Before we begin

- **Big spoiler alert for those who like cool and fantastic looking CFD pictures.**
- **Cool and fantastic looking pictures do not imply that the results are correct.**

As we are going to do some post-processing, we want to remind you that CFD does not stand for

- **C**olorful **F**luid **D**ynamics
- **C**areful **F**it of **D**ata
- **C**olors **F**or **D**irectors

ParaView in a nutshell

ParaView executive summary

- ParaView is an open-source, multi-platform data analysis and visualization application for data sets of varying sizes from small to very large.
- ParaView runs on distributed and shared memory parallel systems as well as single processor systems, laptops and desktop workstations.
- It has been successfully tested on Windows, Linux, Mac OS X, IBM Blue Gene, Cray XT3 and various Unix workstations and clusters.
- It can be run on supercomputers to analyze datasets of petascale size.
- It has a mature, feature-rich, flexible, and intuitive user interface.
- ParaView is built on an extensible architecture based on open standards.
- Commercial maintenance and support.
- ParaView uses a permissive BSD license that enables the broadest possible audience (including commercial organizations), to use the software royalty free, for most purposes.

ParaView in a nutshell

ParaView executive summary

- The data exploration can be done interactively in 3D or programmatically using ParaView's batch processing and Python scripting capabilities.
- Under the hood, ParaView uses the Visualization Toolkit library (VTK) as the data processing and rendering engine and has a user interface written using the Qt cross-platform application framework.

http://www.ParaView.org

http://www.vtk.org

http://www.qt.io

ParaView in a nutshell

ParaView executive summary

- ParaView is slowly becoming the standard *de-facto* for scientific visualization. It is being used in the academia, government, and many commercial institutions worldwide.
- PavaView development started in 2000 as a collaborative effort between Los Alamos National Laboratories and Kitware Inc. (lead by James Ahrens).
- ParaView is in continuous development. As of today, the primary contributors are:

Sandia National Laboratories

Kitware

Los Alamos NATIONAL LABORATORY EST. 1943

ARL

U.S. Army Research Laboratory

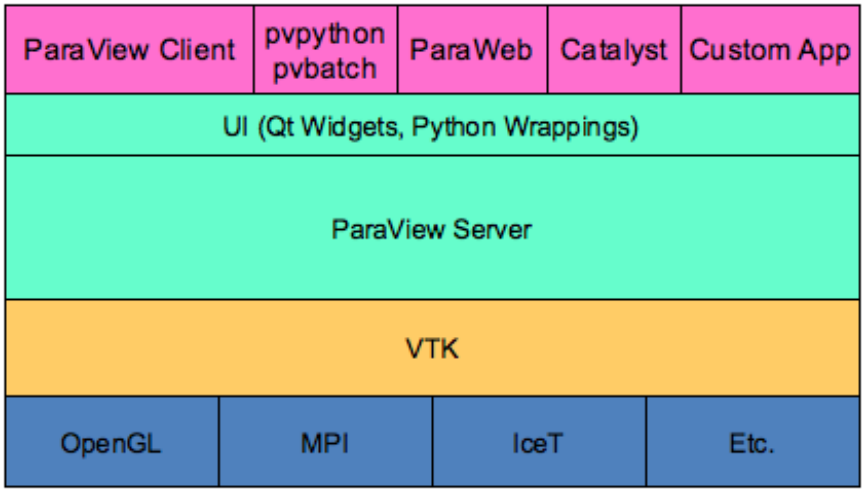
ASC™

Advanced Simulation and Computing Program

ParaView in a nutshell

ParaView technicalities

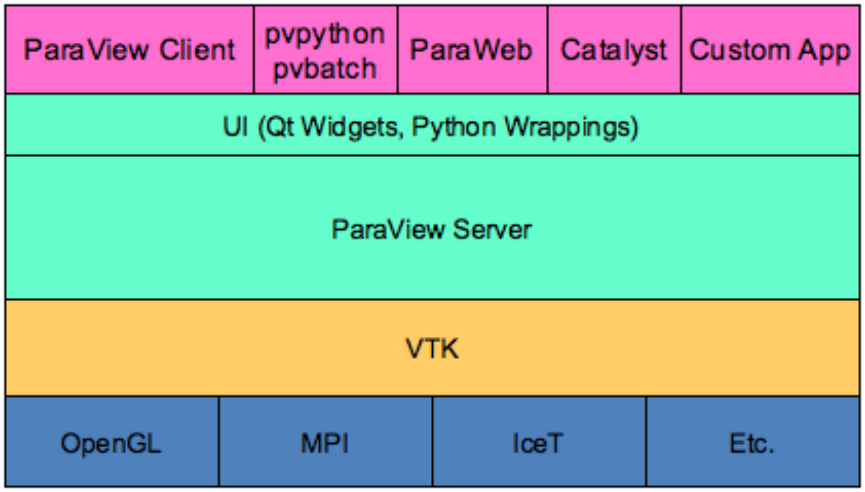
- When you install ParaView, you are basically installing a client, a server, the VTK library, and a few additional applications and libraries (pvpython, pvbatch, ParaWeb, Catalyst and so on).
- The Visualization Toolkit (VTK) library provides the basic visualization and rendering algorithms.
- To provide basic functionalities such as rendering, parallel processing, file I/O, and parallel rendering, VTK uses several additional libraries.
- The application most people associate with ParaView is really just a small client application (`ParaView`) built on top of a tall stack of libraries that provide ParaView with its functionalities.
- We are going to use the client (`ParaView`), and whenever we refer to ParaView we are referring to the client.
- `pvbatch` is also a Python interpreter that runs Python scripts for ParaView. The one difference is that while `pvpython` is meant to run interactive scripts, `pvbatch` is designed for batch processing.
- We are going to address some basic scripting using `pvbatch`.



ParaView in a nutshell

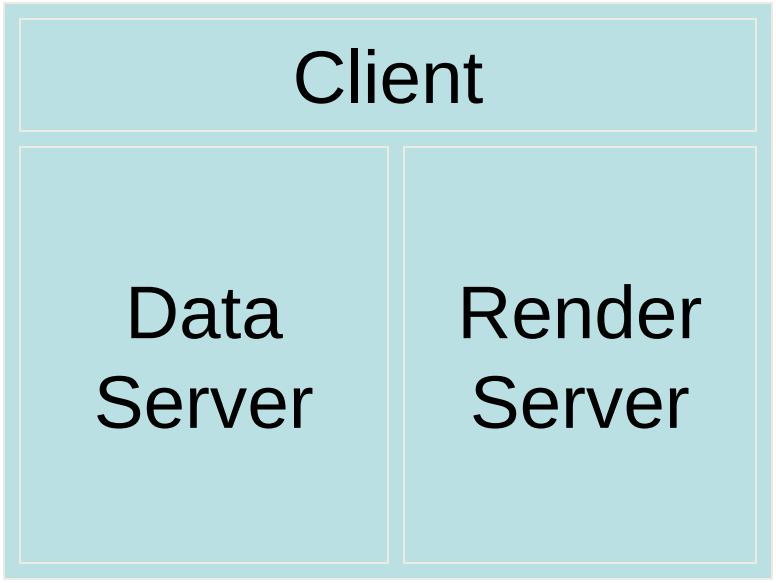
ParaView technicalities

- The ParaView server (`pvserver`), is used for remote visualization. This executable does all of the data processing and, potentially, the rendering. You can connect `ParaView` to `pvserver` running remotely on an HPC resource. This allows you to build and control visualization and analysis on the HPC resource from your desktop as if you were simply processing it locally.
- Alternatively, you can use `pvdataserver` and `pvrenderserver`. These can be thought as the `pvserver` split into two separate executables: one for the data processing part, `pvdataserver`, and one for the rendering part, `pvrenderserver`. Splitting these into separate processes makes it possible to perform data processing and rendering on separate computing nodes.
- We are not going to address how to use `pvserver`, `pvdataserver`, and `pvrenderserver`.
- The main reason is that in order to use these applications requires some IT knowledge that we can not cover on these slides, nevertheless, the way we use the GUI or the scripting capabilities is the same as if we were using `ParaView`.



ParaView in a nutshell

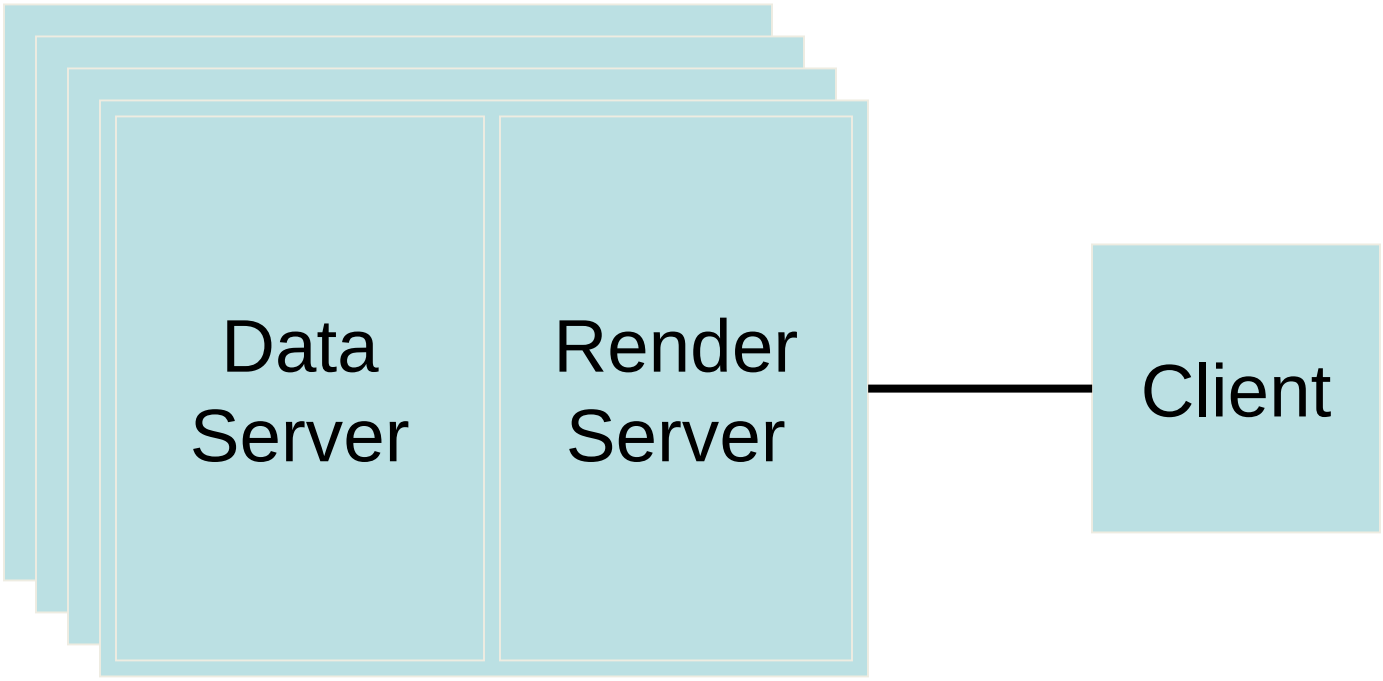
ParaView architecture – Standalone execution



- In standalone mode, the client, data server, and render server are all combined into a single serial application.
- When you run the ParaView application, you are automatically connected to a built-in server so that you are ready to use the full features of ParaView.
- **This is how we are going to work.**

ParaView in a nutshell

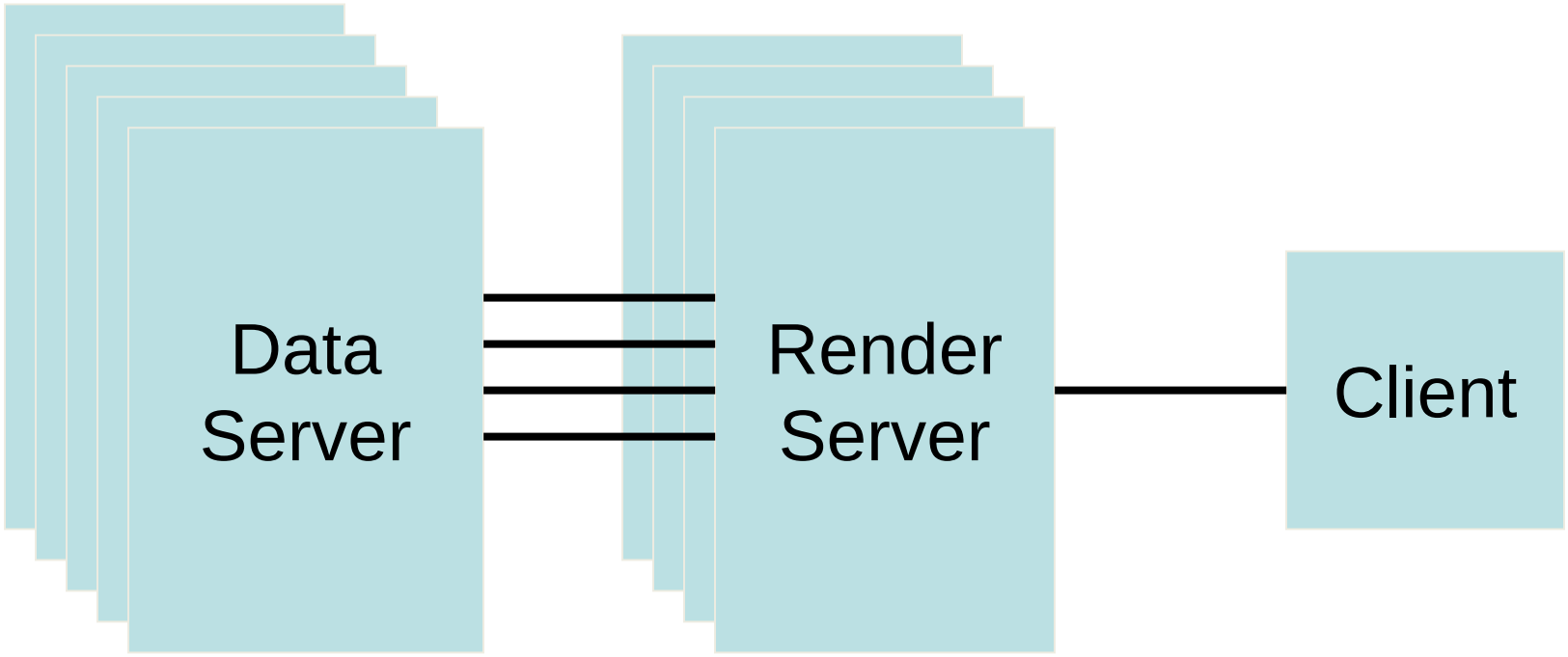
ParaView architecture – Client-Server execution



- In client-server mode, you execute the `pvserver` program on a parallel machine and connect to it with the ParaView client application.
- The `pvserver` program has both the data server and render server embedded in it, so both data processing and rendering take place there.
- The client and server are connected via a socket.

ParaView in a nutshell

ParaView architecture – Client-Render Server-Data Server execution



- In this mode, all three applications are running in separate programs.
- The client is connected to the render server via a single socket connection.
- The render server and data server are connected by many socket connections, one for each process in the render server.

ParaView in a nutshell

http://ParaViewweb.kitware.com

- **What is that?**

  ParaViewWeb is a collection of components that enables the use of ParaView's visualization and data analysis capabilities within Web applications.

  In other words, a ParaView client on the web.

- **Where can I get it?**

  You already have it. It is distributed with ParaView binaries.

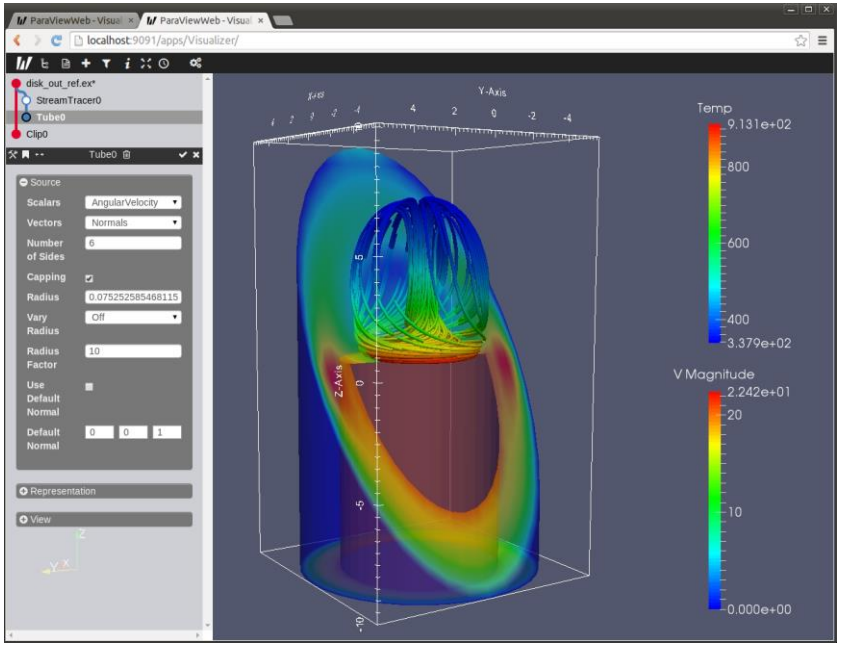
ParaView in a nutshell

- **How do I use it?**

From the terminal (in linux):

$>ParaView_PATH/bin/pvpython
ParaView_PATH/lib/ParaView-4.4/site-packages/ParaView/web/pv_web_visualizer.py
--content ParaView_PATH/share/ParaView-4.4/www/
--data-dir /PATH_TO_DATA
--port 8080

Then open a browser (preferably chrome) and go to http://localhost:8080/apps/Visualizer/



ParaView in a nutshell

http://www.ParaView.org/in-situ/

- **What is that?**

  Catalyst is a library for in situ (also known as co-processing or co-visualization) integration into simulations and other applications.

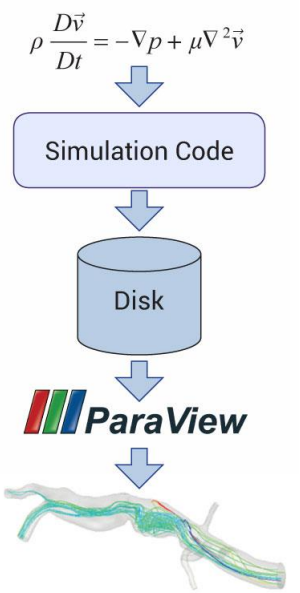
  In other words, the post-processing is done as the simulation runs. No need to read the simulation data to do post-processing.

  Where can I get it?

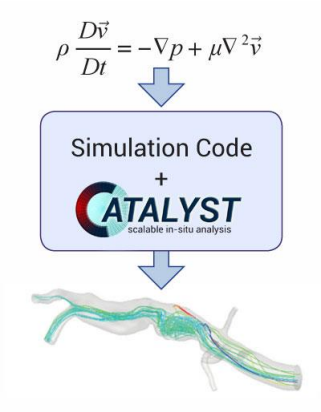
  You already have it. It is distributed with ParaView binaries.

ParaView in a nutshell

- **Advantages of in situ analysis and visualization:**
  - The post-processing is done as the simulation is running, we do not need to wait for the simulation to end.
  - No more expensive IO.
  - We have the full computational power of the supercomputer available to do this processing.
  - Writing the results from the analysis and visualization will be much smaller and faster than the original full simulation data.



Traditional post-processing approach



In situ analysis and visualization, co-processing or co-visualization.

ParaView vs. paraFoam

- paraFoam is the main post-processor distributed with OpenFOAM®. However, you can use other alternatives (commercial or open source).
- Just to make it clear, paraFoam is a wrapper of a third-party open source product named ParaView (www.ParaView.org). In theory paraFoam and ParaView are the same.
- paraFoam/ParaView comes with many built-in filters. By using these filters you can manipulate your data in order to create vectors, streamlines, iso-surfaces, cut-planes, plot over a lines, and so on.
- As we are going to work with OpenFOAM®, let's make clear the difference between ParaView and paraFoam.
- The main post-processing tool provided with OpenFOAM® is a reader module to run with ParaView. The module is compiled into two libraries, **PV4FoamReader** and **vtkPV4Foam**.
- To visualize data obtained from OpenFOAM, you do not need to use paraFoam, you can use an stand-alone version of ParaView.
- In the user guide, it is written that it is highly recommended to use paraFoam.
- However, we have found that paraFoam is significantly slower than ParaView. Therefore, we prefer to use ParaView.

ParaView vs. paraFoam

- Basically, when we use paraFoam we load the libraries `PV4FoamReader` and `vtkPV4Foam`. We also create the empty file *`case.OpenFOAM`* used by paraFoam to load the case.
- If you do not want to use paraFoam, just create the empty file *`case.OpenFOAM`* and open it with ParaView.
- In the terminal type,

  - `$> touch case.OpenFOAM`
  - `$> paraview case.OpenFOAM`

- Or in one single step,

  - `$> touch case.OpenFOAM | paraview case.OpenFOAM`

ParaView vs. paraFoam

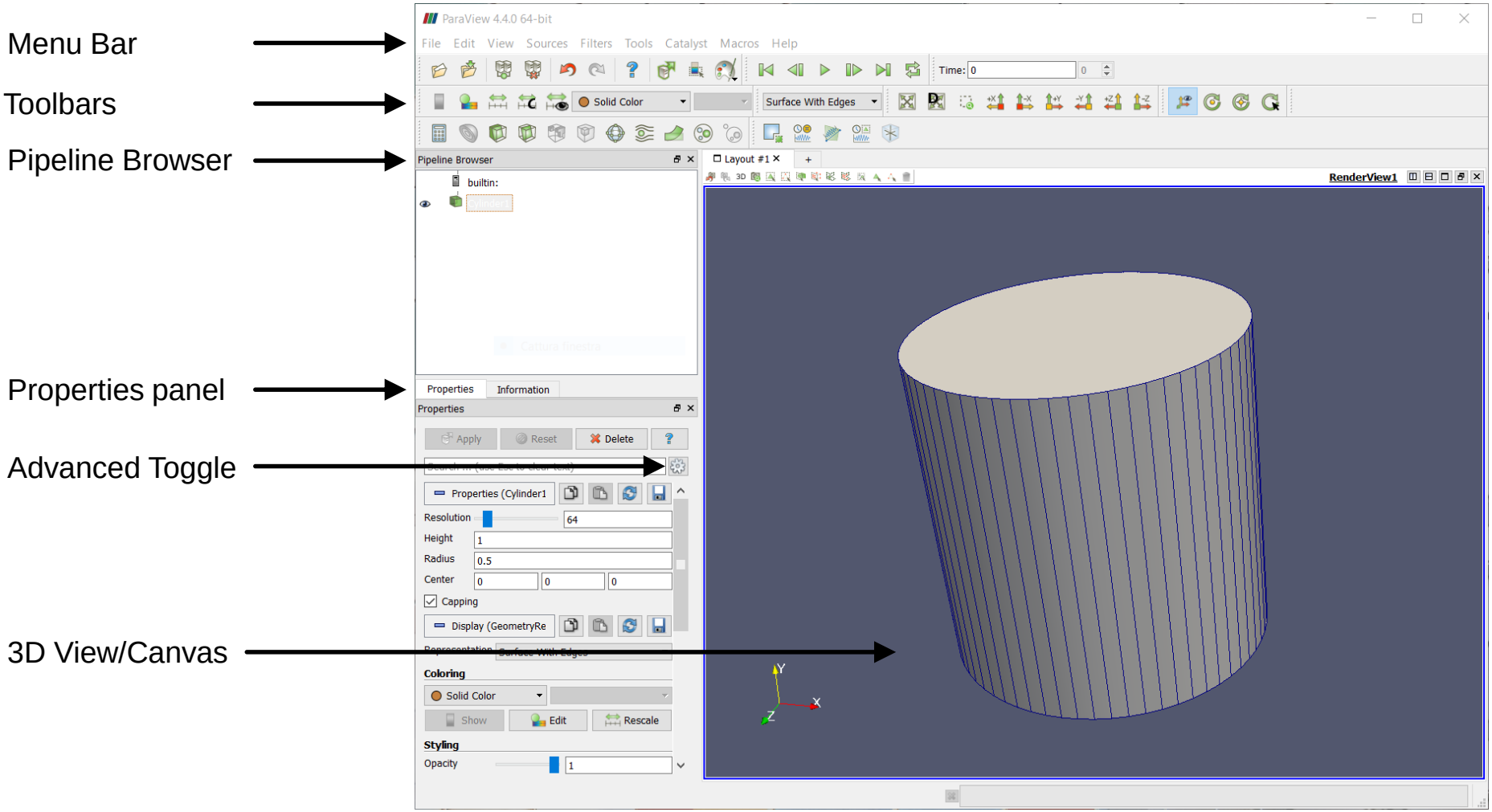
- FYI, if you use the default building options, paraFoam is compiled with no MPI support and no Python support.
- To compile paraFoam with MPI support, in the file *makeParaView4* (located in the directory `$WM_THIRD_PARTY_DIR`) set the option **withMPI** to true,
  - **withMPI = true**
- While you are working with the file *makeParaView4*, you might consider enabling Python support,
  - **withPYTHON = true**
- In the workstations, you will find a stand-alone version of ParaView (version 5.0.1) with MPI support and Python support.

