only if a Velocity Cache is available (see  Container). Sub-Grid noise works without a Velocity Cache, but for smooth animation you should also have the Velocity Cache available.
Both options increase the render time.

Velocity Displacement	Velocity Displacement warps the field forward or backward in time, depending on the value you set. Rendering times will be higher, so this is disabled in fast preview mode. Low values can add more detail while introducing very little noticeable deformation. Higher values can change the appearance of the field very much, but produce interesting effects. Note that you will have to enable Cache Velocity in the Simulation/Cache tab in order to be able to use Velocity Displacement. 
Noise Intensity	Specifies how strong the noise will affect the shaded fluid.
Smallest Size	Specifies the size of the smallest noise.
Largest Size	Specifies the size of the largest noise. Note that the larger the difference between Smallest and Largest scale, the more render time will ne needed.
Small Power	Specifies how strong small detail is with respect to the next larger ones. A small power of 1.0 will make curls of all sizes equally strong. A small power of 0.5 will make small curls half as strong as curls of twice the size.
Period	The turbulence field changes over time. This value specifies the number of frames between it's phases.

Simulation Window

Frame	The frame number that is currently being simulated. All frames before that have already been saved to the cached and can be rendered in the editor or to the picture viewer.
Elapsed	Time that has elapsed since this simulation run started.
Estimated	An estimate of the time remaining to finish the simulation run based on the elapsed time and the remaining number of frames. It does not consider the velocities of the fluid, which have a large impact on simulation time. Therefore, if velocities vary a lot during the course of a simulation, this estimate may change a lot.
Grid	Shows the current resolution of the adaptive container.
Workspace	The amount of working memory occupied by the simulation core for processing.
Disk cache	The size of the simulation cache on disk.
Use CPUs/GPU	If you have supported GPUs, this drop-down box will let you select which one to use. See GPU Simulation for more details about fluid simulation on GPUs. If "Use CPUs" is selected, all available CPU cores will be used instead.
Render when finished	Let's TurbulenceFD automatically trigger a render after the simulation has finished. This is identical to choosing "Render to Picture Viewer" from CINEMA's Render menu. Remember to setup the render settings correctly before leaving the workstation. Also, make sure that the destination file(s) do not already exist. Cinema will pop up a dialog asking whether to overwrite the existing files, waiting for your answer instead of rendering.
Update Editor Time-line	Update the time-line of the editor as the simulation progresses. This can be used to watch the simulation run using the fast slice-of-volume display. Note that the editor preview will also slow down the simulation somewhat.
Start	Simulate the fluids in this container from the beginning of the selected frame range (see Simulation/Timing/Frame Range).
Stop	Stop the simulation. It will finish simulating the current frame in order to allow continuing the simulation later. The button will change to "Abort". Click it again if you want to abort immediately and don't need to continue the simulation later.
Continue	Continue a simulation after the last simulation state. This last state is always saved when the simulation stops. Be it because you click the Stop button or because all frames from the selected range have been simulated. Therefore this also allows you to add more frames at the end of the range.
Up-Res	Post-process a simulation to increase the resolution and add more detail. A new cache directory will be created using the name of the current cache with the word "upres" appended. You can switch back to the previous cache using the Container/Simulation Caches list. See the Simulation/Up-Res'ing tab for more options. This post-process is faster than simulating at the high resolution in the first place. It will also retain the shape of the lores simulation and only add small detail that couldn't be resolved on the lores grid before.

Render Settings

Settings in the TurbulenceFD Renderer tab of the CINEMA's Render Settings dialog:

Output sRGB (not using DeGamma)	This option is available only in CINEMA 4D versions prior to R12. Unless you are using DeGamma (http://www.thirdpartyplugins.com/degamma/) to employ a linear workflow throughout your scene, TurbulenceFD will convert the fire and smoke colors to sRGB to ensure proper lighting and display of HDR fire colors. Uncheck this box if you are using DeGamma.
Adaptive Step Size	Should be checked to improve render performance. Disable it only if you're know what you're doing.
Noise Threshold	Specifies the noise threshold that the adaptive step size accepts before refining.
Fluid Motion Blur	Enables motion blur based on the fluid velocity. Note that you will also need to cache velocity channel in your simulation.
Motion Blur Sub-Steps	The number of samples per frame to use for the motion blur. If you see noisy results, try increasing this value. It will also increase the render times though.

Preferences

Settings in the TurbulenceFD tab of the CINEMA's preference dialog:

Default Cache Base Directory	Specifies the base directory to use for simulation caches on new TurbulenceFD objects (see Cache parameters below).
Autosave Project To Cache	If checked, the project file will be saved to the cache directory when a simulation is started. That way you always have a reference copy of the parameters and setup that were used to create a cache.
Automatically check for updates	Check for new versions of TurbulenceFD automatically. It is recommended to leave this option enabled, especially during the Beta phase. When a new version becomes available, you will be notified so you can decide whether to update.
Ask before overwriting simulation cache	If the currently selected cache is not empty and you start a simulation, TurbulenceFD will ask before overwriting the cache.