ParaView GUI overview

The following GUI description is the same for paraFoam

ParaView GUI overview

Default User Interface

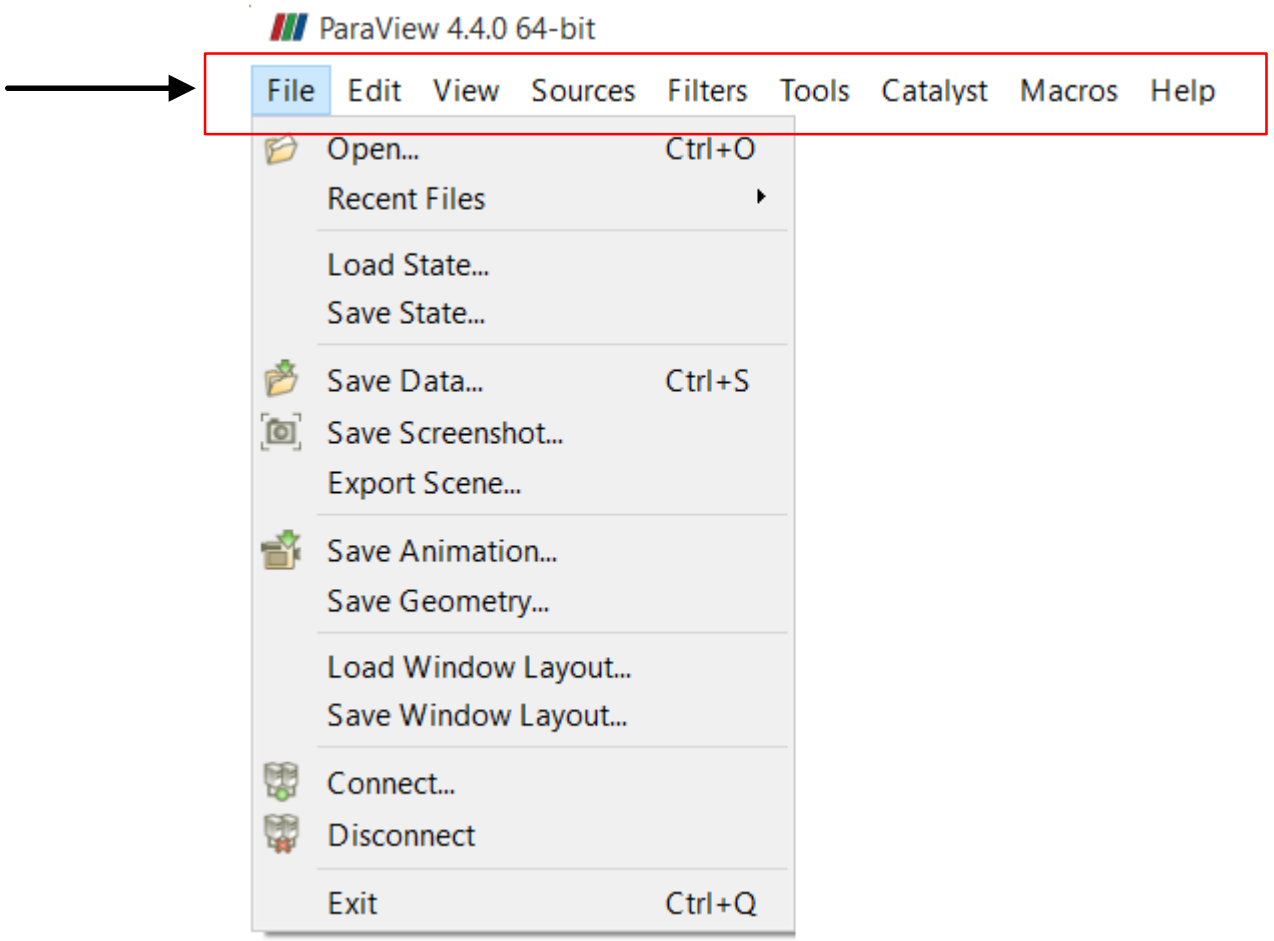


- Note that the GUI layout is highly configurable, so that it is easy to change the look of the window.

ParaView GUI overview

Menu bar

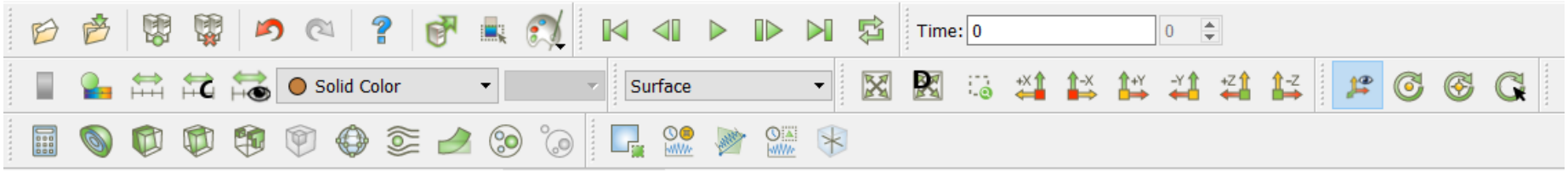
- As with just about any other program, the menu bar allows you to access the majority of features



ParaView GUI overview

Toolbars

- The toolbars provide quick access to the most commonly used features within ParaView.
- By right clicking on the bar or selecting `View → Toolbars`, you can show/hide what features to see on the toolbar.



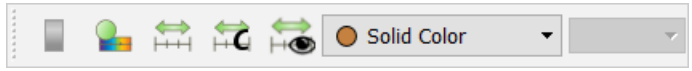
ParaView GUI overview

Toolbars

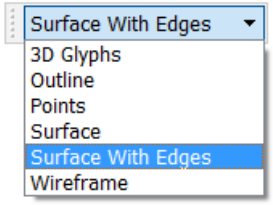
- Main Controls

- VCR Controls

- Current Time Controls



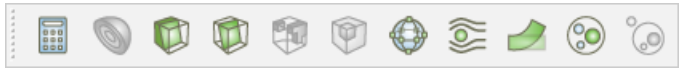
- Active Variable Controls



- Representation Toolbar

- Camera Controls

- Center Axes Controls



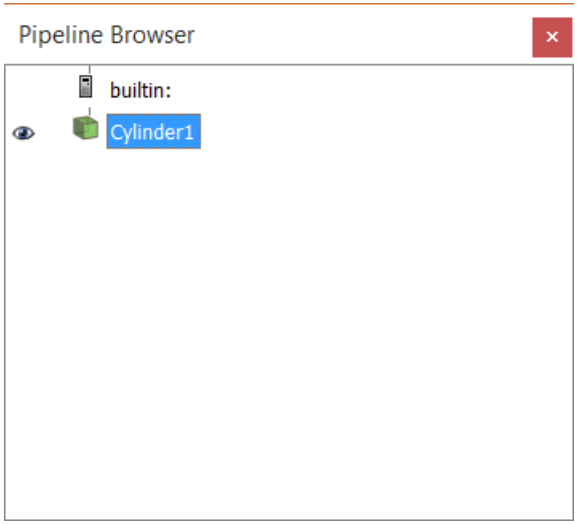
- Common Filters

- Data Analysis Toolbar

ParaView GUI overview

Pipeline Browser

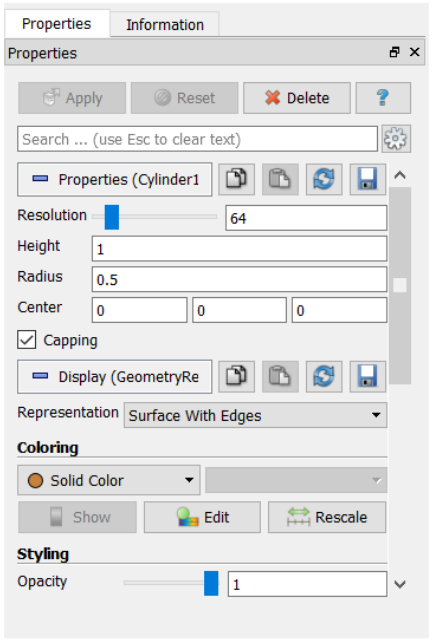
- ParaView manages the reading and filtering of data with a pipeline.
- The pipeline browser allows you to view the pipeline structure and select pipeline objects.
- It provides a convenient list of pipeline objects with an indentation style that shows the pipeline structure.
- By clicking on the eyeball icon you can show/hide objects.

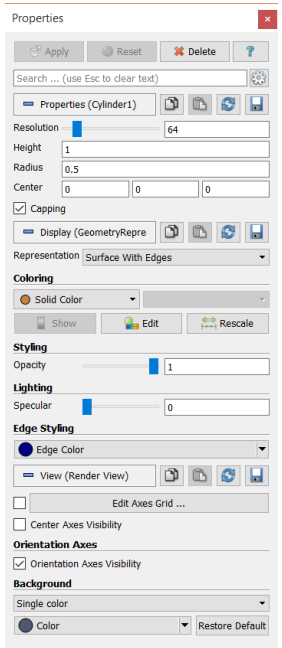


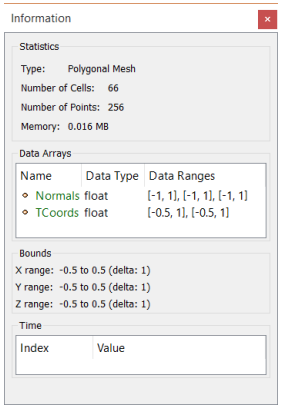
ParaView GUI overview

Pipeline Browser

- The properties panel allows you to view and change the parameters of the current pipeline object.
- On the properties panel there is an advanced properties toggle  that shows and hides advanced controls.
- The properties are by default coupled with an information tab that shows basic summary of the data produced by the pipeline object.





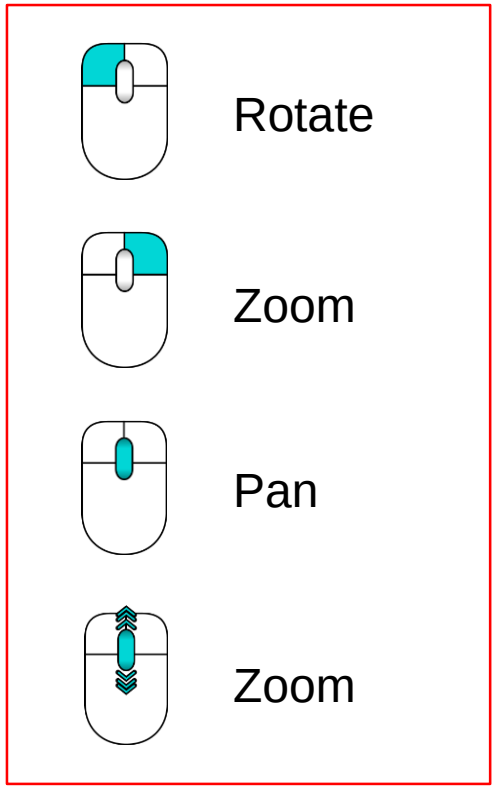


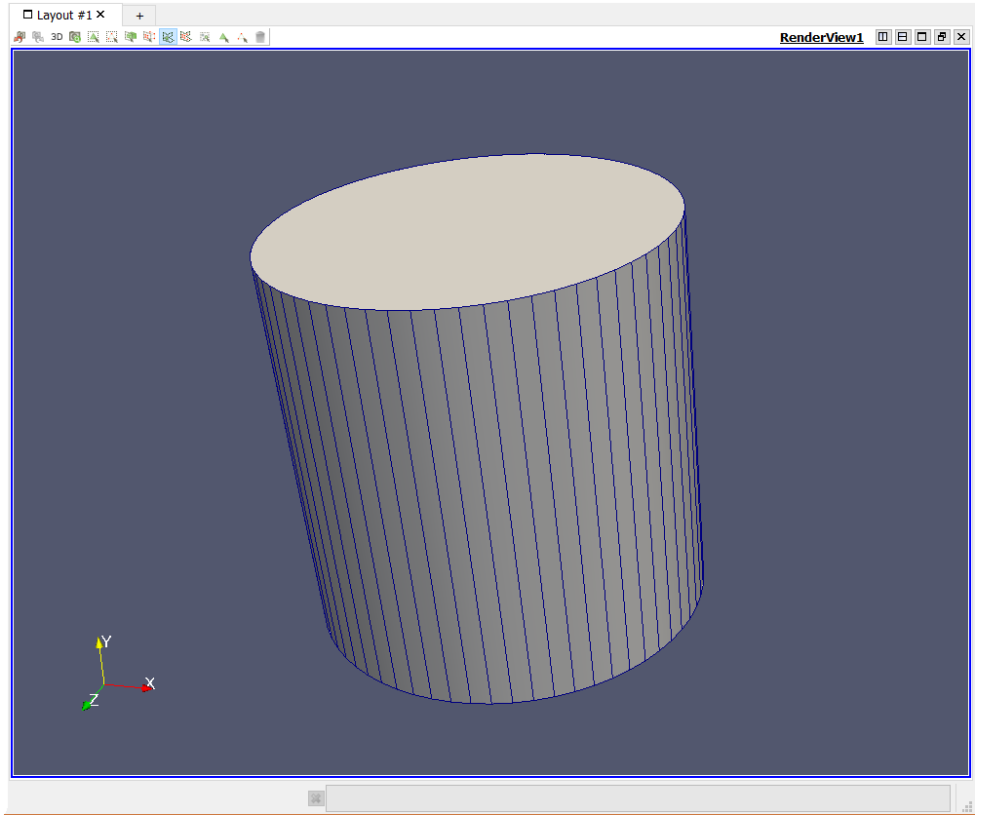
ParaView GUI overview

Mouse interaction

- You can modify the mouse buttons interaction in the tab camera of the menu `Edit → Settings`.

Mouse interaction in the 3D view

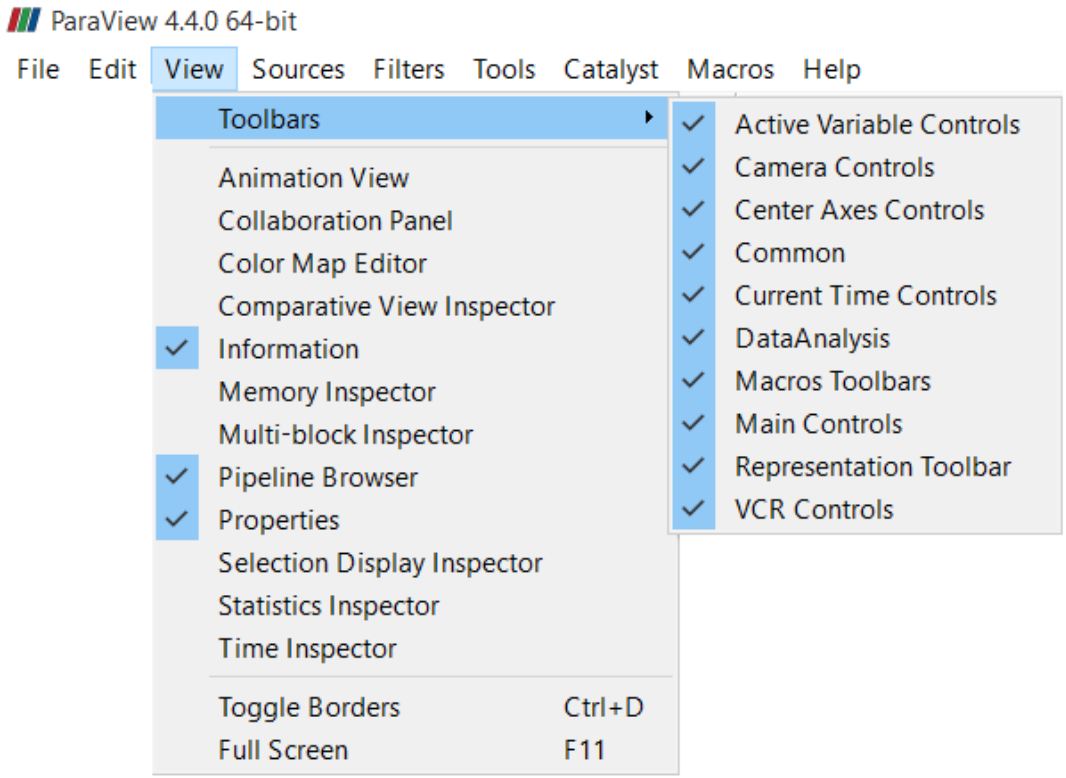




ParaView GUI overview

User Interface Components

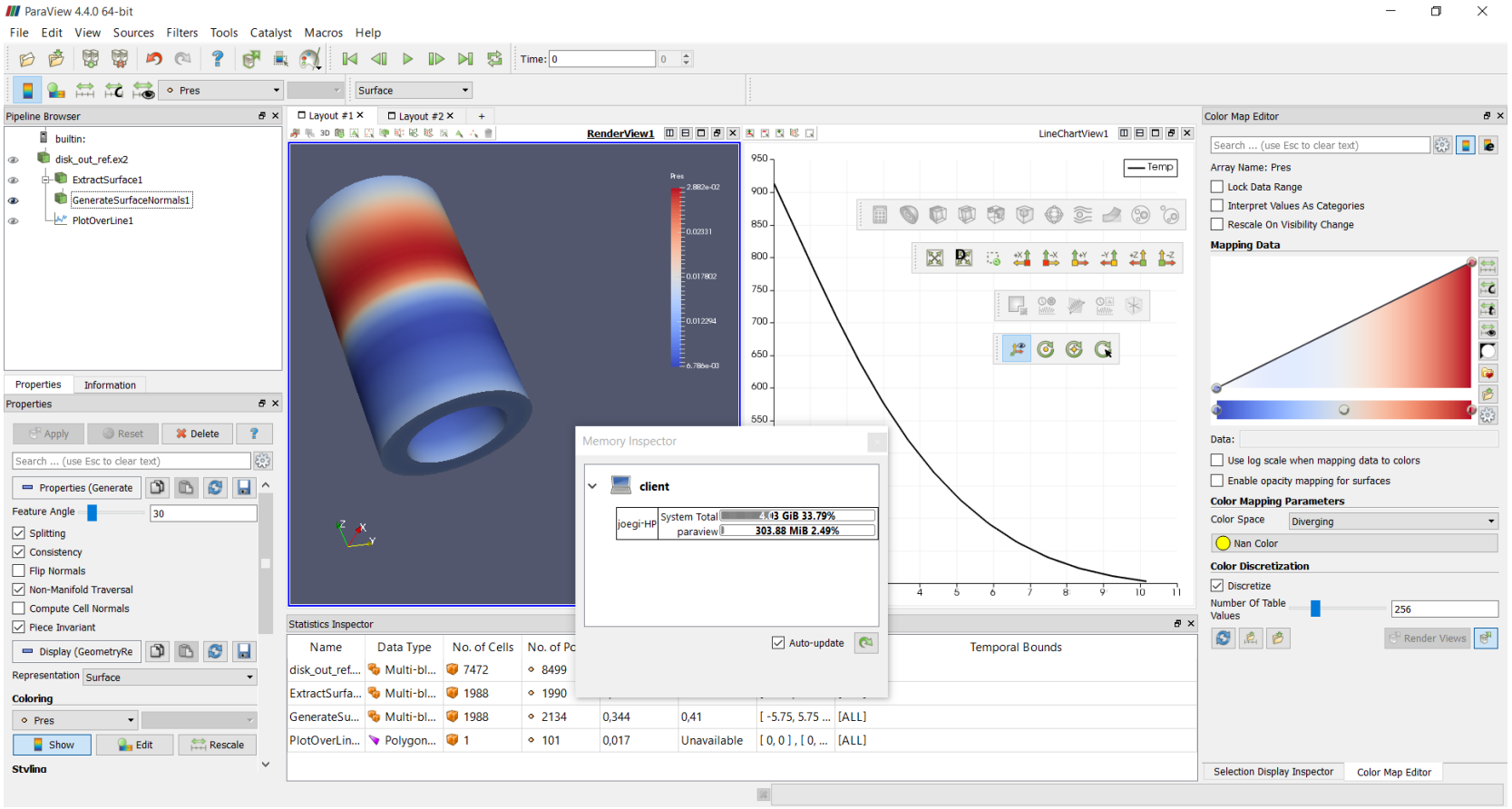
- By right clicking on the toolbar or selecting `View → Toolbars`, you can show/hide what features to see on the user interface.



ParaView GUI overview

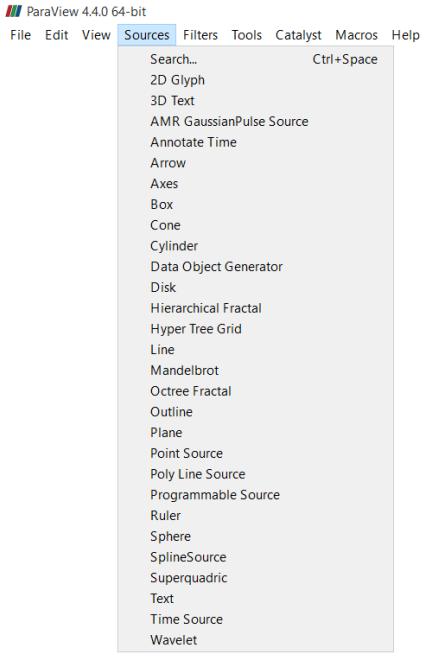
User Interface Customization

- Remember, the default user interface is highly customizable.
- The GUI is designed to present data so that you may view, interact with, and explore your data.



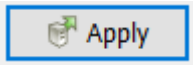
Basic usage – Sources and filters

- The following basic usage description is the same for paraFoam.
- There are two ways to get data into ParaView: read data from a file (a `Reader`) or generate data with a source object (`Sources`).
- All sources are located in the `Sources` menu.
- Sources are very handy when exploring ParaView's features.
- They can be used to add annotations and new objects to interact with the visualization pipeline.

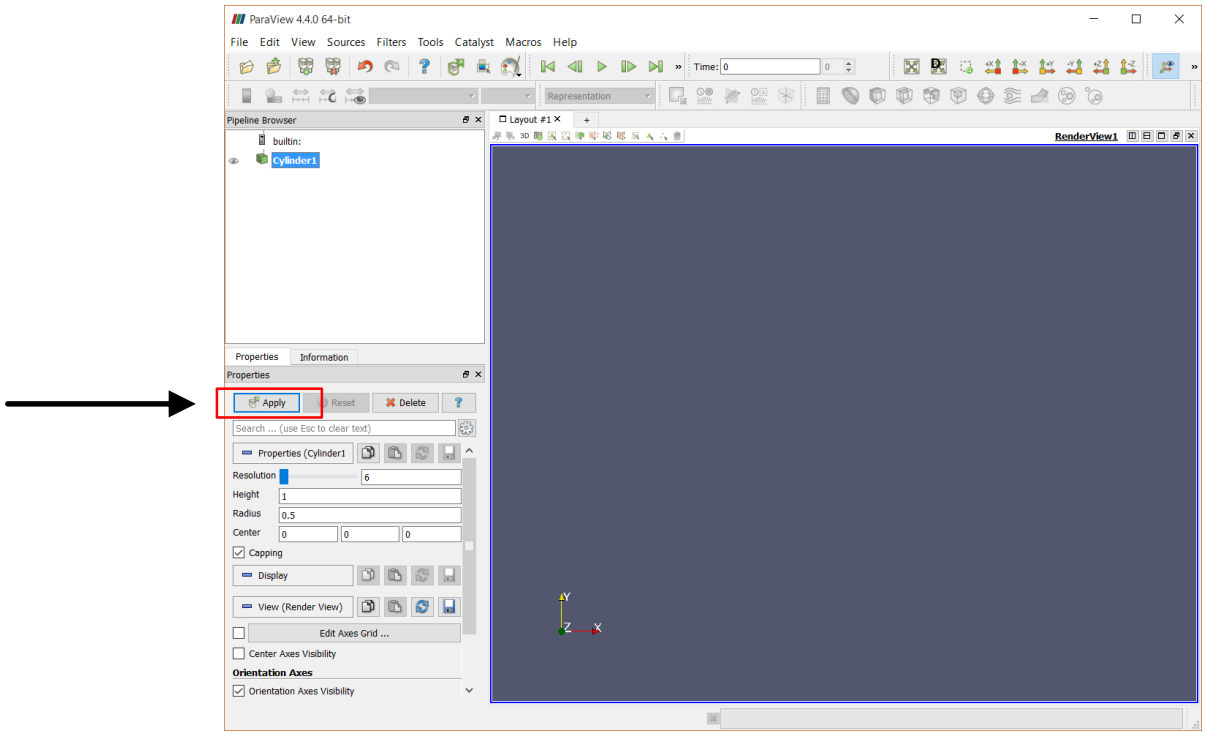


Basic usage – Sources and filters

Creating a Source

- Go to the `Sources` menu and select `Cylinder`.
- You will notice that an item named `Cylinder1` is added to and selected in the pipeline browser.
- You can see the properties of the cylinder source in the properties panel.
- Click the Apply button 

to accept the default parameters.



Basic usage – Sources and filters

Interacting with the 3D view

- At this point, experiment in the 3D view by pressing the mouse buttons to perform different rotate, pan and zoom operations.
- Also, try using the buttons in conjunction with the `shift` and `ctrl` modifier keys.
- Interact with the `Camera Controls` toolbar to change the view and zoom the data.

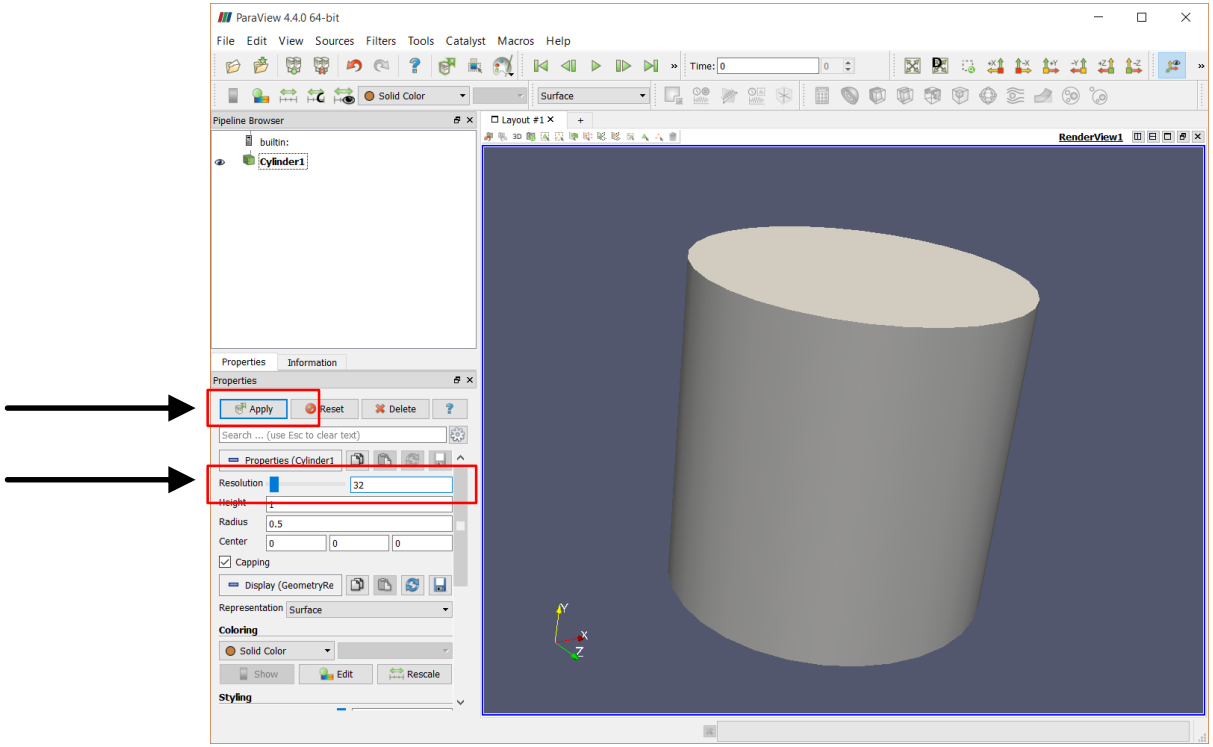
- Interact with the `Center Axes Controls` toolbar, to change the center of rotation of the source.

Basic usage – Sources and filters

Creating a Source

- To change the cylinder properties, select the object in the pipeline browser and modify the properties.
- Increase the cylinder resolution to 32.
- Remember, every time you modify an object in the pipeline browser you need to press the apply button 

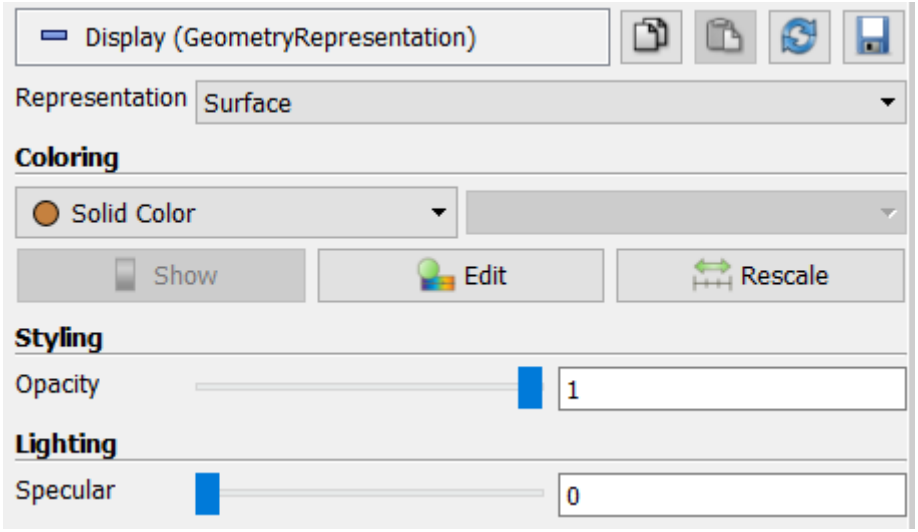
to accept the new parameters.



Basic usage – Sources and filters

Creating a Source

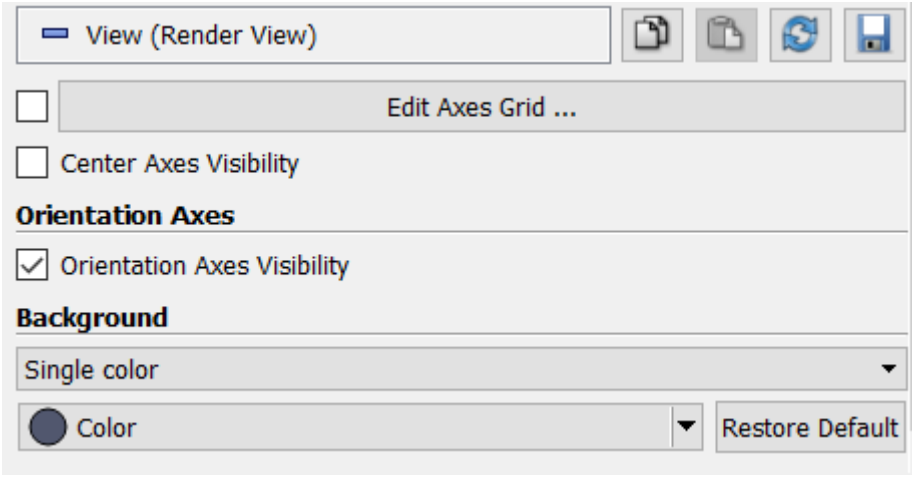
- If you scroll down the properties panel, you will notice a set of `Display` properties.
- These options affects how the object is visualize in the 3D view. Try to change the color and opacity of the source.
- You will also notice that you do not need to press the Apply button 



Basic usage – Sources and filters

Creating a Source

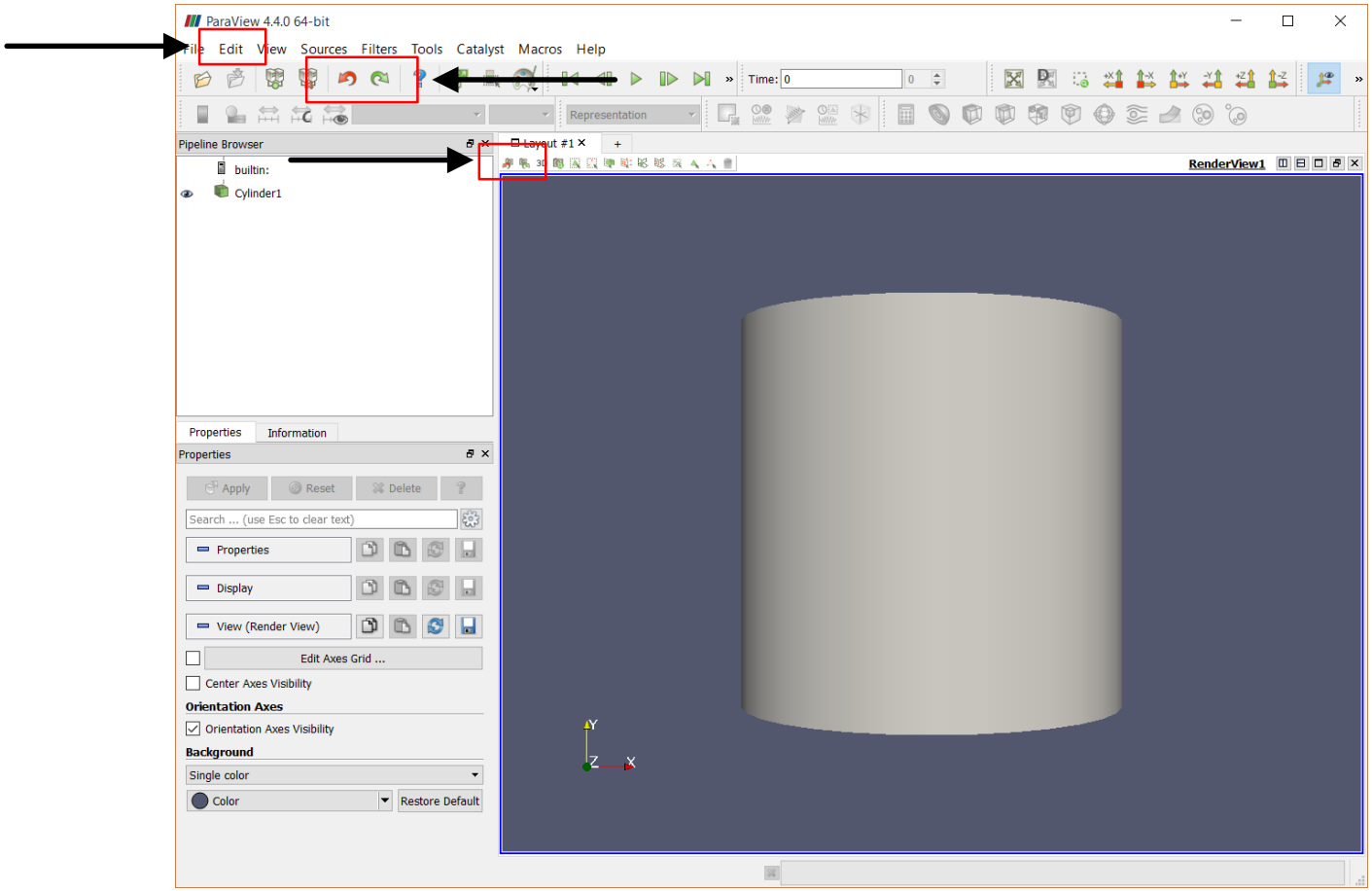
- If you scroll down the properties panel, you will notice a set of `View` properties.
- These options affects the background of the 3D view, and shows axes visibility and the bounding box visibility.
- You will also notice that you do not need to press the Apply button 



Basic usage – Sources and filters

Creating a Source

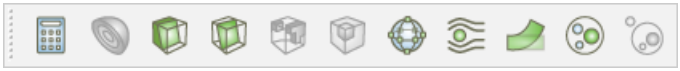
- You can undo and redo pipeline and camera modifications.
- You can access the undo and redo functions from the `Edit` menu or the toolbar.



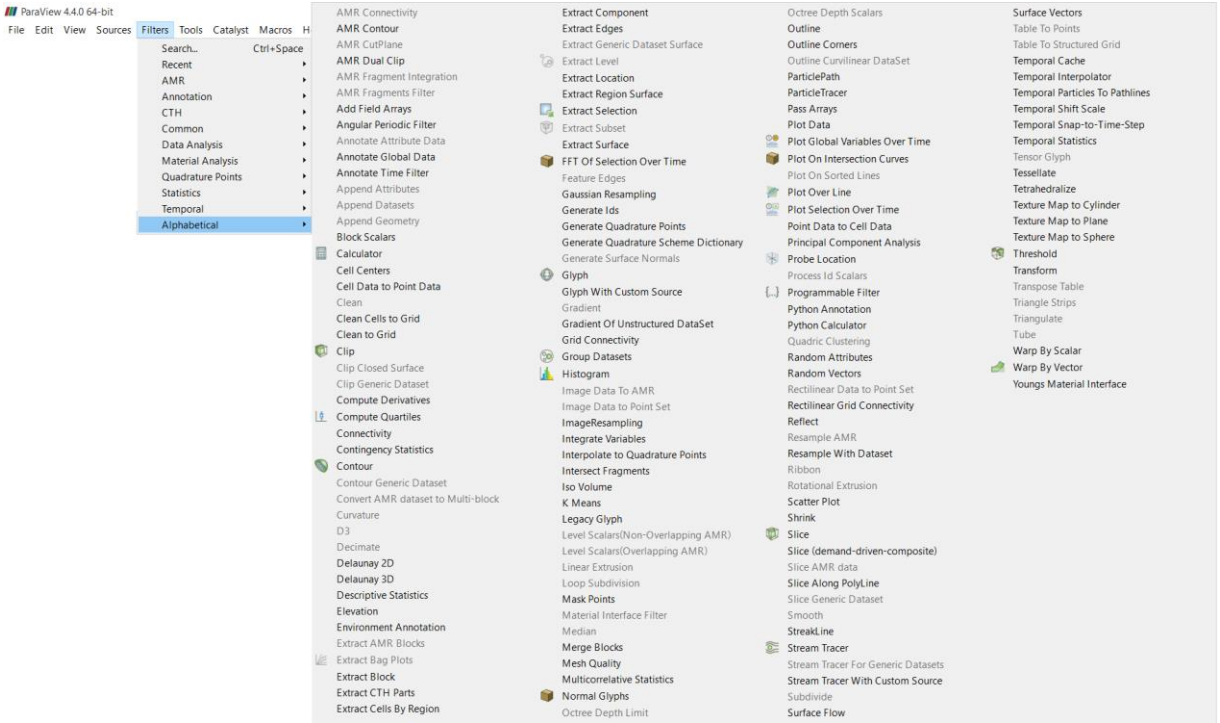
Basic usage – Sources and filters

Creating a Filter

- Filters are functions that generate, extract or derive features from the data.
- They are attached to readers, sources, or others filters.
- You can access the most commonly used filters from the `Common Filters` toolbar



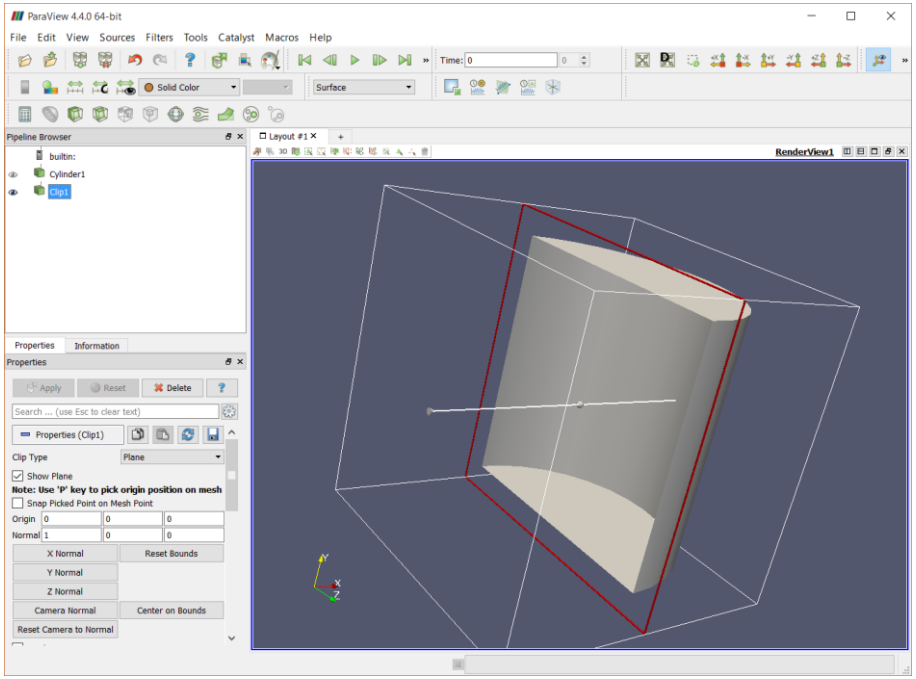
- You can access all the filters from the menu `Filter`.



Basic usage – Sources and filters

Creating a Filter

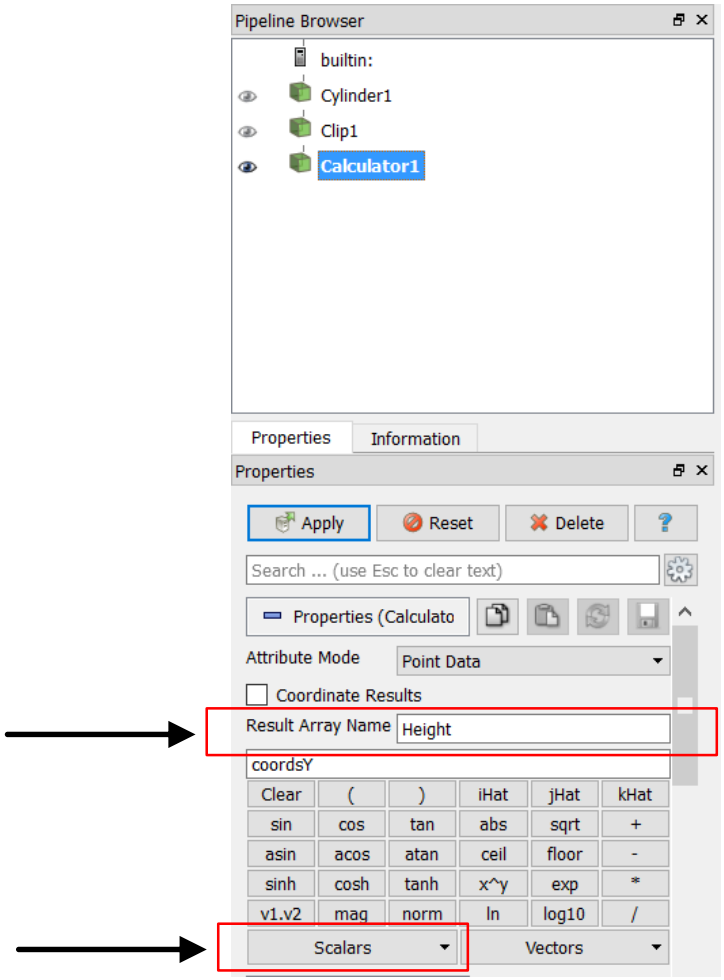
- Let us apply a filter to the cylinder source.
- From the `Common Filters` toolbar select the `Clip` filter  and use the default options.
- You will see that the new filter is automatically selected and rendered.
- Remember, to show/hide objects in the pipeline browser press the eyeball icon 



Basic usage – Sources and filters

Creating a Filter

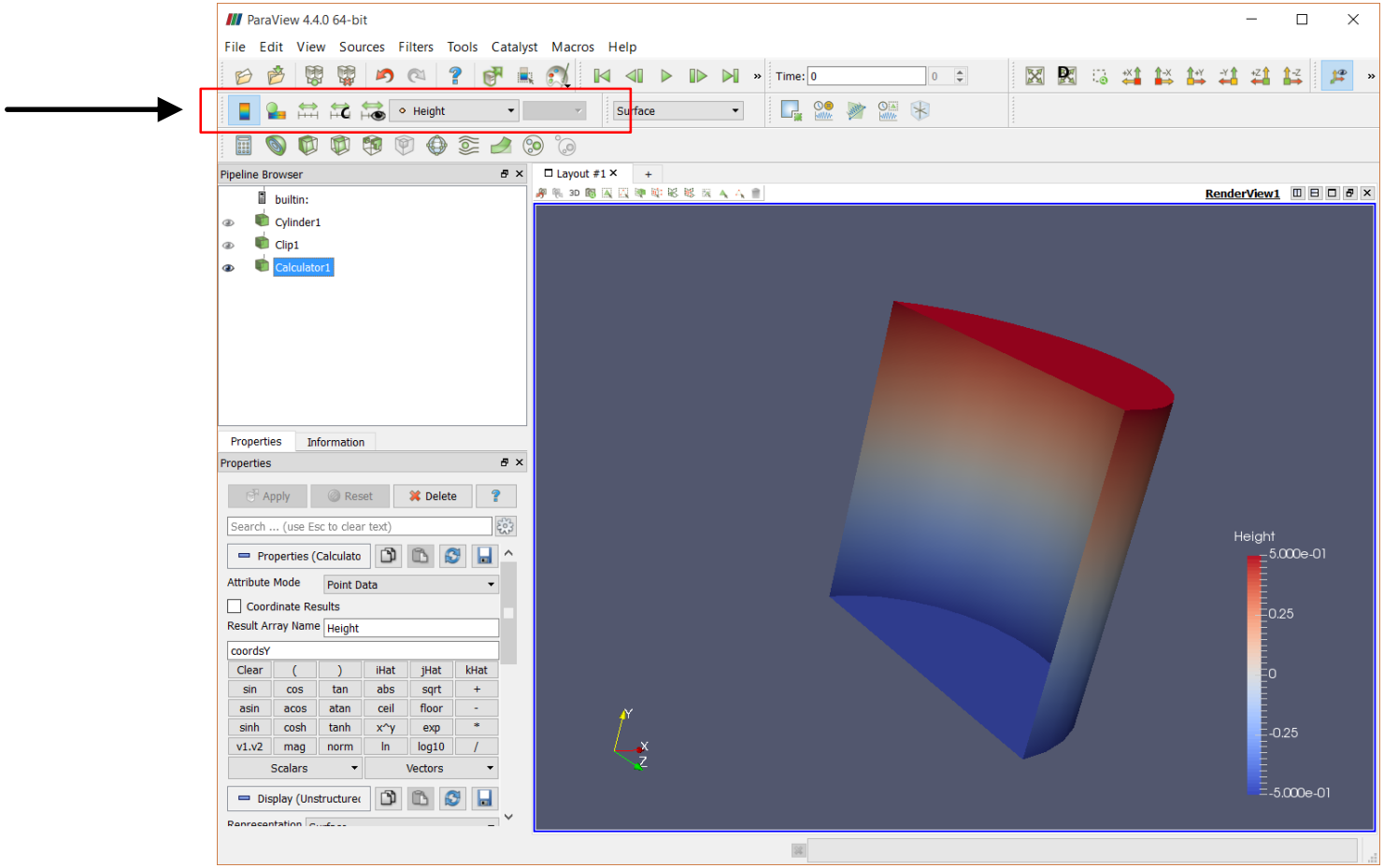
- Let us apply the calculator filter to the `Clip1` object.
- From the `Common Filters` toolbar select the `Calculator` filter 
- Change the `Result Array Name` to `Height`.
- Select from the `Scalars` drop-down menu `coordsY`.
- Click Apply 



Basic usage – Sources and filters

Creating a Filter

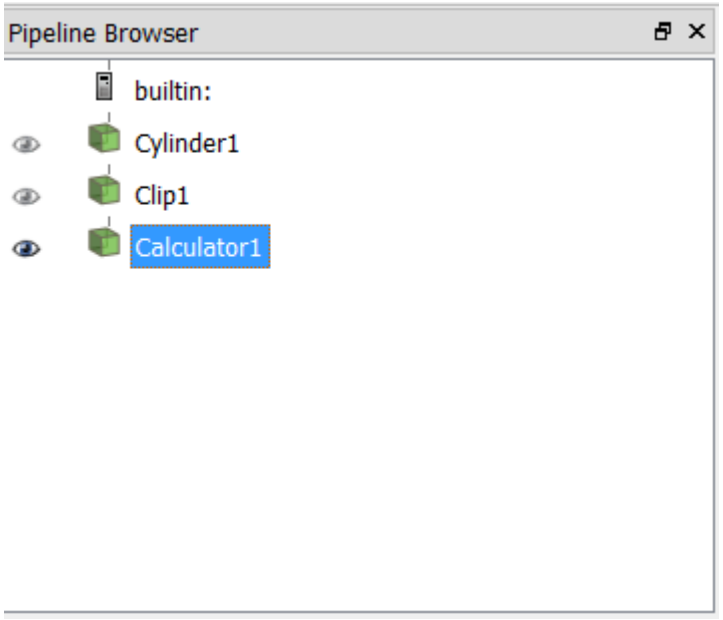
- Let us color the cylinder by the variable created using the calculator.
- From the `Active Variables Controls` toolbar select `Height`.



Basic usage – Sources and filters

Creating a Filter

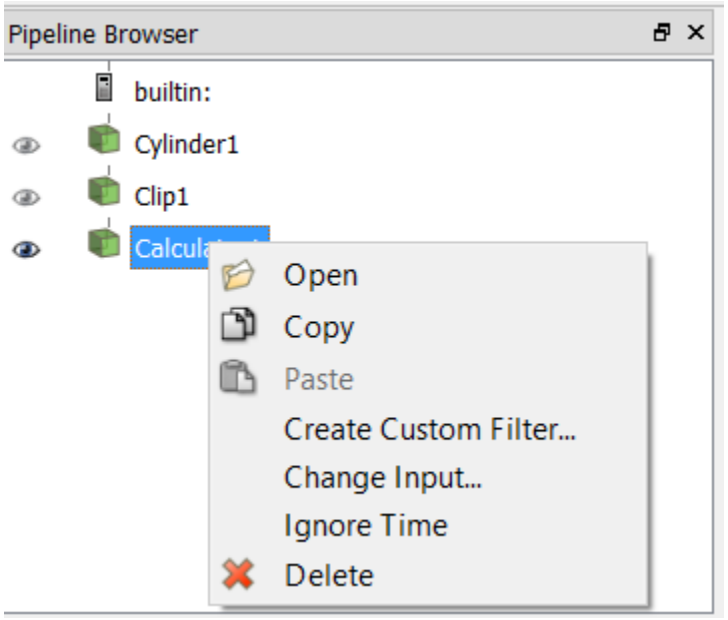
- Notice the hierarchy in the pipeline browser.
- The `Clip1` filter is applied to the `Cylinder1` source, and the `Calculator1` filter is applied to the `Clip1` filter.
- The object `Calculator1` will inherit everything from `Clip1` (and therefore from `Cylinder1`), but `Clip1` will not inherit anything from `Calculator1`.



Basic usage – Sources and filters

Deleting a Source or a Filter

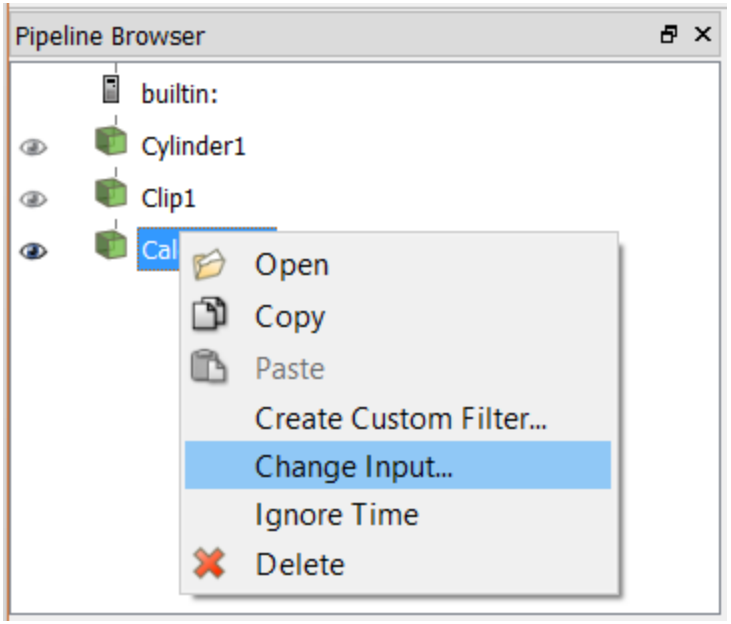
- If you want to erase a source or a filter, just right click on the object in the pipeline browser and select `Delete`.
- Alternatively, go to the `Edit` menu and select `Delete`.



Basic usage – Sources and filters

Changing Source hierarchy

- If you want to change how the objects are connected in the pipeline browser (and hence the hierarchy), just right click on the object in the pipeline browser and select `Change Input`.

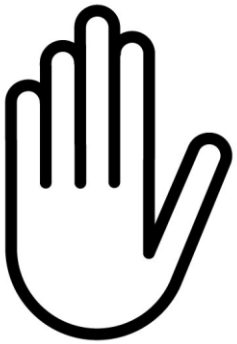


- Alternatively, go to the `Edit` menu and select `Change Input`.
- At this point try to add more sources and filters, and play around with the different options available.

Scientific visualization with paraFoam/ParaView

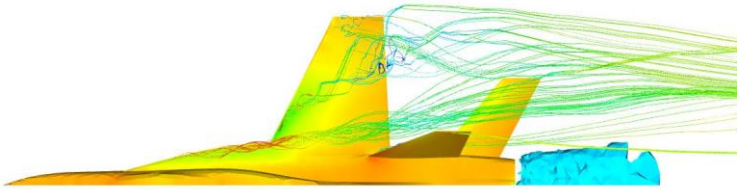
- Let’s do some scientific visualization using paraFoam. Go to the directory:

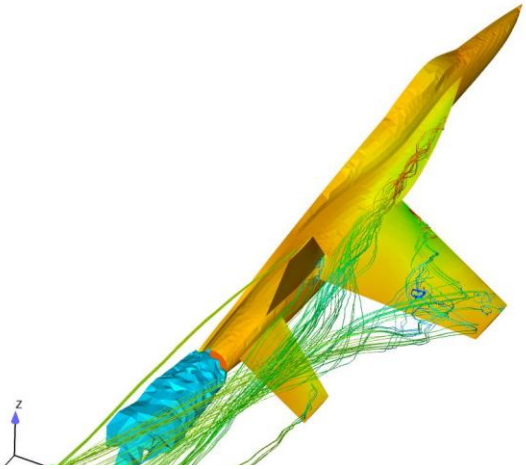
```
yf17
```

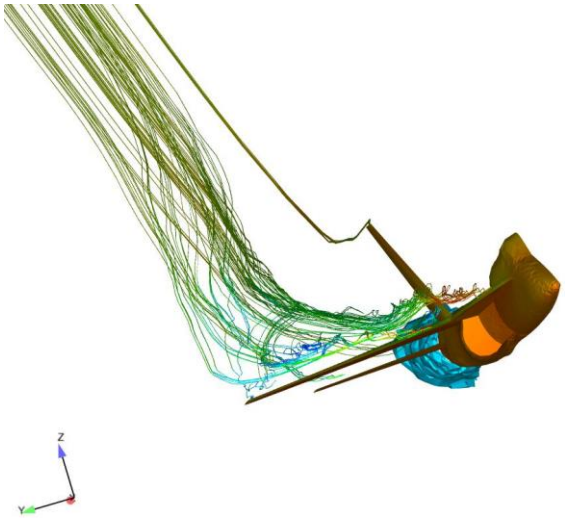


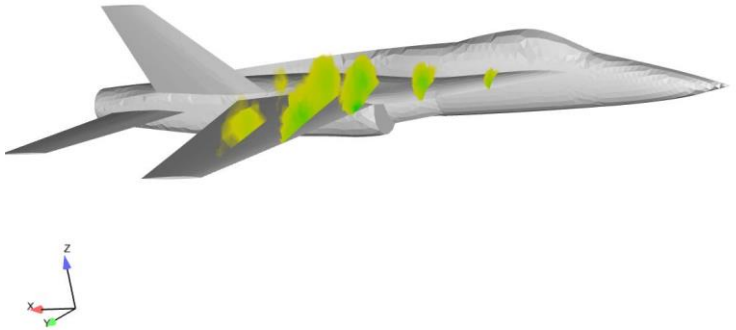
- From this point on, please follow me.
- We are all going to work at the same pace.
- Remember, `$PTOFC` is pointing to the path where you unpacked the tutorials.

Scientific visualization with paraFoam/ParaView



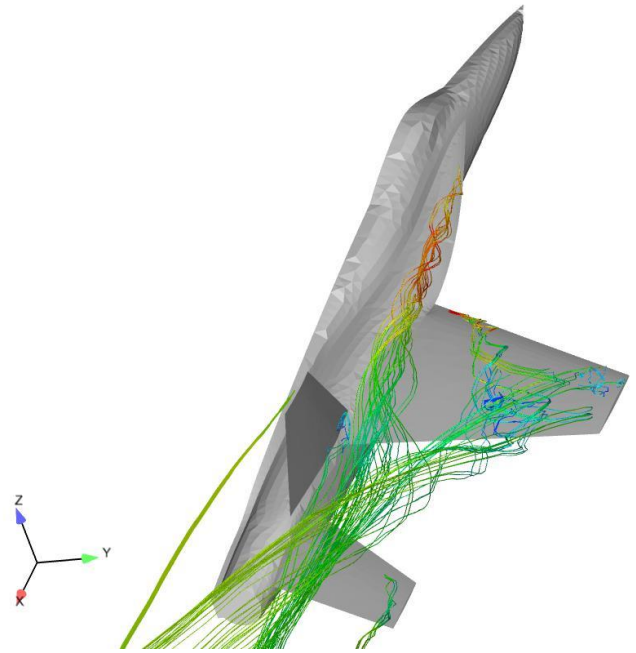






Some pretty pictures

Scientific visualization with paraFoam/ParaView



Some pretty pictures – Kind of a qualitative validation

Scientific visualization with paraFoam/ParaView

What are we going to do?

- We will use this case to introduce paraFoam/ParaView.
- We do not need to generate the mesh or run the simulation, the mesh and solution are already in the case directory. However, all the dictionaries to run the simulation are included.
- To find the numerical solution we used the solver `sonicFoam`.
- `sonicFoam` is a transient solver for trans-sonic/supersonic, laminar or turbulent flow of compressible gas.
- In this case we ran an inviscid simulation (Euler equations), we imposed boundary conditions at the intake and exhaust of the airplane to let the flow go out (engine intake) and go in (exhaust gases) of the domain, and we set a free stream velocity with an incidence angle.

Scientific visualization with paraFoam/ParaView

Visualizing the case

- Go to the directory **`$PTOFC/101PARAVIEW/yf17`** and type in the terminal:

1. `$> tar -xzvf c0.tar.gz`
2. `$> cd c0`

- We will do the post processing using paraFoam, in the terminal type

1. `$> paraFoam`

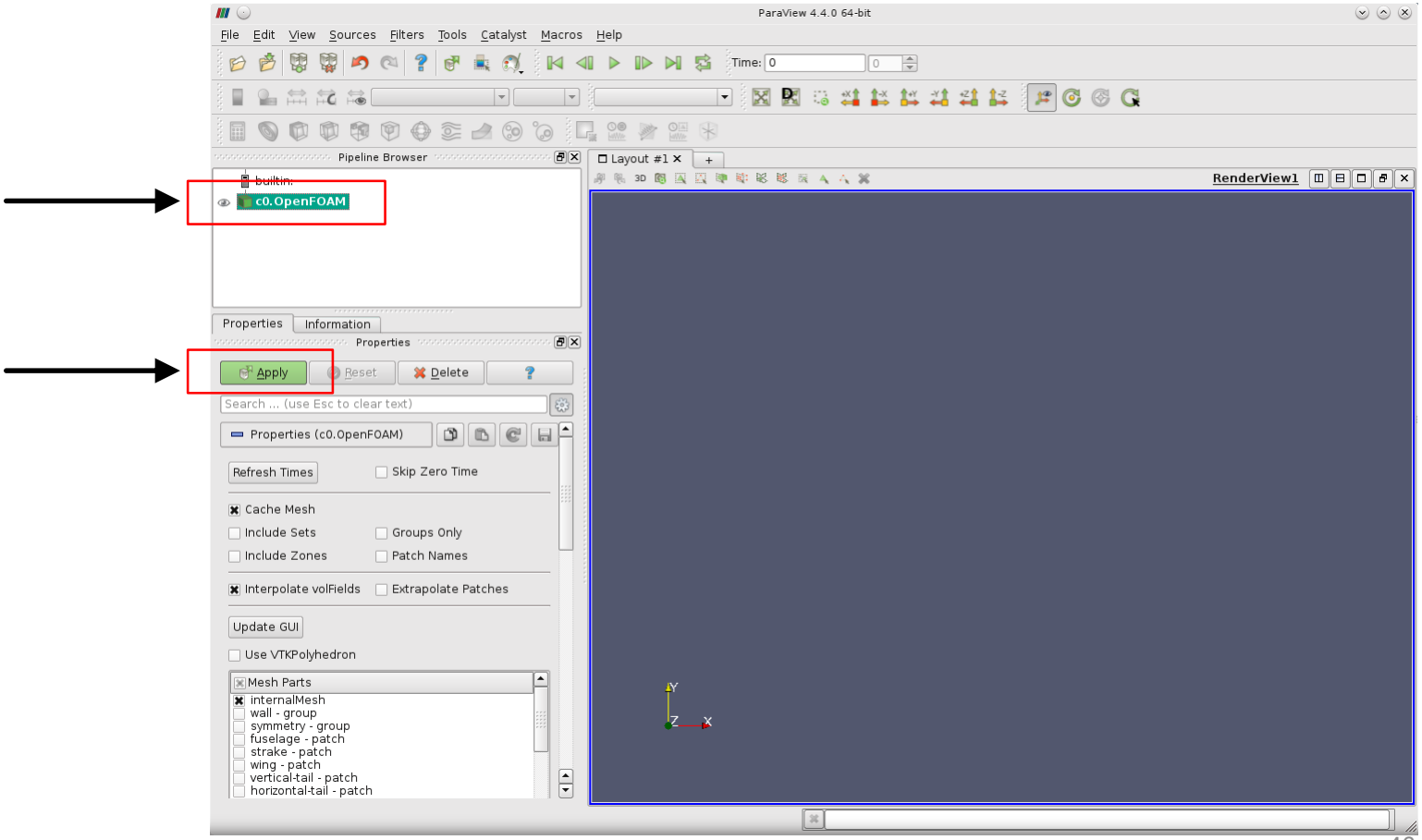
- If you want to use ParaView instead, in terminal type

1. `$> touch case.foam`
2. `$> paraview case.foam`

Scientific visualization with paraFoam/ParaView

Reading in data

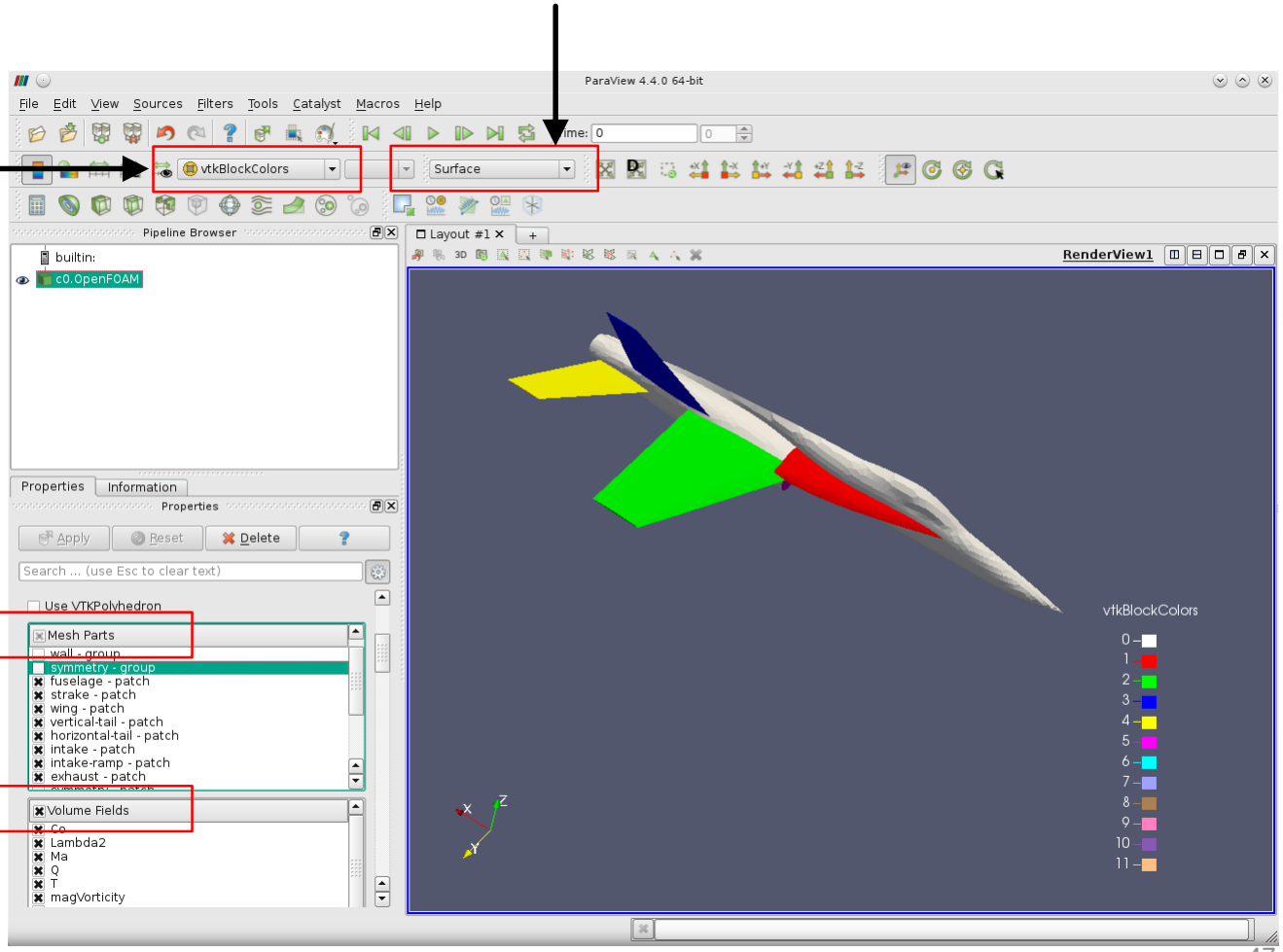
- The object `c0.OpenFOAM` in the pipeline browser is the newly created object (`Reader`).
- To load the data, click Apply 



Scientific visualization with paraFoam/ParaView

Data representation and field coloring

- From the `Representation` toolbar select `Surface`.
- From the `Active Variables Controls` toolbar select `vtkBlockcolors`.



Uncheck `internalMesh` and check all the mesh parts related to the airplane

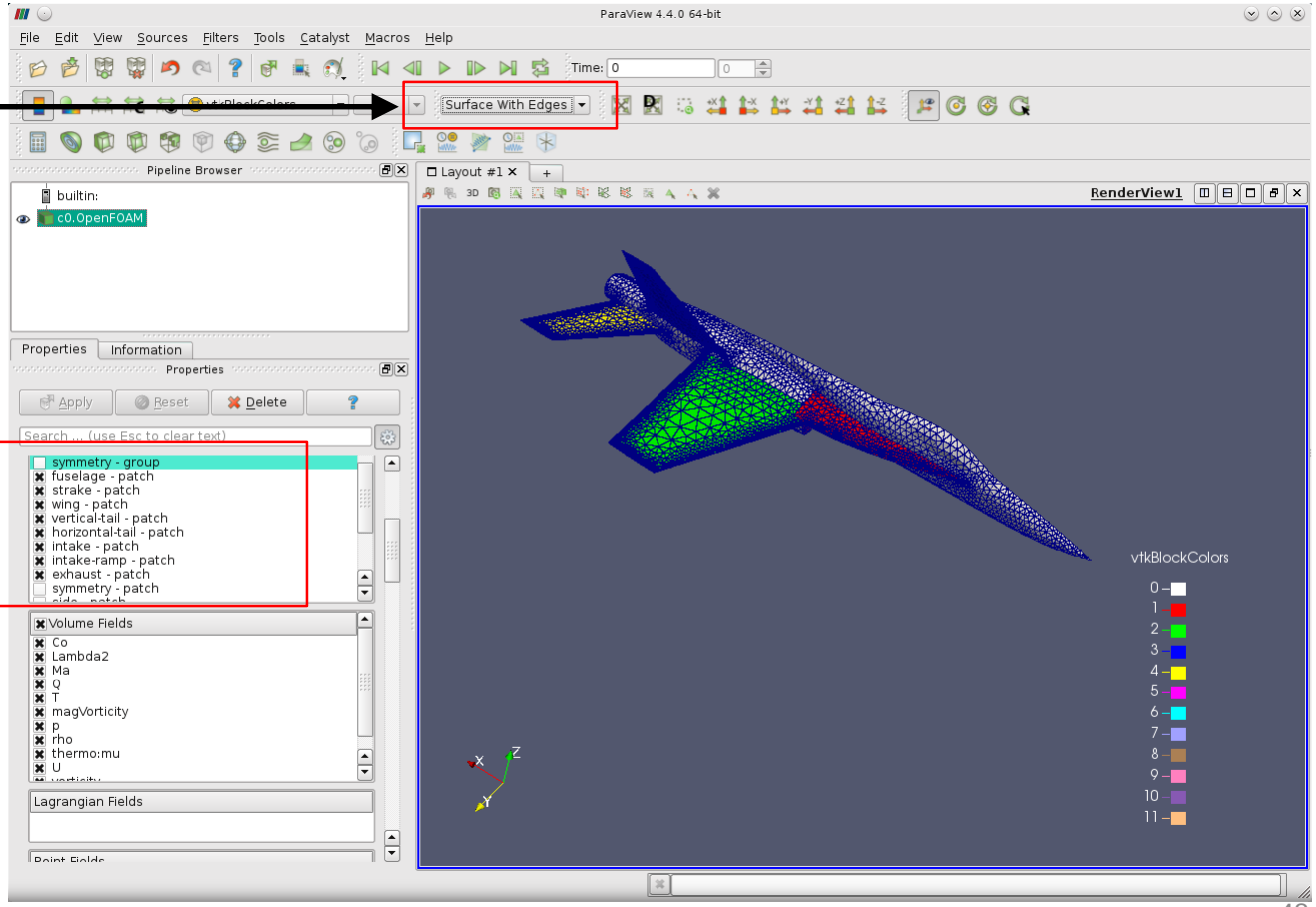
Select all fields

Scientific visualization with paraFoam/ParaView

Data representation and field coloring

- From the `Representation` toolbar select `Surface with Edges`.
- Feel free to play with the other available representation options. For the moment, avoid the volume representation.

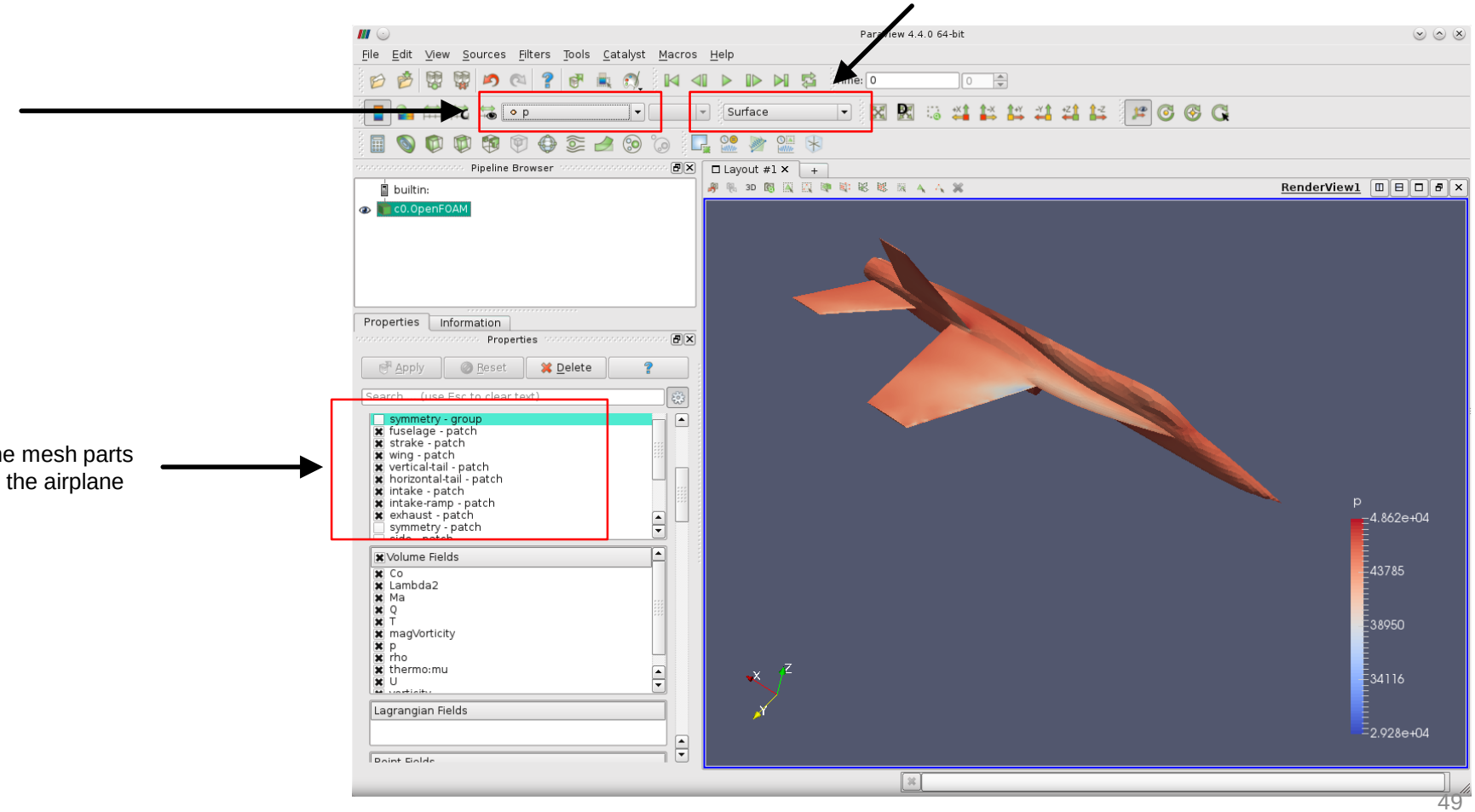
Select only the mesh parts associated to the airplane



Scientific visualization with paraFoam/ParaView

Data representation and field coloring

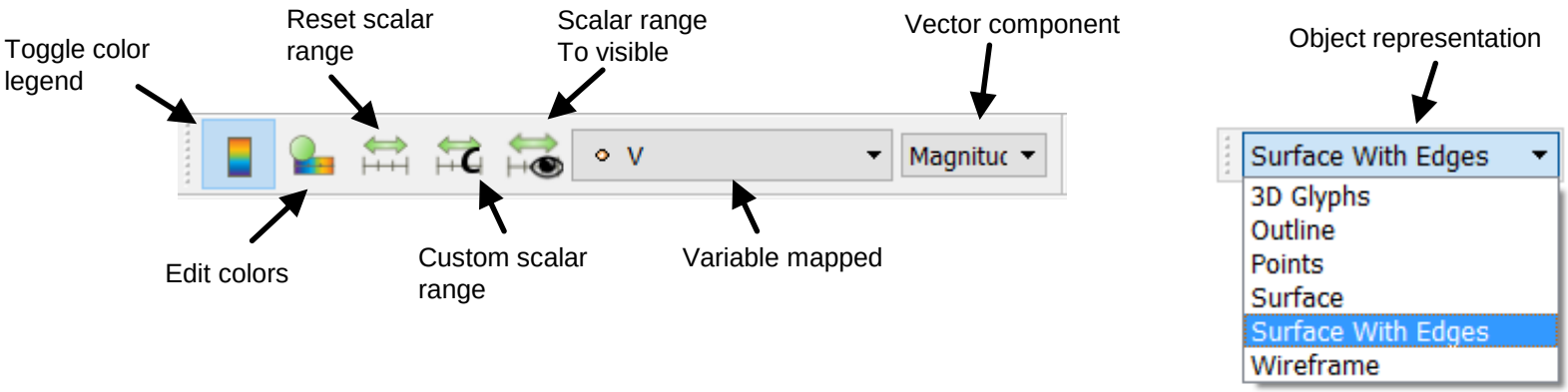
- From the `Active Variables Controls` toolbar select `p` (pressure).
- From the `Representation` toolbar select `Surface`.



Select only the mesh parts associated to the airplane

Data representation and field coloring

- Before continuing, let us take a quick look at the `Active Variables Controls` and `Representation` toolbars.

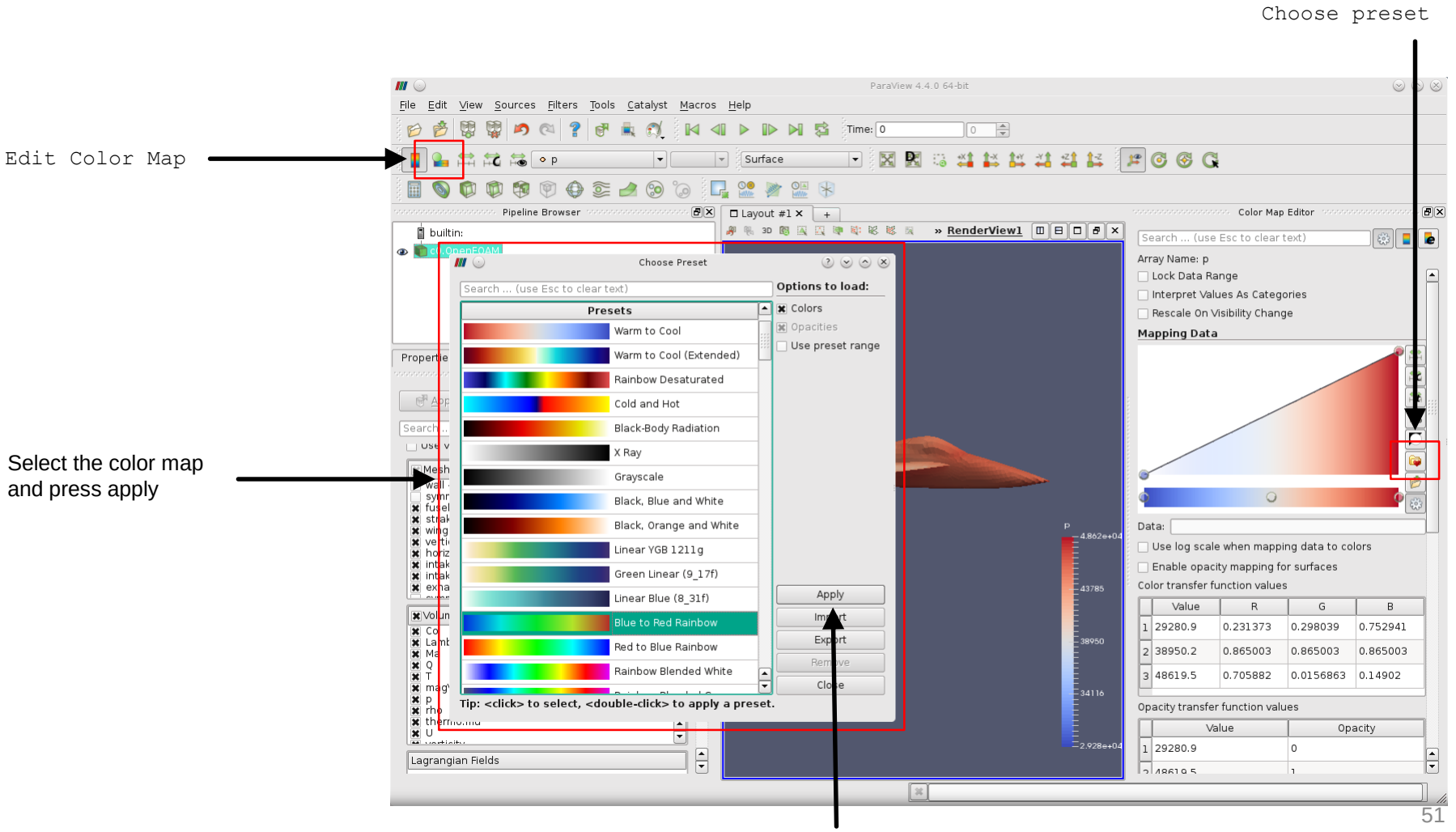


- In the variables drop-down menu, the symbol ◦ means that the variable is plot in the nodes.
- In the variables drop-down menu, the symbol means that the variable is plot in the cell center.

Scientific visualization with paraFoam/ParaView

Data representation and field coloring

- If you want to change the color map, click the `Edit Color Map` button, then `Choose preset` and select the color map of your preference.

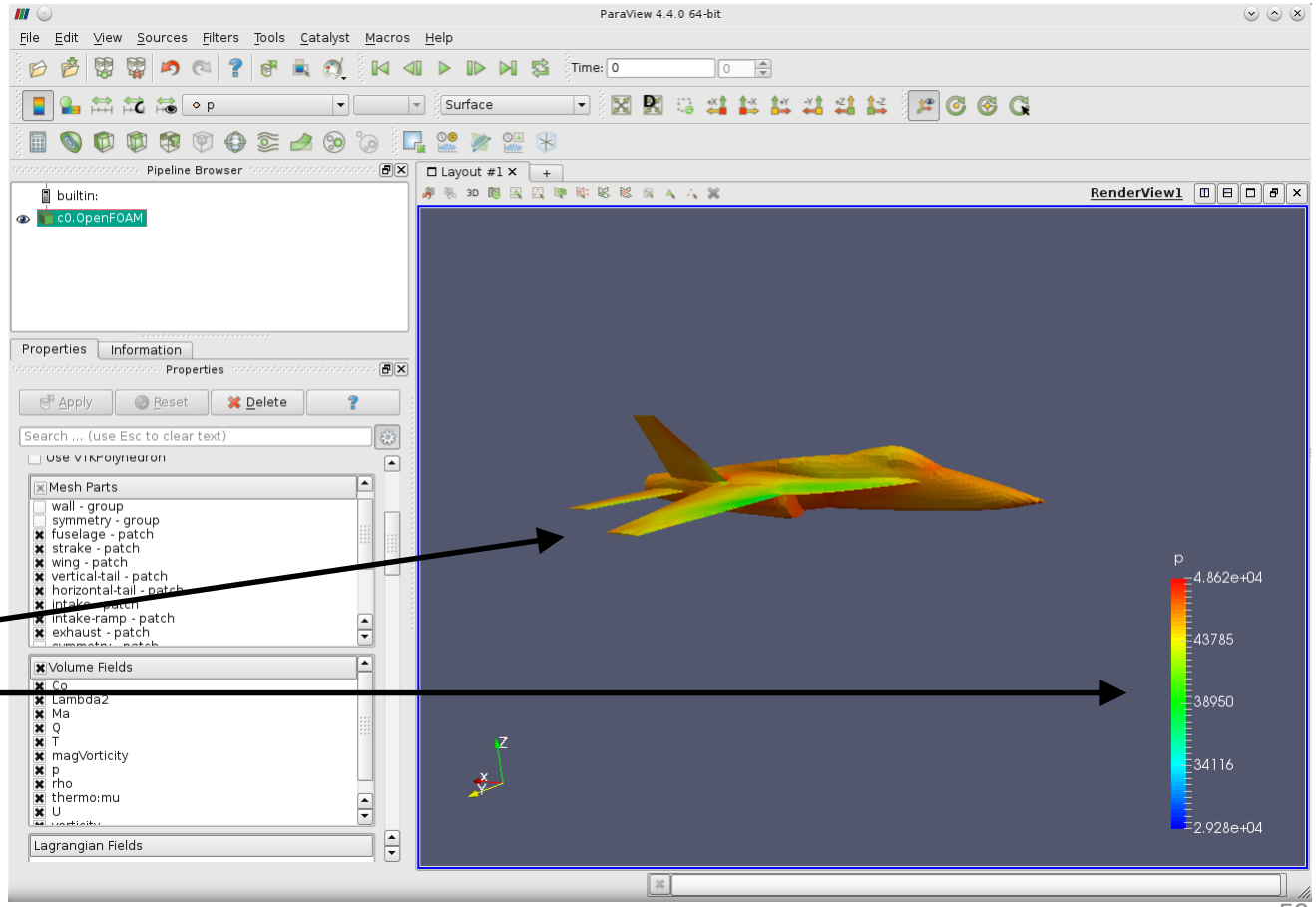


Scientific visualization with paraFoam/ParaView

Data representation and field coloring

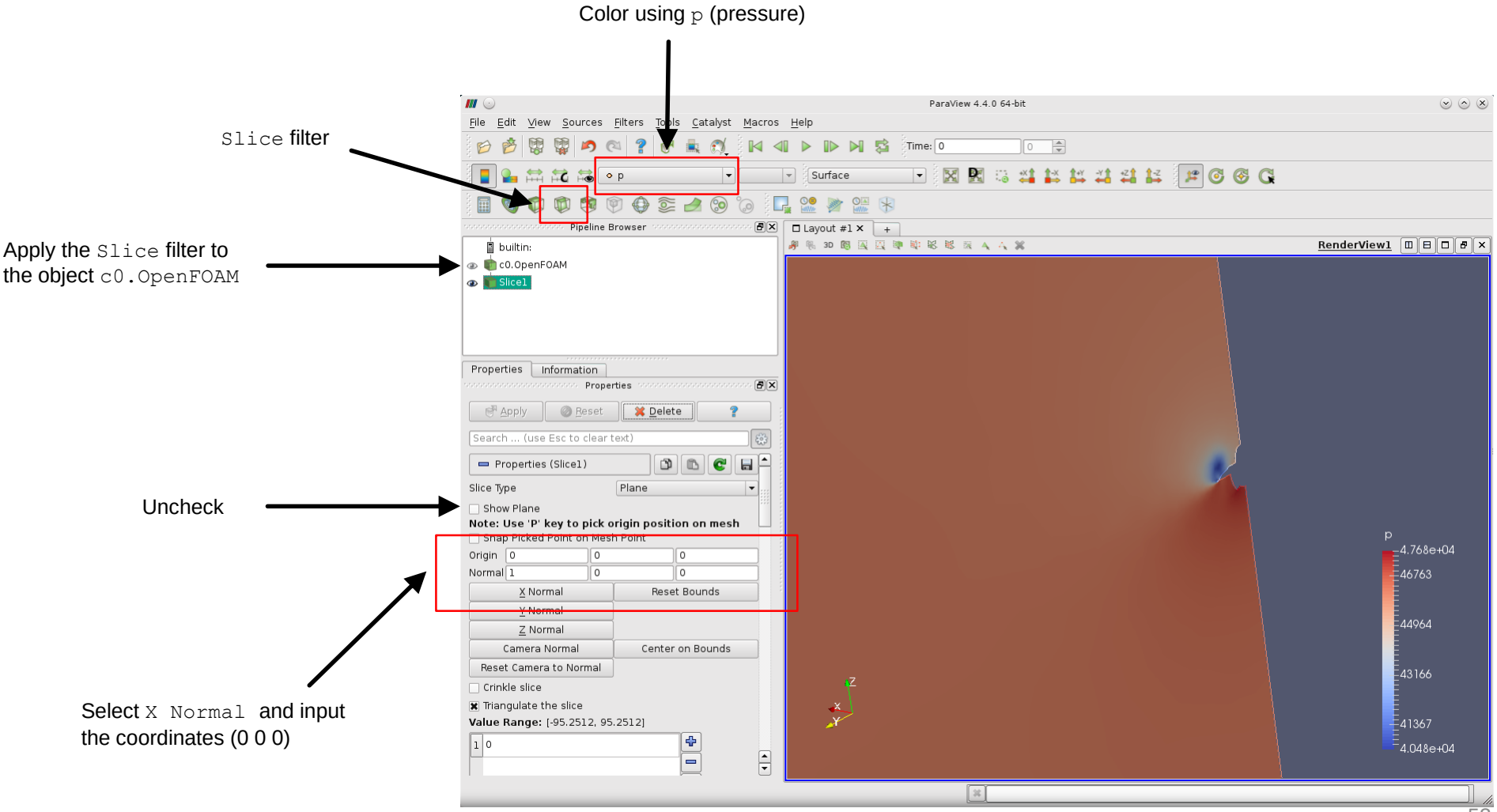
- If you want to change the color map, click the `Edit Color Map` button, then `Choose preset` and select the color map of your preference.

New color map



Adding a filter

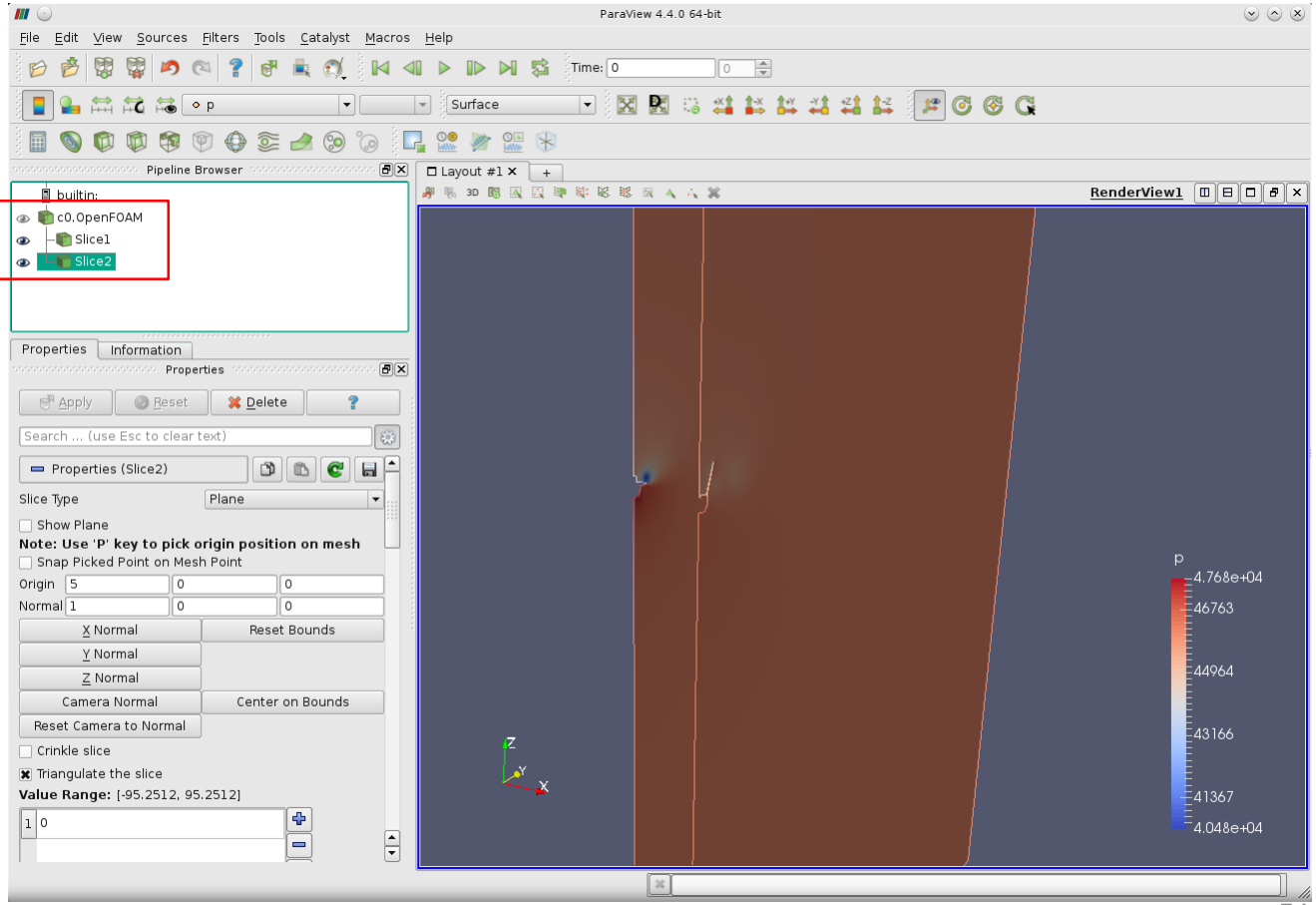
- From the menu `Filters → Alphabetical` or the toolbar, select the `Slice` filter 



Scientific visualization with paraFoam/ParaView

Adding a filter

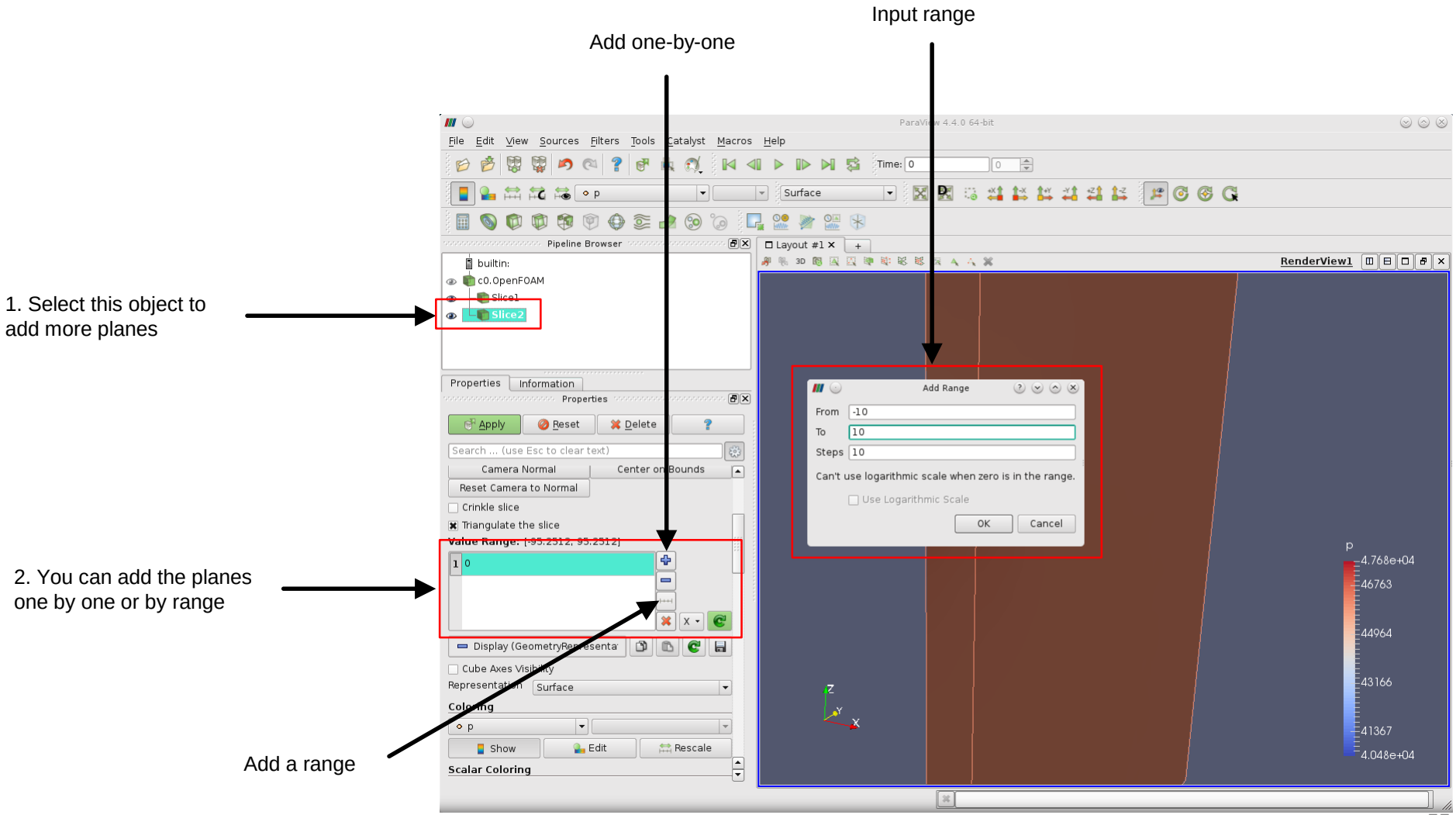
- From the menu `Filters → Alphabetical` or the toolbar, select the `Slice` filter 



- You can apply more than one filter to an object in the pipeline.
- In this case we are using two `Slice` filters in the same object.
- To apply the second slice filter you will need to select the object `c0.OpenFoam` and apply the slice filter.
- Remember, the hierarchy is extremely important.

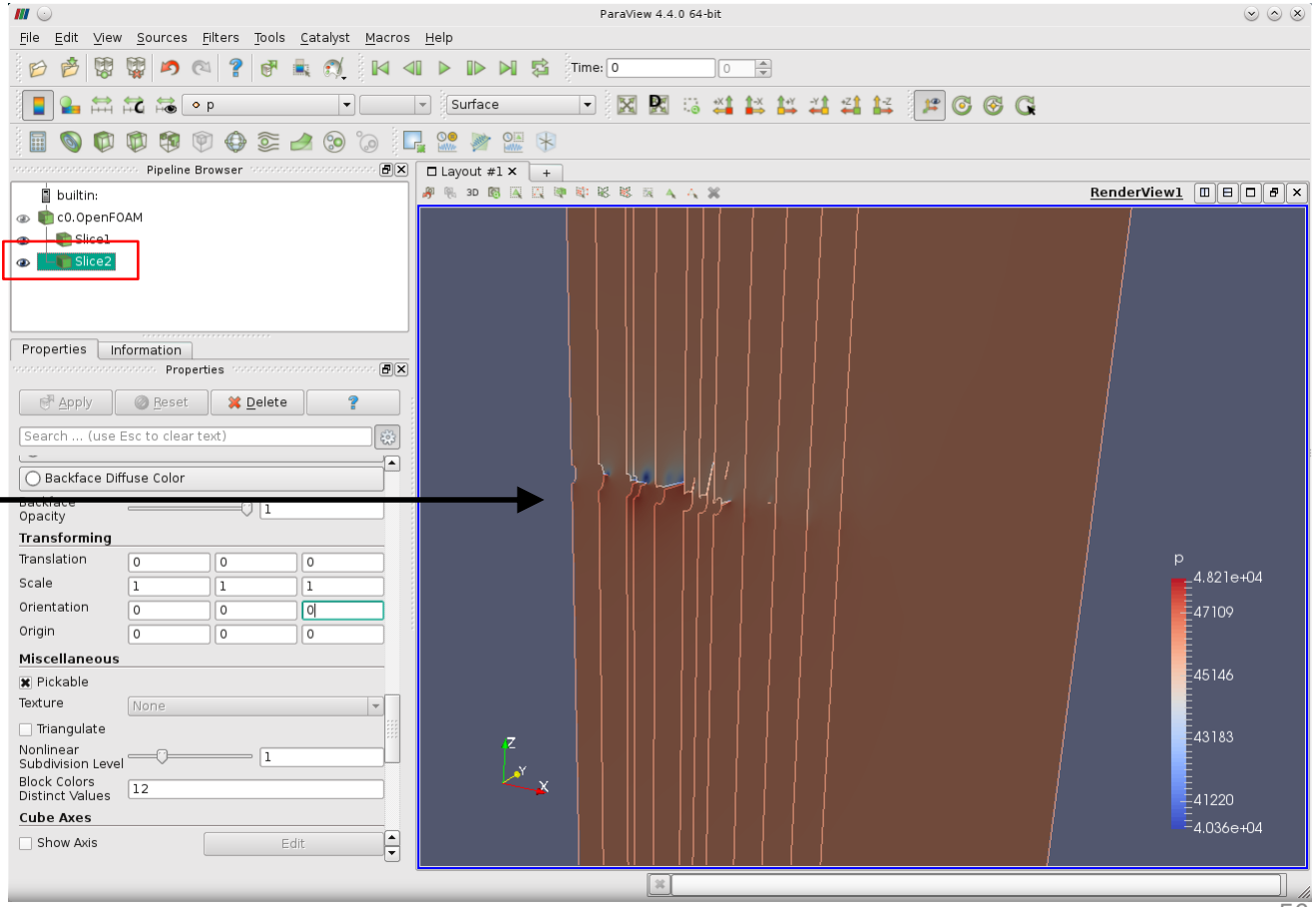
Adding a filter

- From the menu `Filters → Alphabetical` or the toolbar, select the `Slice` filter 



Adding a filter

- From the menu `Filters → Alphabetical` or the toolbar, select the `Slice` filter 



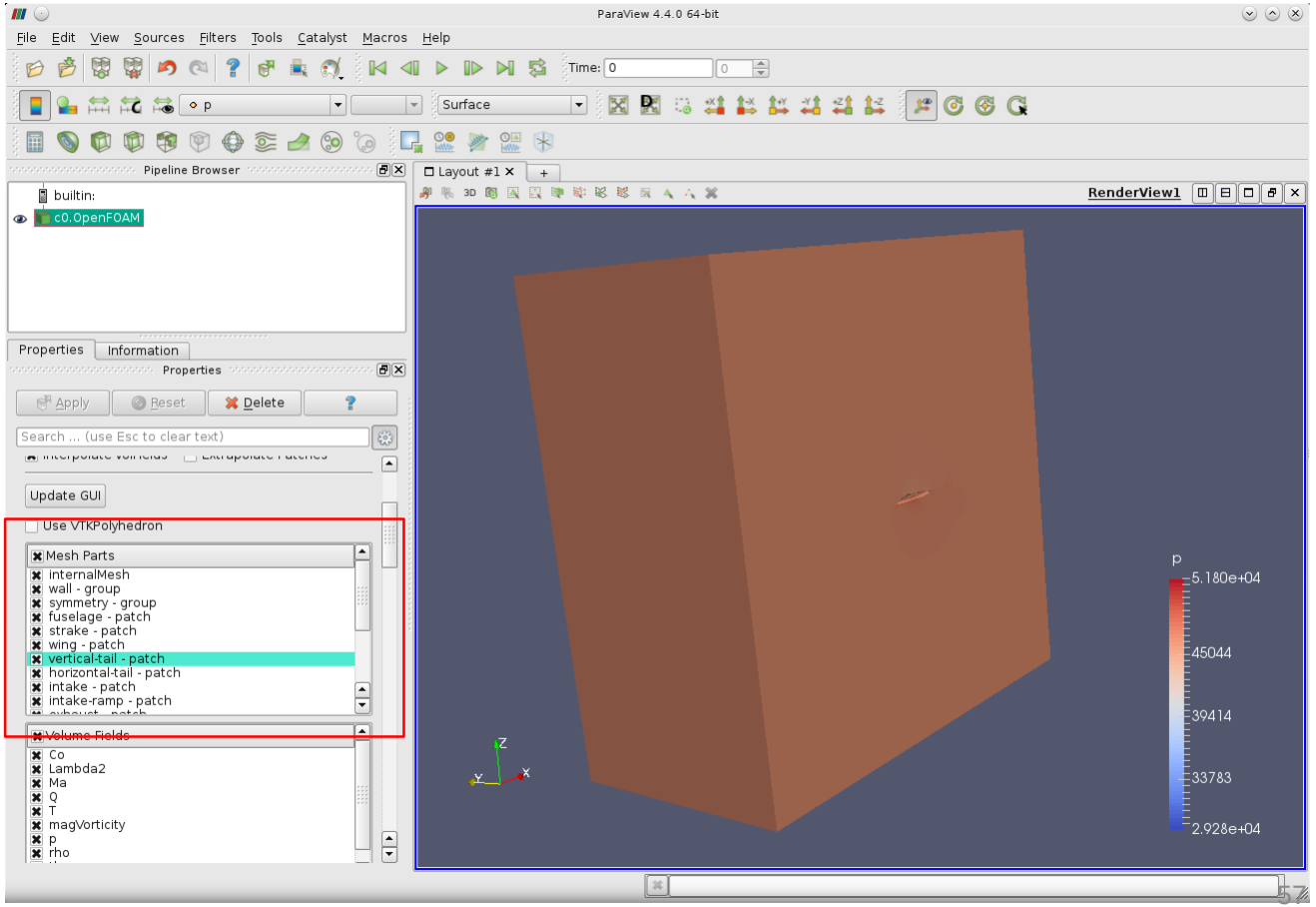
We added 10 planes to the `Slice2` object

Scientific visualization with paraFoam/ParaView

Creating a visualization pipeline

- We will use the filter `Extract Block` to create an object containing the mesh part `internalMesh` and a second object containing the mesh parts associated to the airplane geometry.
- This is particularly helpful if you want to apply a filter to the volume mesh and visualize the airplane surface.
- You can also load a second case, but this takes more memory.

Select all mesh parts →



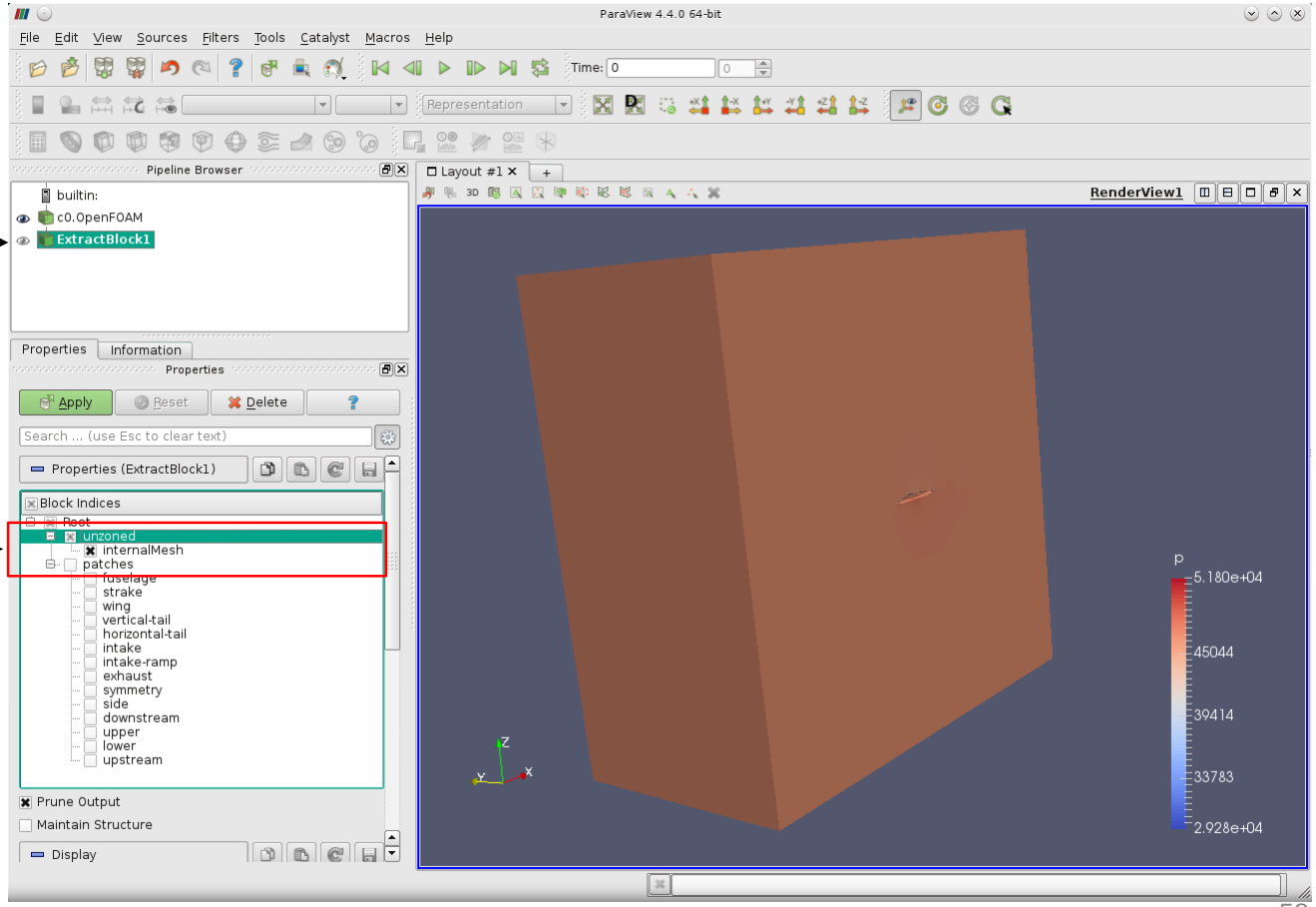
Scientific visualization with paraFoam/ParaView

Creating a visualization pipeline

- Select the filter `Extract Block`, from the menu `Filters → Alphabetical`.
- We will use this filter to create the object `ExtractBlock1` containing the mesh part `internalMesh`.

New object →

Select `internalMesh` →

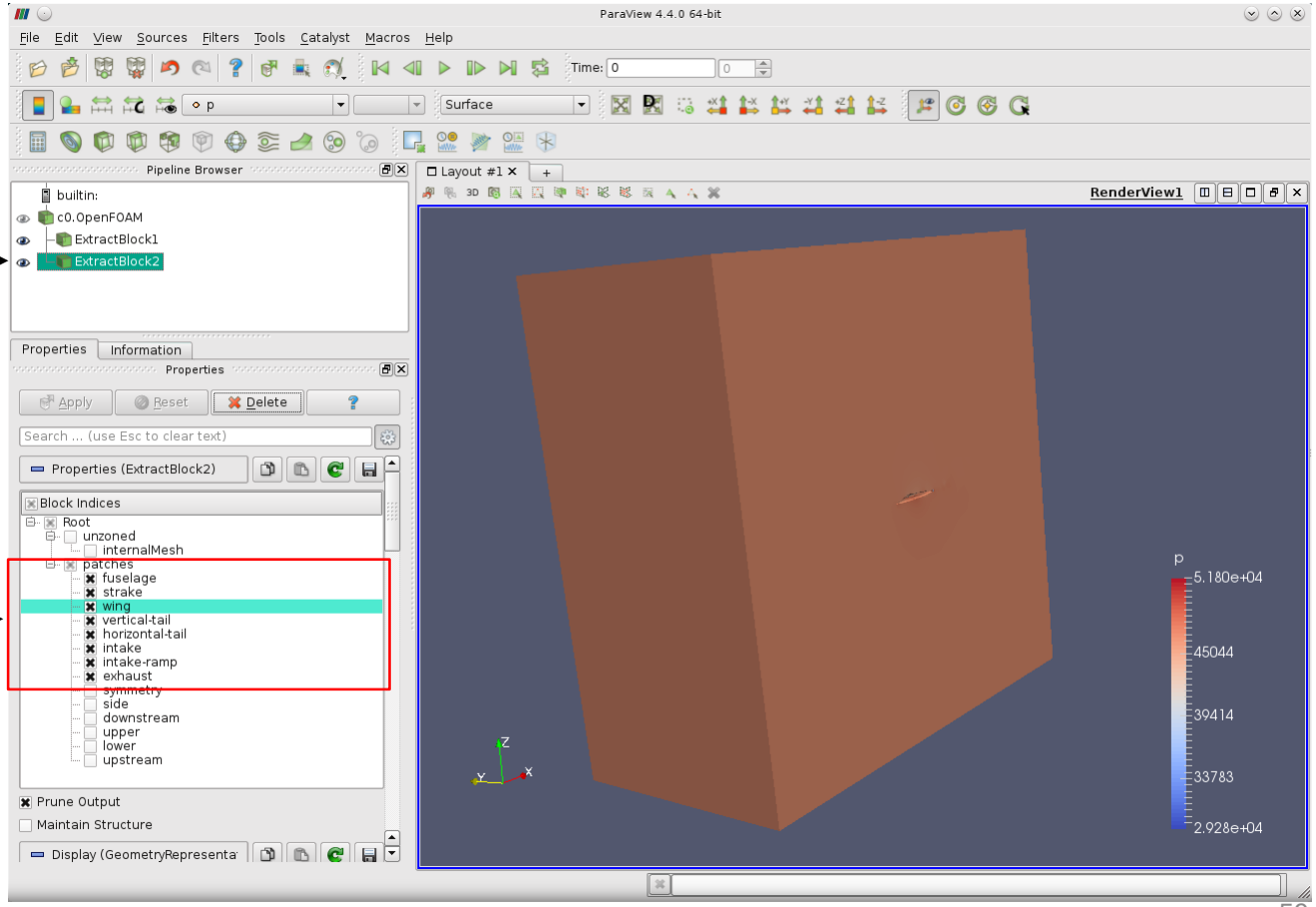


Creating a visualization pipeline

- Select the object `c0.OpenFOAM`, and from the menu `Filters → Alphabetical` select the filter `Extract Block`.
- We will use this filter to create the object `ExtractBlock2` containing the mesh parts associated to the airplane geometry.

New object

Select all the airplane parts

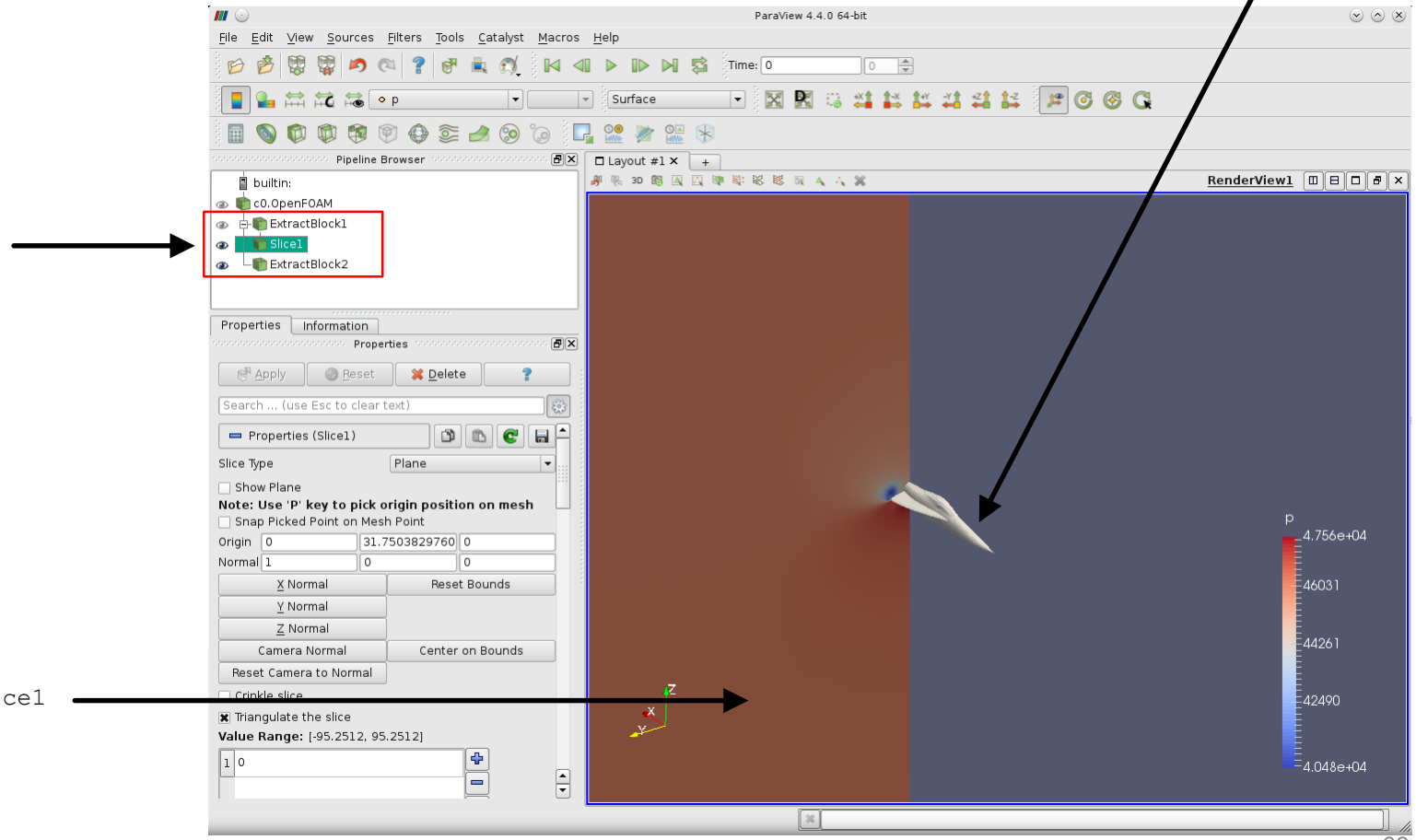


Scientific visualization with paraFoam/ParaView

Creating a visualization pipeline

- Create a `Slice` using the object `ExtractBlock1` and color it by pressure field (`p`).
- Visualize the object `ExtractBlock2` using solid color.
- Previously, we were not able to plot both objects at the same time.

`ExtractBlock2`



`Slice1`

Scientific visualization with paraFoam/ParaView

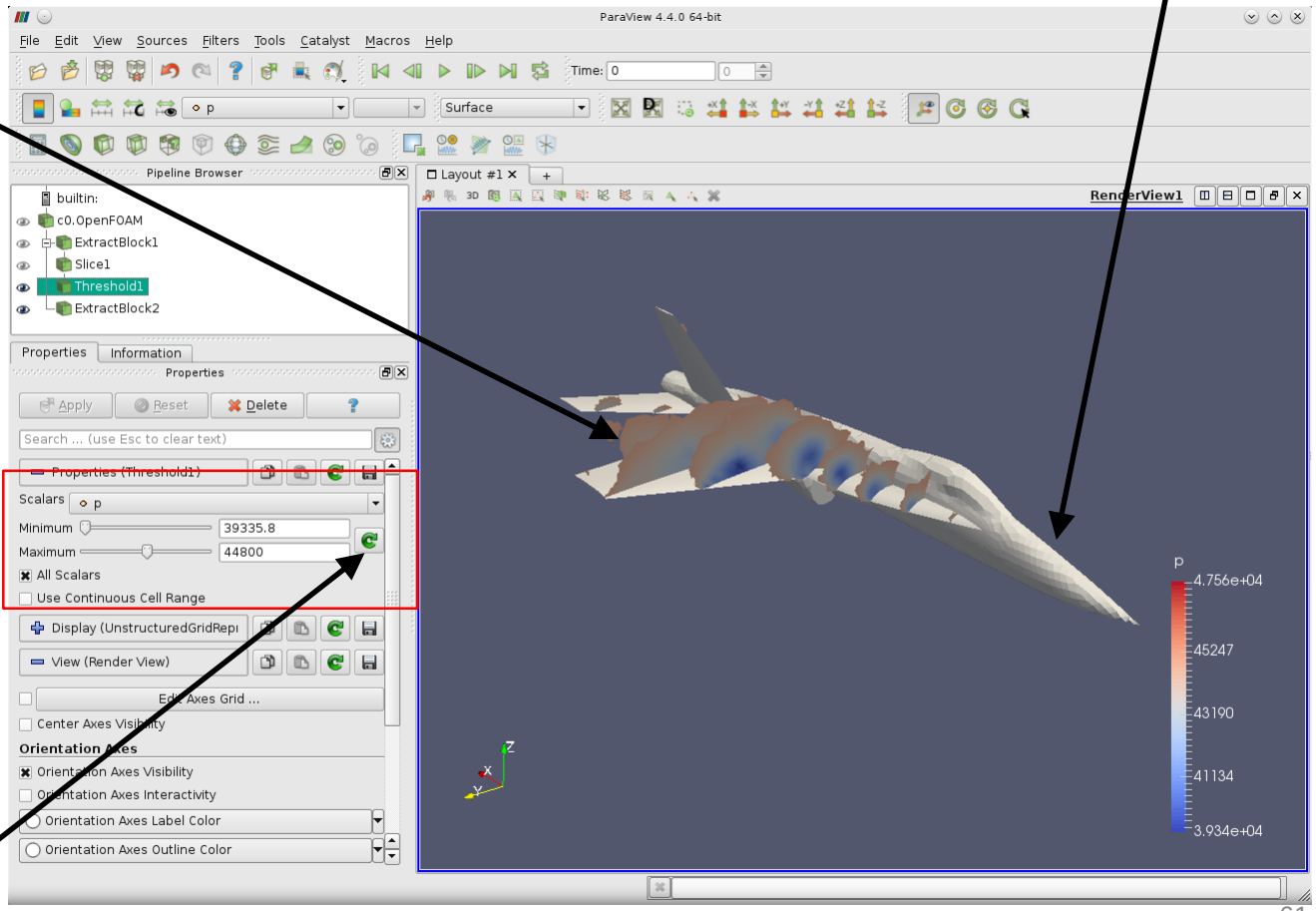
Creating a visualization pipeline

- Select the object `Slice1`, and from the menu `Filters → Alphabetical` or the toolbar, select the `Threshold` filter 

`ExtractBlock2`

`Threshold1`

- Select the input variable (`p`) and set the desired range.
- This filter will cut-off the values that are outside the selected range.



Reset range

Scientific visualization with paraFoam/ParaView

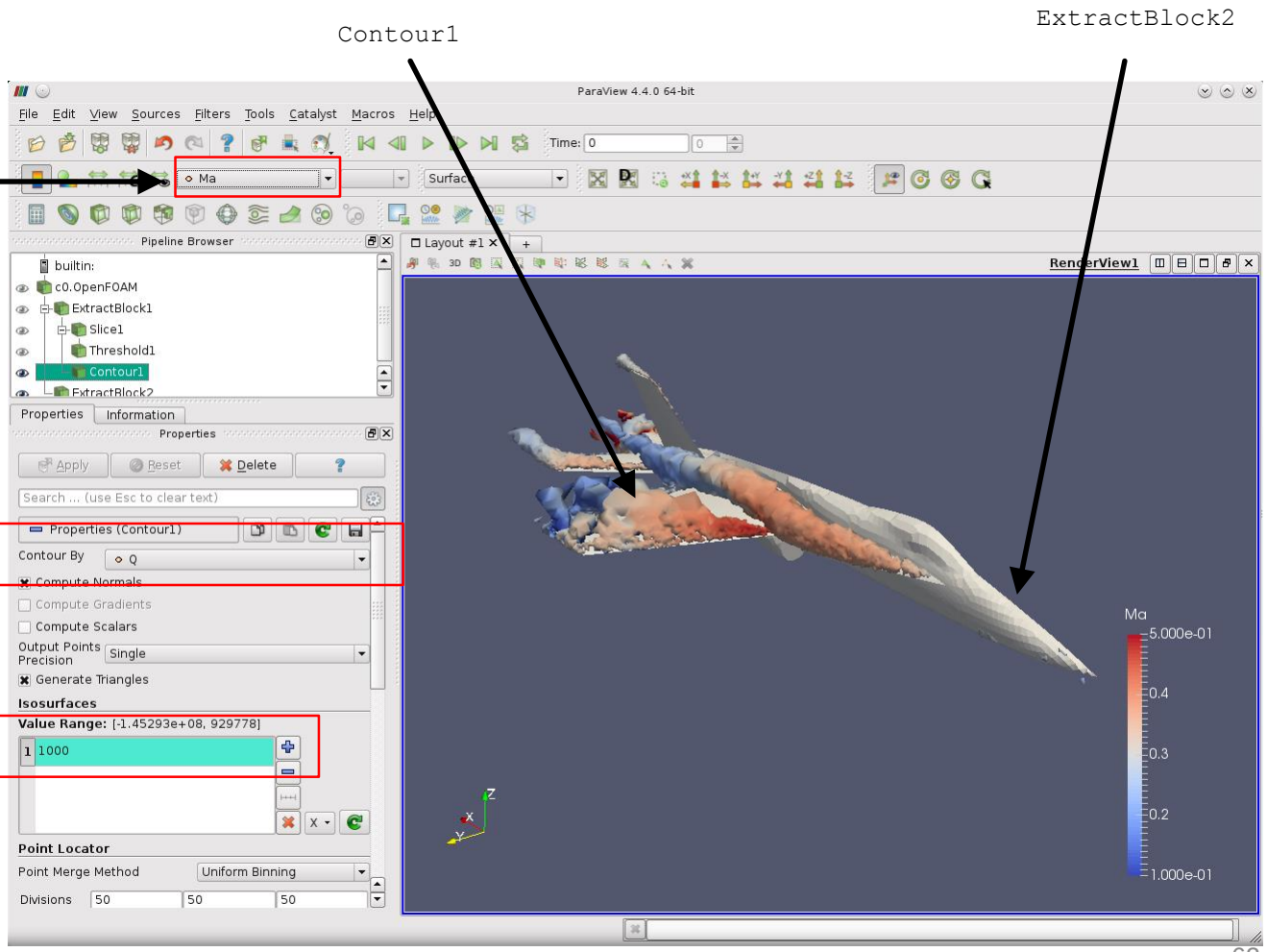
Creating a visualization pipeline

- Select the object `ExtractBlock1`, and from the menu `Filters → Alphabetical` or the toolbar, select the `Contour` filter 

Color the iso-surfaces using `Ma` (Mach number)

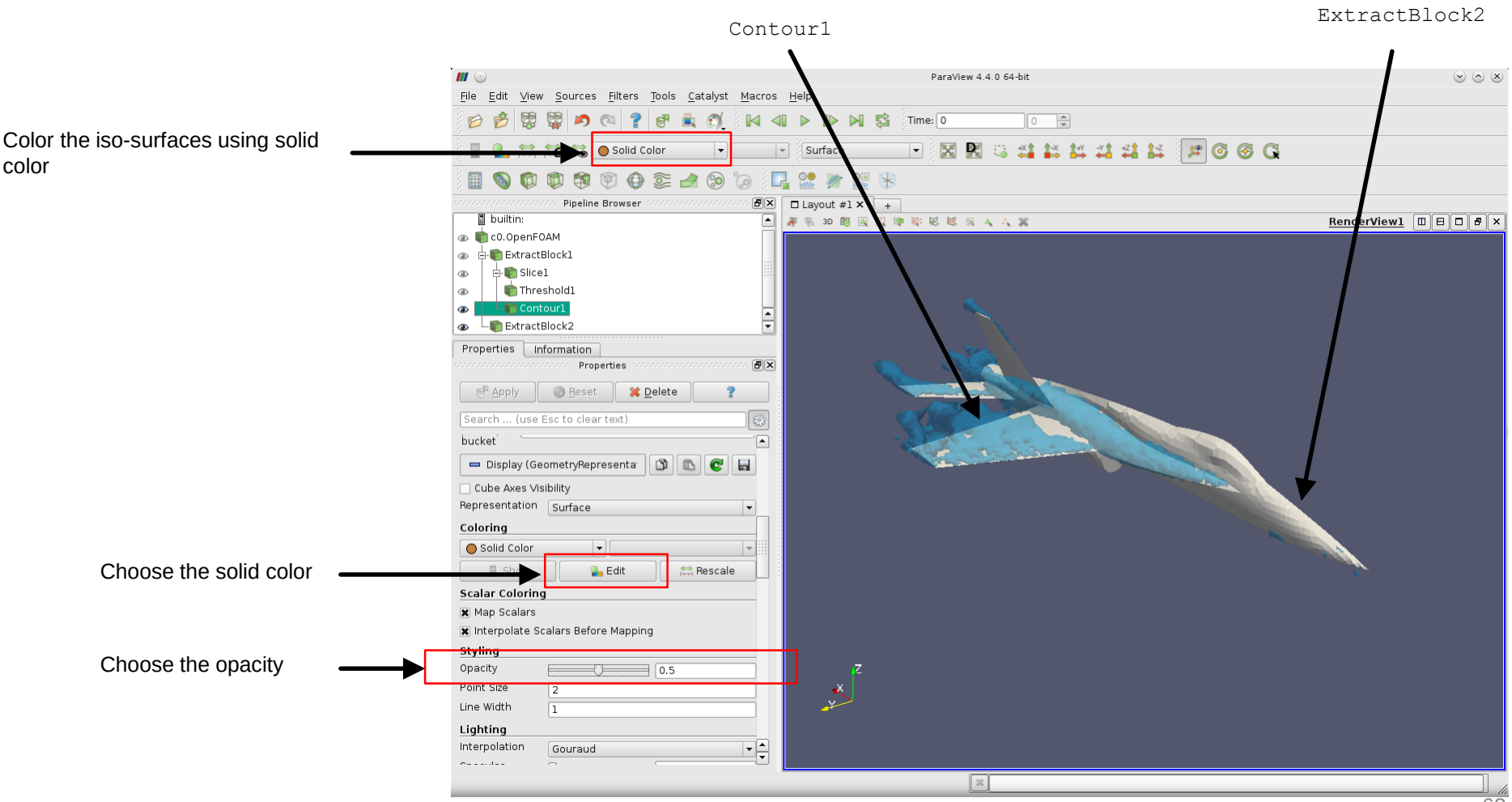
- Select `Q` (Q-criterion).
- This variable is used to capture the vortices.

- To capture the vortices, the input value must be positive.
- In this case, use a value of 1000.



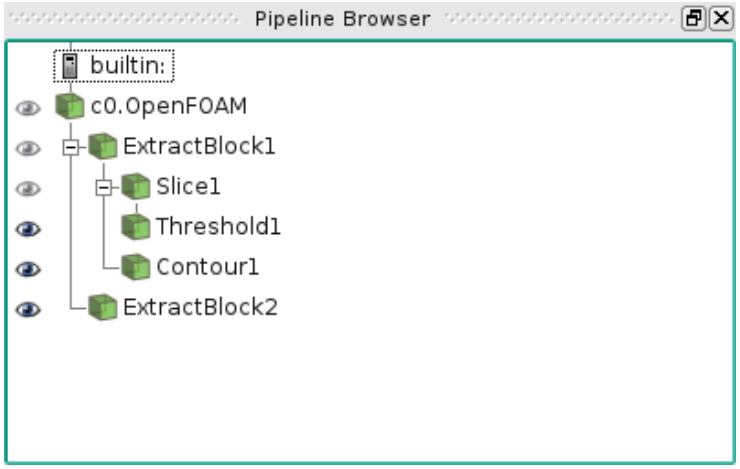
Creating a visualization pipeline

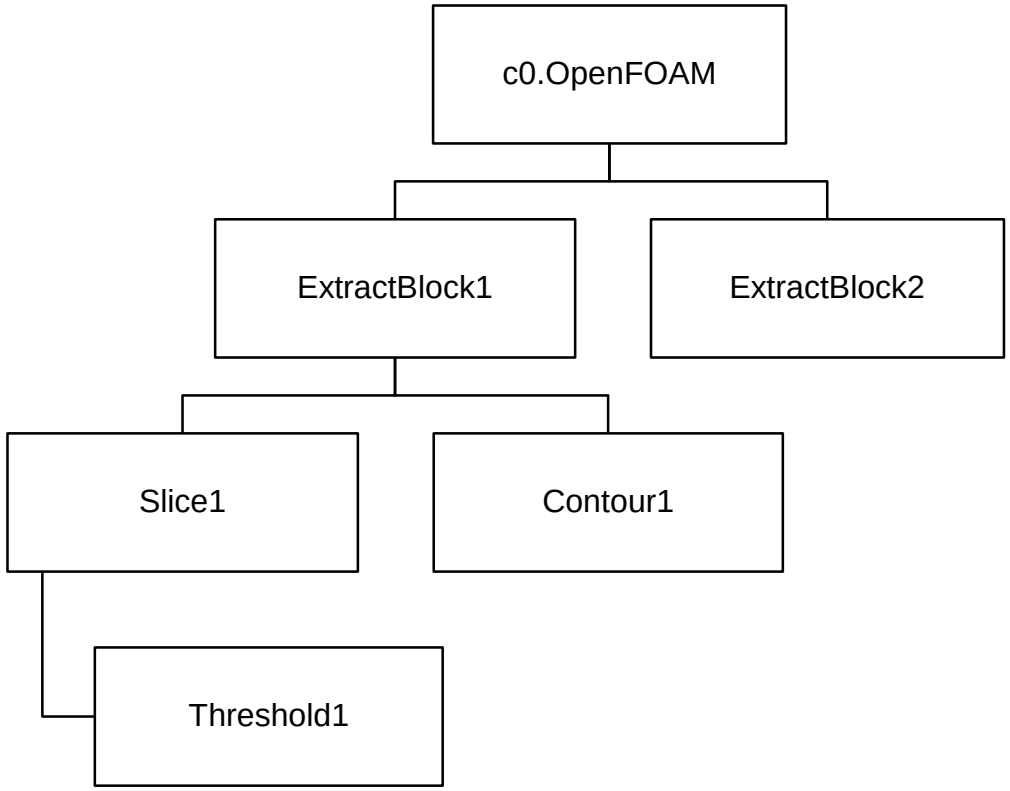
- From the menu `Filters → Alphabetical` or the toolbar, select the `Contour` filter 
- Apply this filter to the object `ExtractBlock1`.



Creating a visualization pipeline

- At this point take a look at the pipeline browser and study the hierarchy.

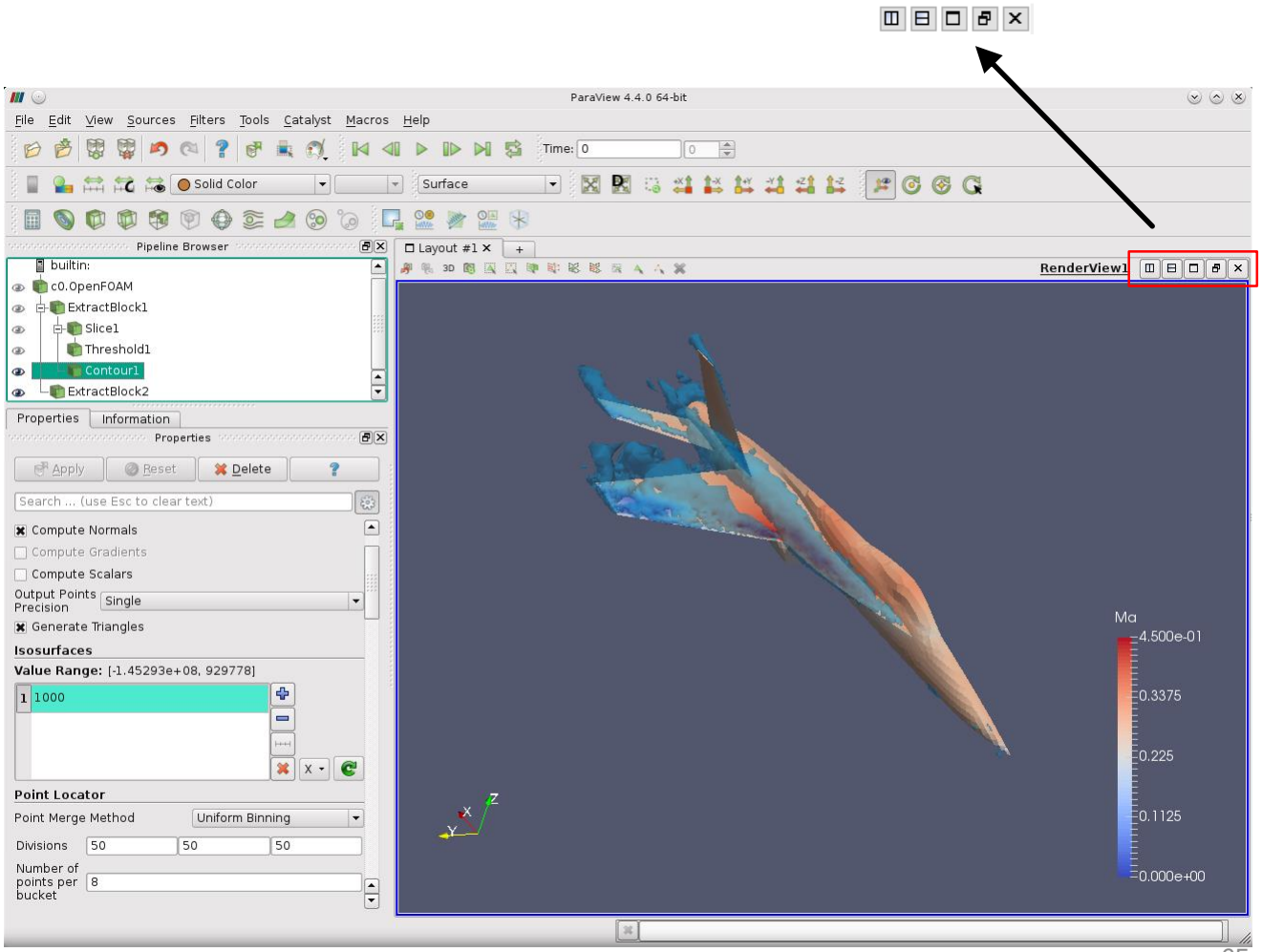




Scientific visualization with paraFoam/ParaView

Multiview

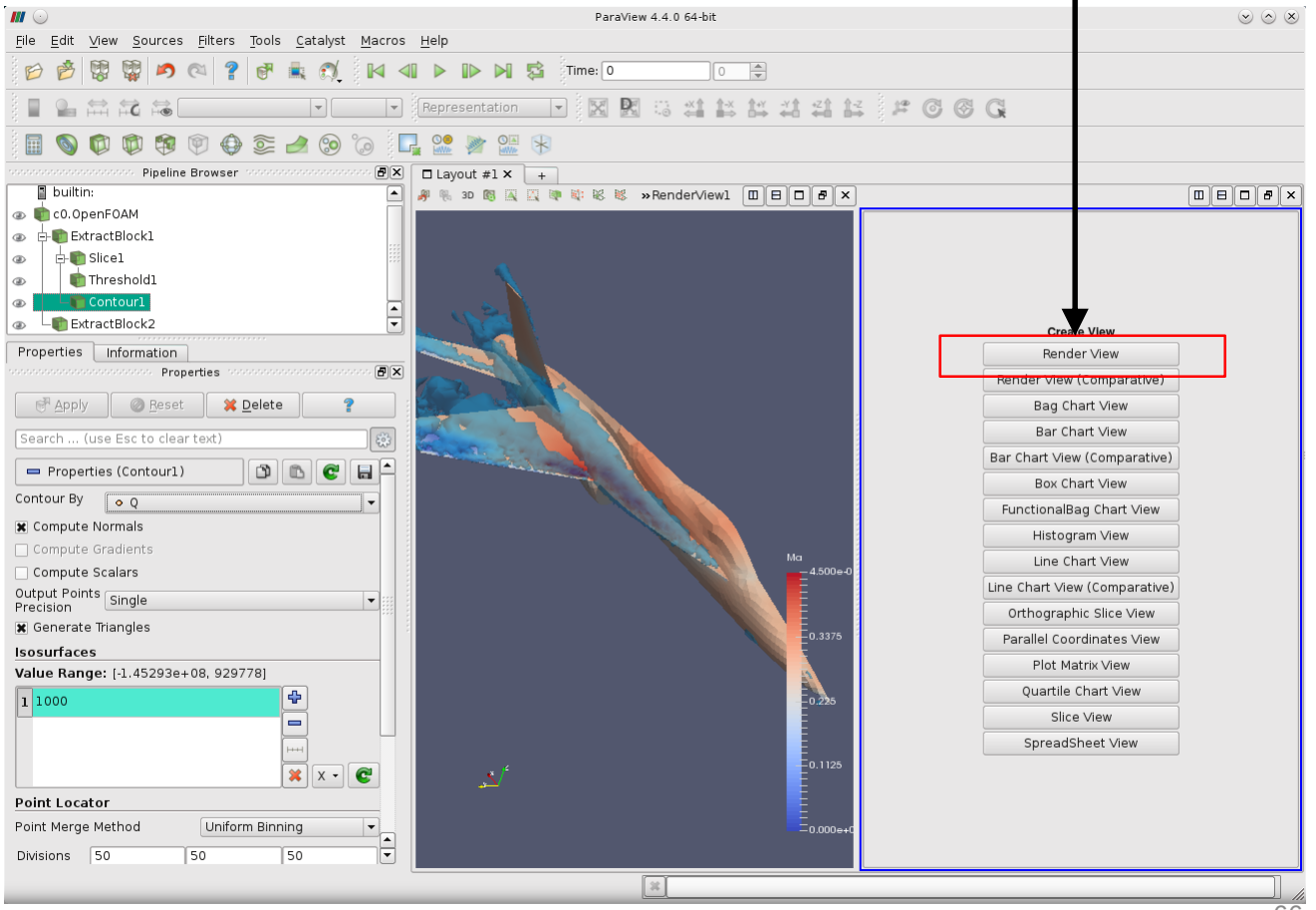
- Click in the new view icon ◫ to create a new 3D view. This will split the current view in two vertical views.



Scientific visualization with paraFoam/ParaView

Multiview

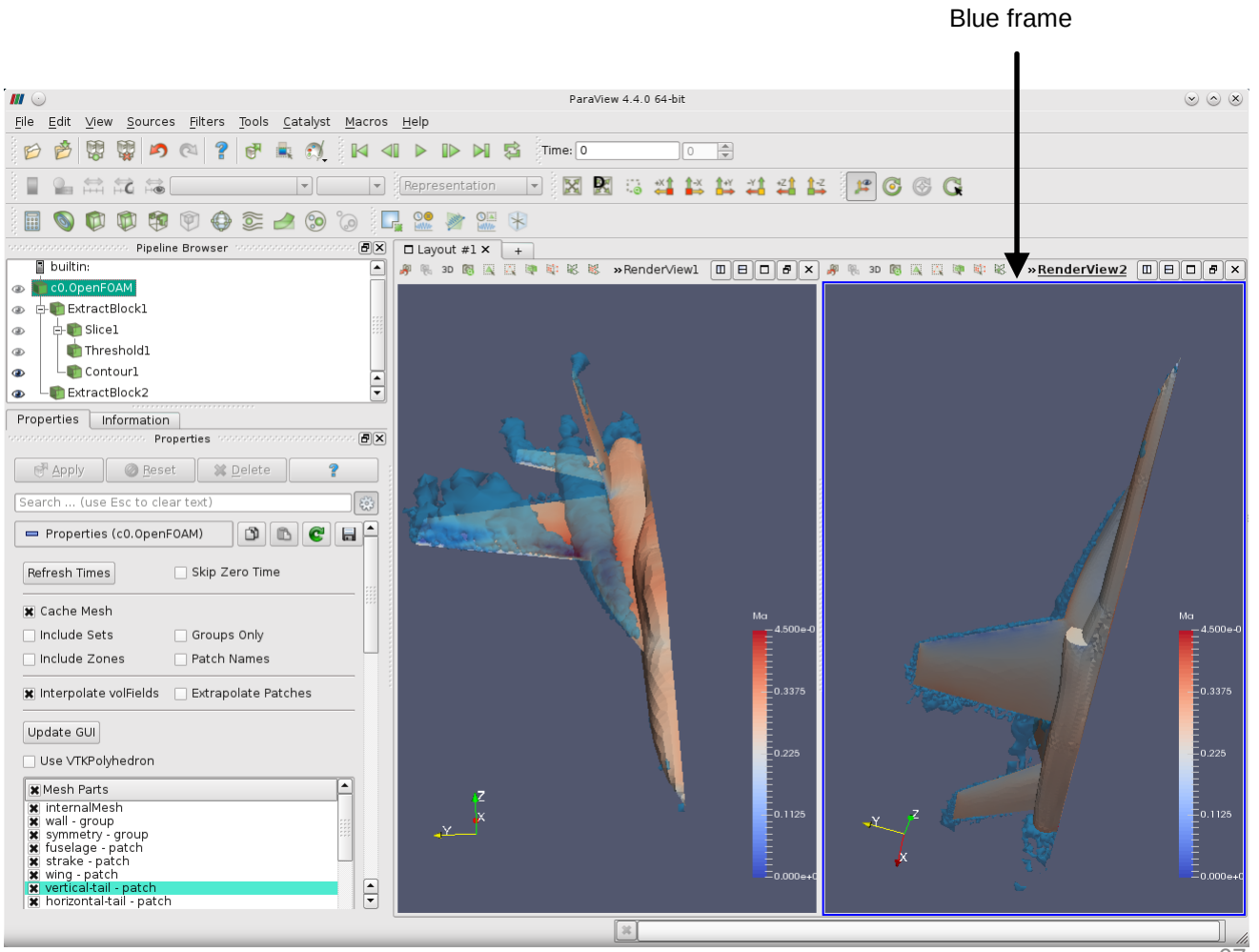
- Select the option `Render View`.
- You can create as many 3D views as you like.



Scientific visualization with paraFoam/ParaView

Multiview

- From the pipeline browser select what you want to render in the new view.
- The blue frame around the 3D view, indicates that this is the active view.



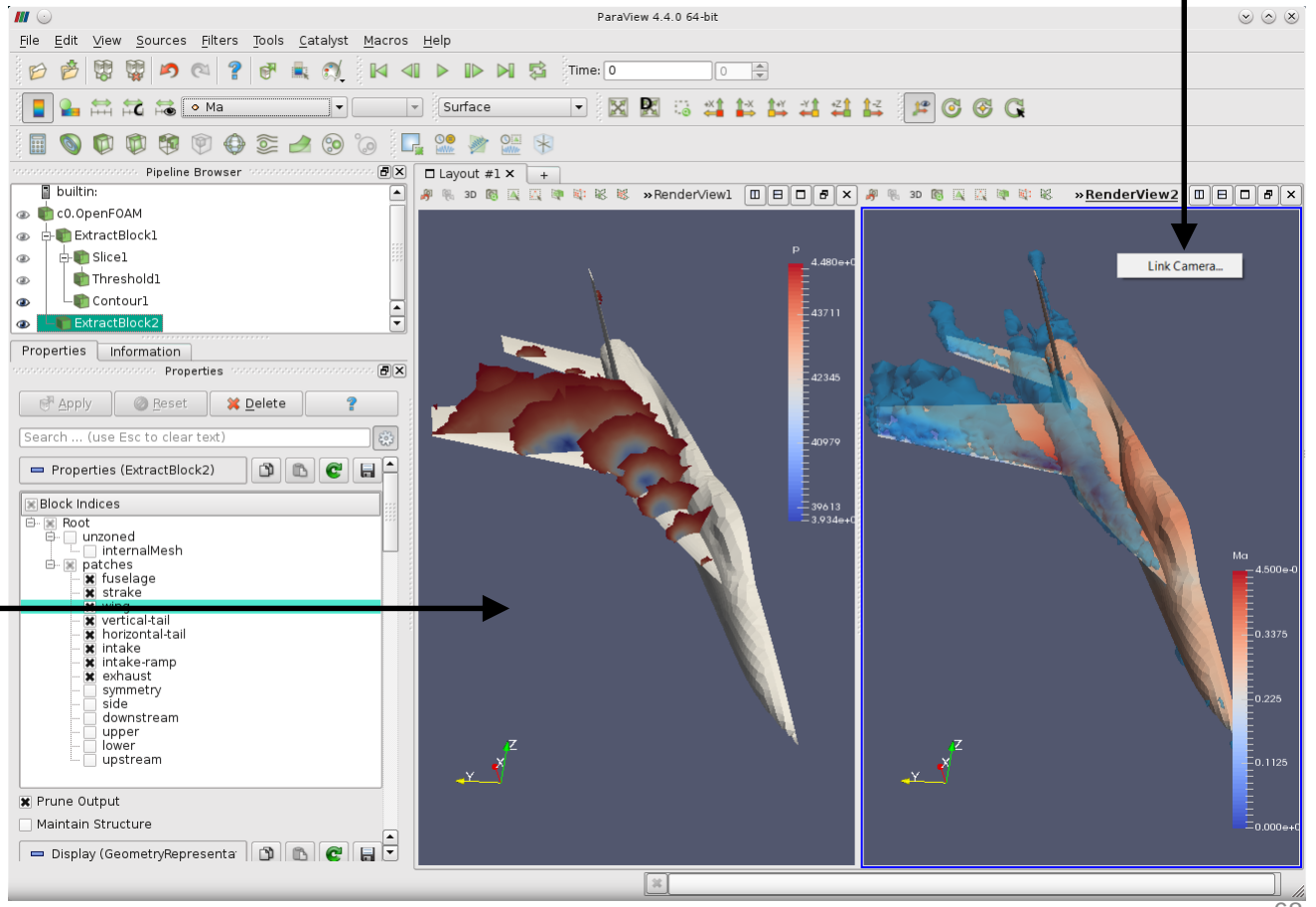
Scientific visualization with paraFoam/ParaView

Multiview

- If you want to link the cameras, right click in the 3D view, select `Link Camera`, and click another view to link with.
- To unlink views, select the menu `Tools → Manage Links`, then remove the link.

1. Select link camera

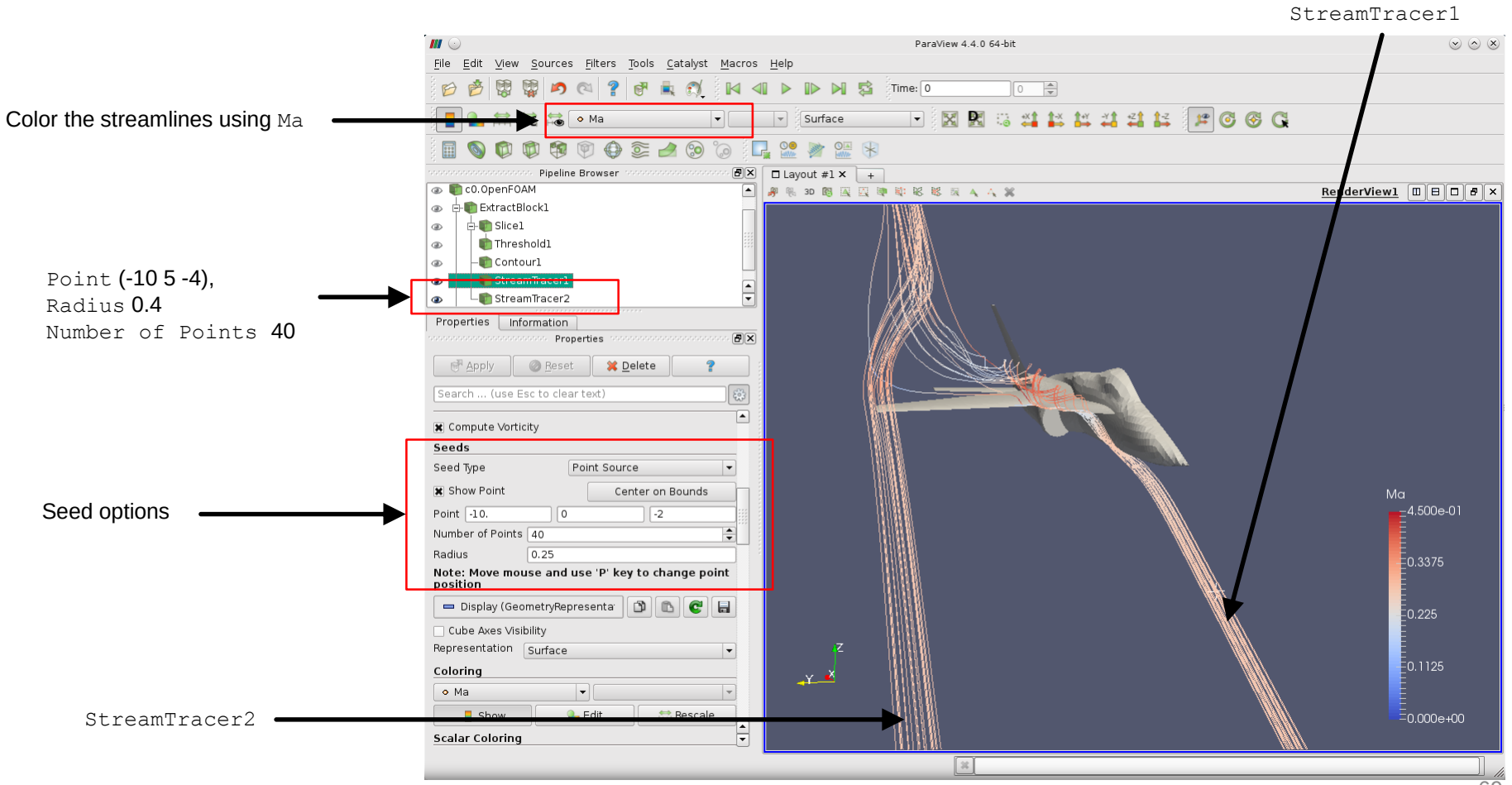
2. Click in this view to link cameras



Scientific visualization with paraFoam/ParaView

Streamlines

- In the pipeline browser select the object `ExtractBlock1` and apply the filter `Stream Tracer`
- In the properties panel, select the variable `U` for `Vectors`, set `Seed Type` to `Point Source`, set the coordinates of the seed `Point` to (-10 0 -2), set the `Radius` to 0.25 and set the `Number of Points` to 40.



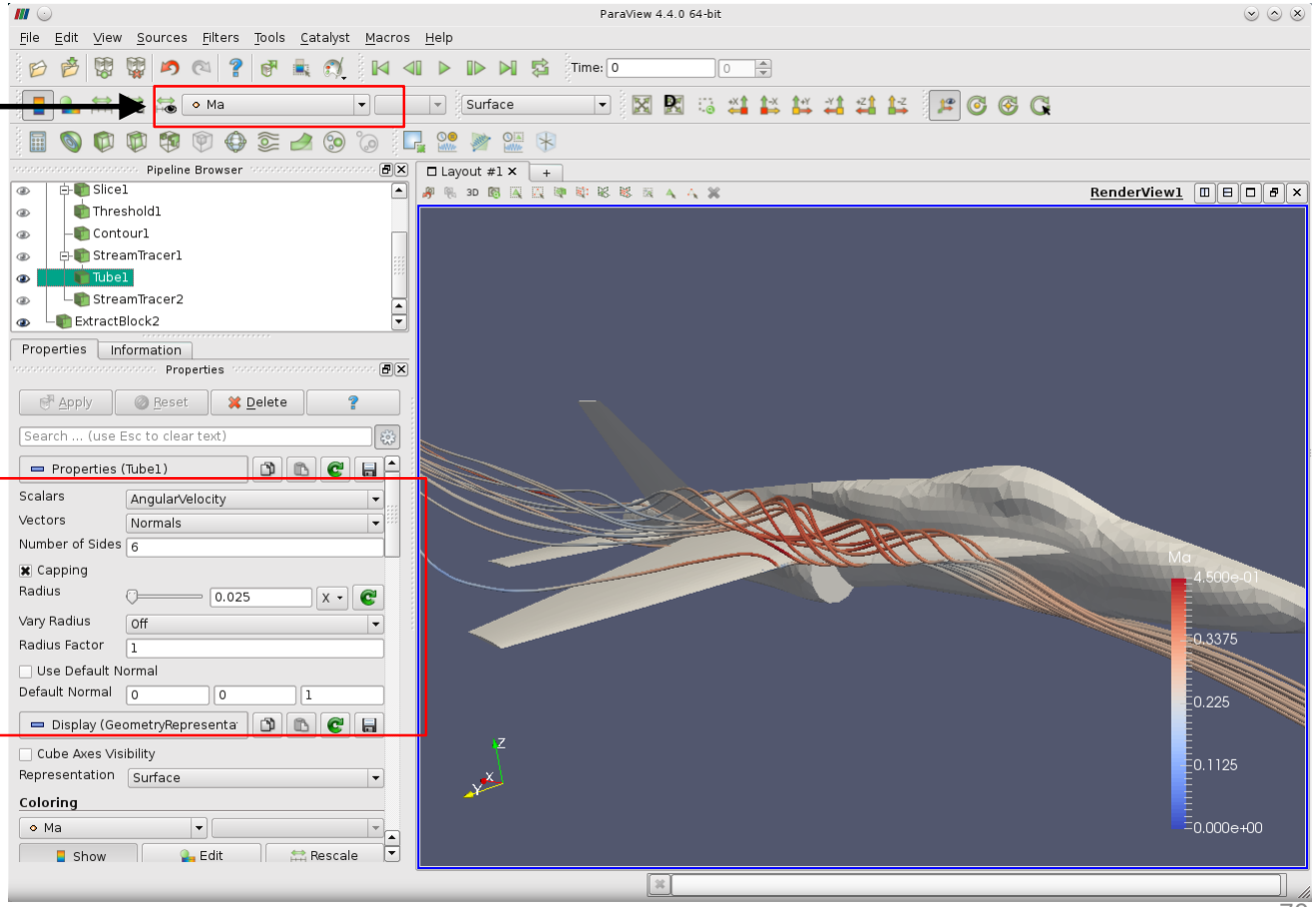
Scientific visualization with paraFoam/ParaView

Tube

- Select the object `StreamTracer1`.
- From the menu `Filters → Alphabetical`, select the `Tube` filter.
- Adjust the tube options to get a similar result.

Color the tubes using `Ma`

Tube options



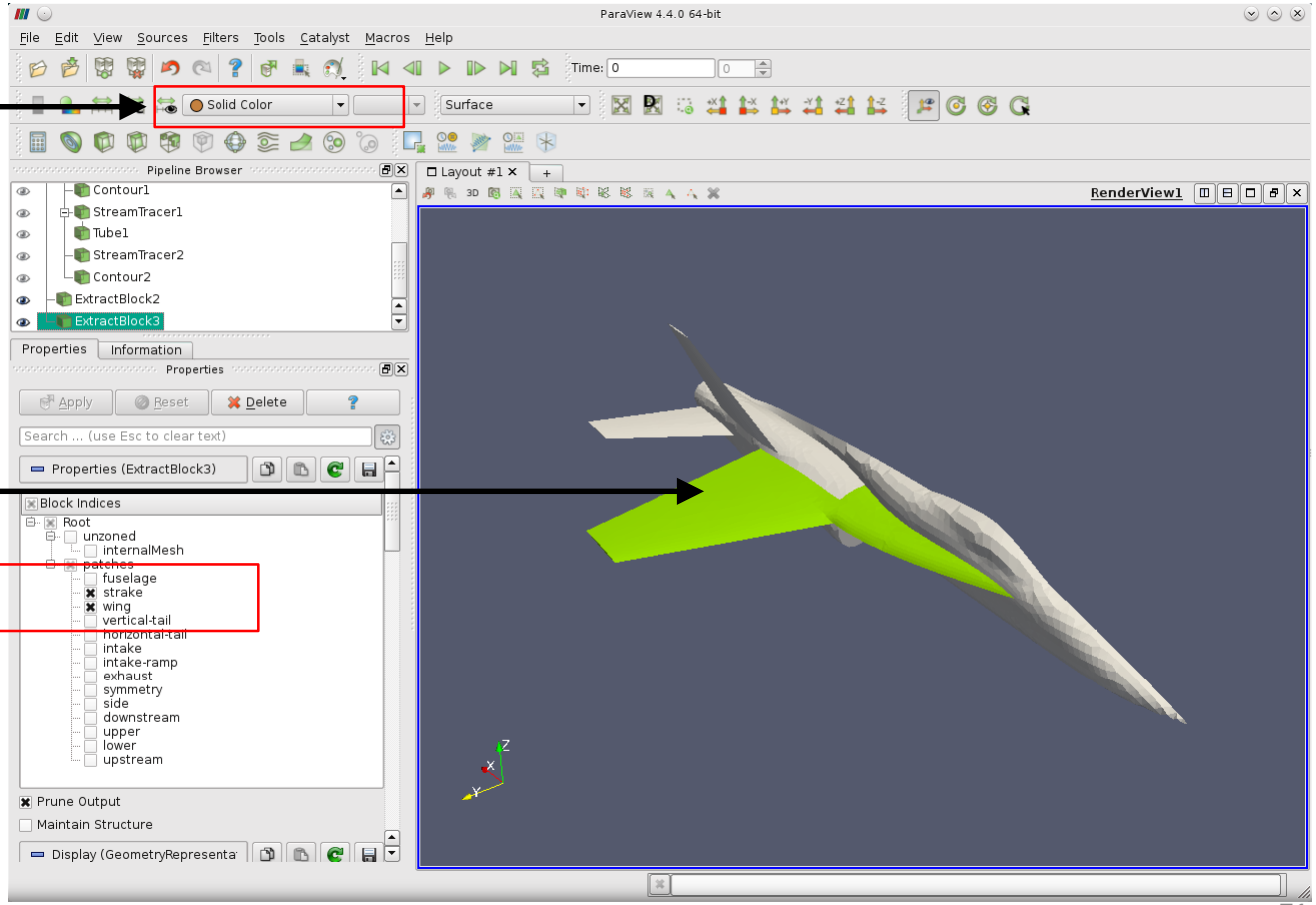
Scientific visualization with paraFoam/ParaView

Releasing streamlines from a surface

- Select the object `c0.OpenFOAM`, and from the menu `Filters → Alphabetical` select the filter `Extract Block`.
- Select the strake and wing parts and create the new object. We will release the streamlines from these surfaces.

Color the object `ExtractBlock3` by `Solid Color`

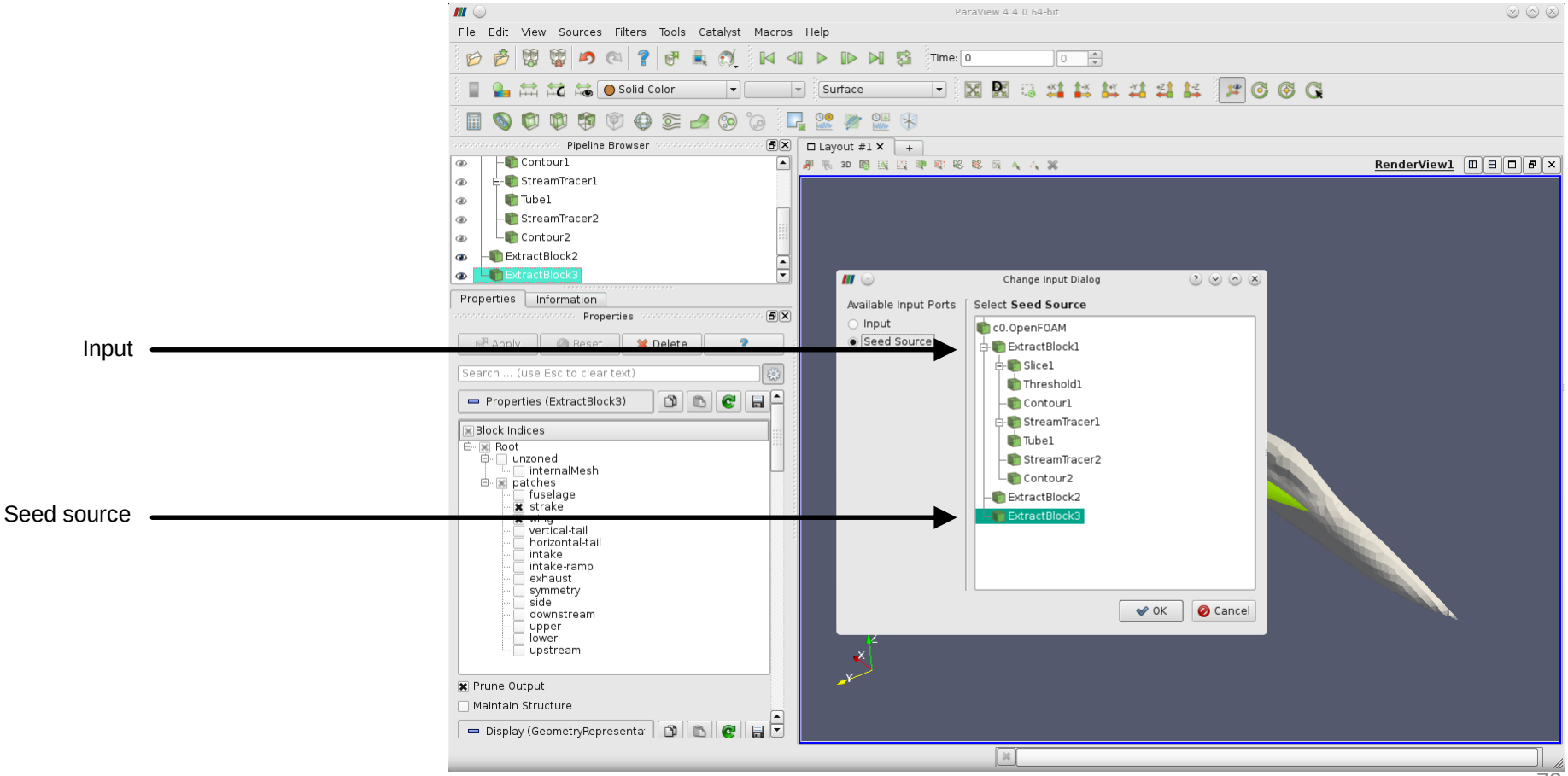
`ExtractBlock3` (in green)



Scientific visualization with paraFoam/ParaView

Releasing streamlines from a surface

- Select the filter `Stream Tracer With Custom Source`, from the menu `Filters → Alphabetical`.
- As seed source select the object `ExtractBlock3` and as input select the object `ExtractBlock1`.

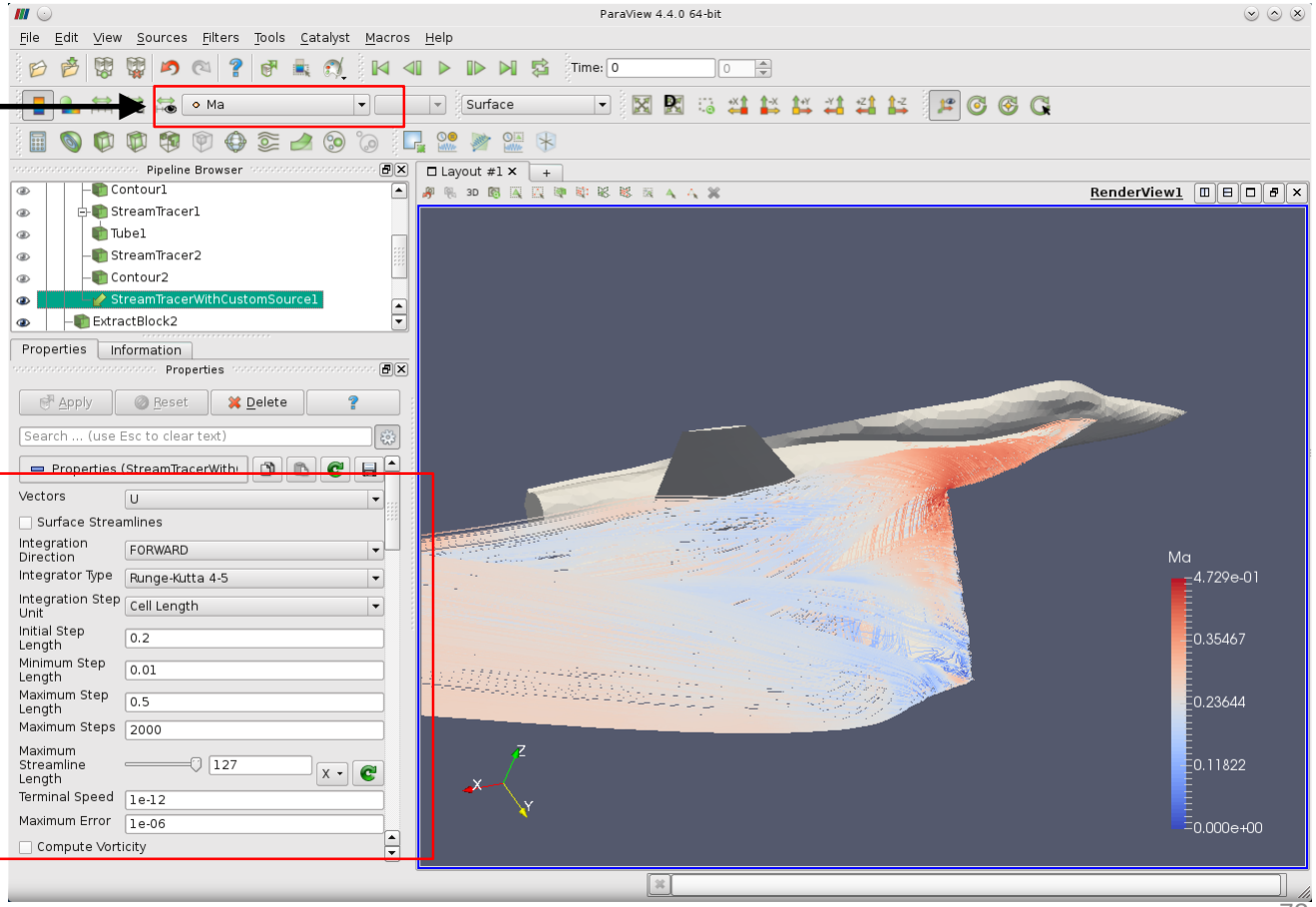


Releasing streamlines from a surface

- Depending on the number of points on the surface and the number of surfaces selected, this operation might take a long time.

Color the streamlines using `Ma`

Streamlines options



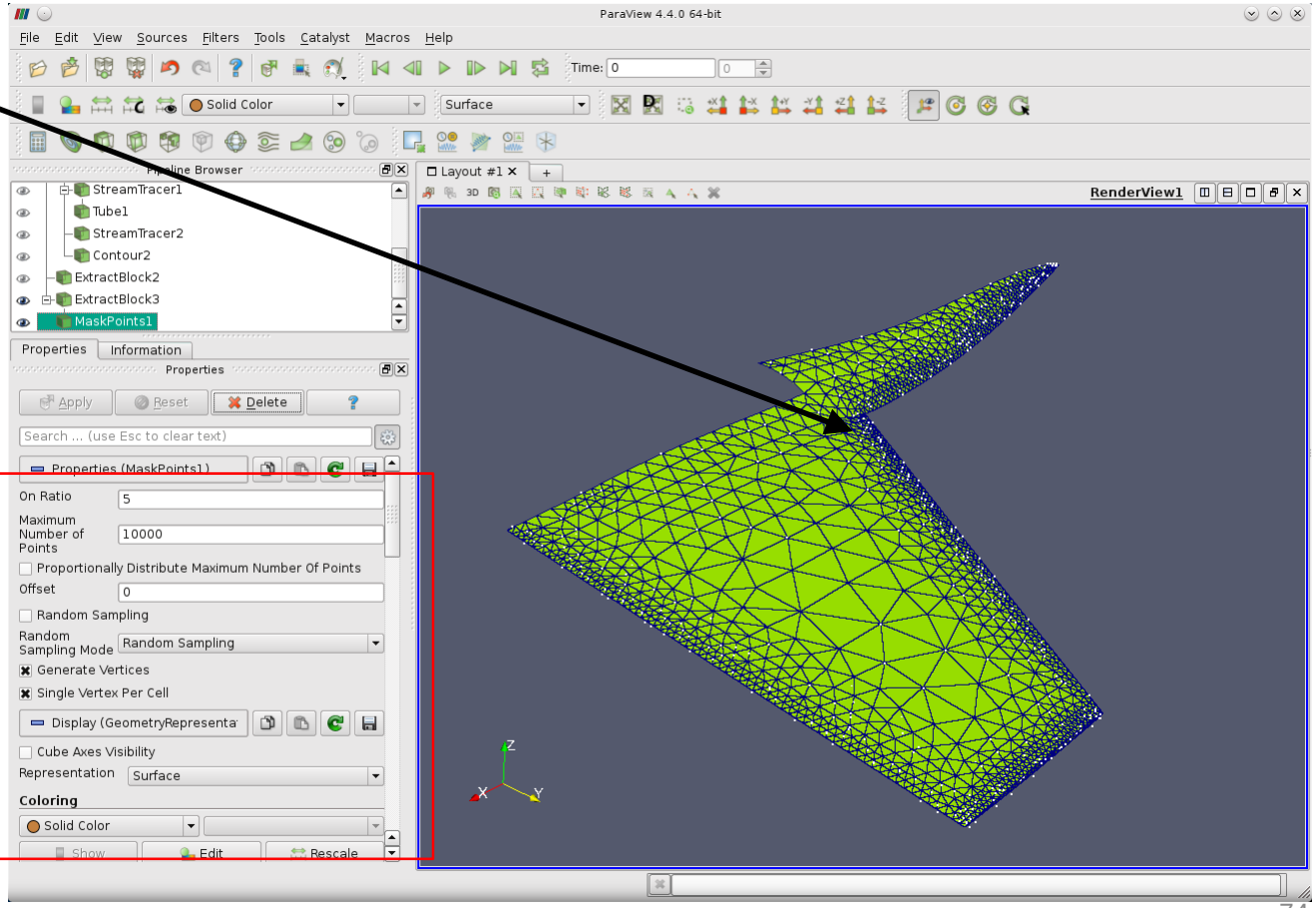
Scientific visualization with paraFoam/ParaView

Releasing streamlines from a surface

- Let's apply a filter to the object `ExtractBlock3` in order to reduce the number of points on the surfaces.
- Select the object `ExtractBlock3`.
- From the menu `Filters → Alphabetical`, select the `Mask Points` filter.

The white dots are the masked points

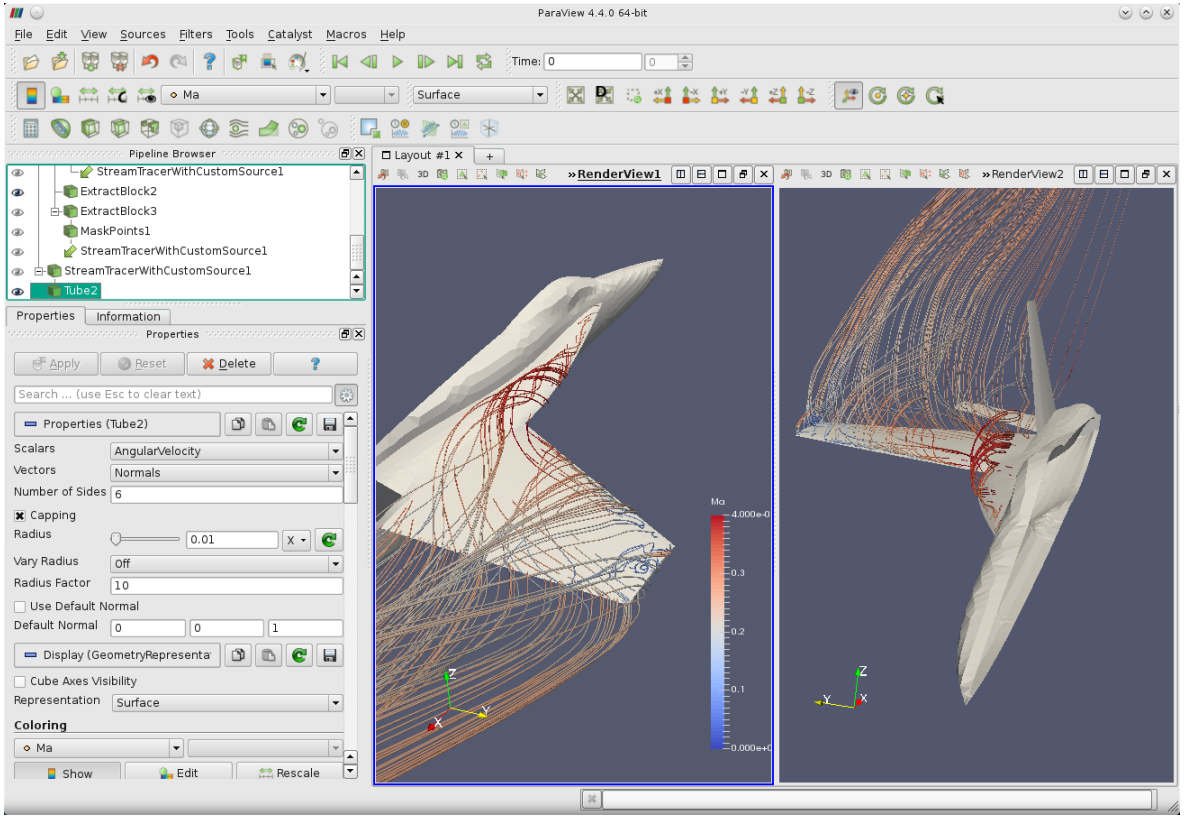
- Masking options.
- Instead of releasing the streamlines from all the surface points, we apply a filter, reducing in this way the number of points.
- Feel free to play with all the options available.



Scientific visualization with paraFoam/ParaView

Releasing streamlines from a surface

- Select the object `MaskPoints1`.
- Select the filter `Stream Tracer With Custom Source`, from the menu `Filters → Alphabetical`.
- As Seed source select the object `MaskPoints1` and as input select the object `ExtractBlock1`.
- Select the object `StreamTracerWithCustomSource1`.
- From the menu `Filters → Alphabetical`, select the `Tube` filter.



Scientific visualization with paraFoam/ParaView

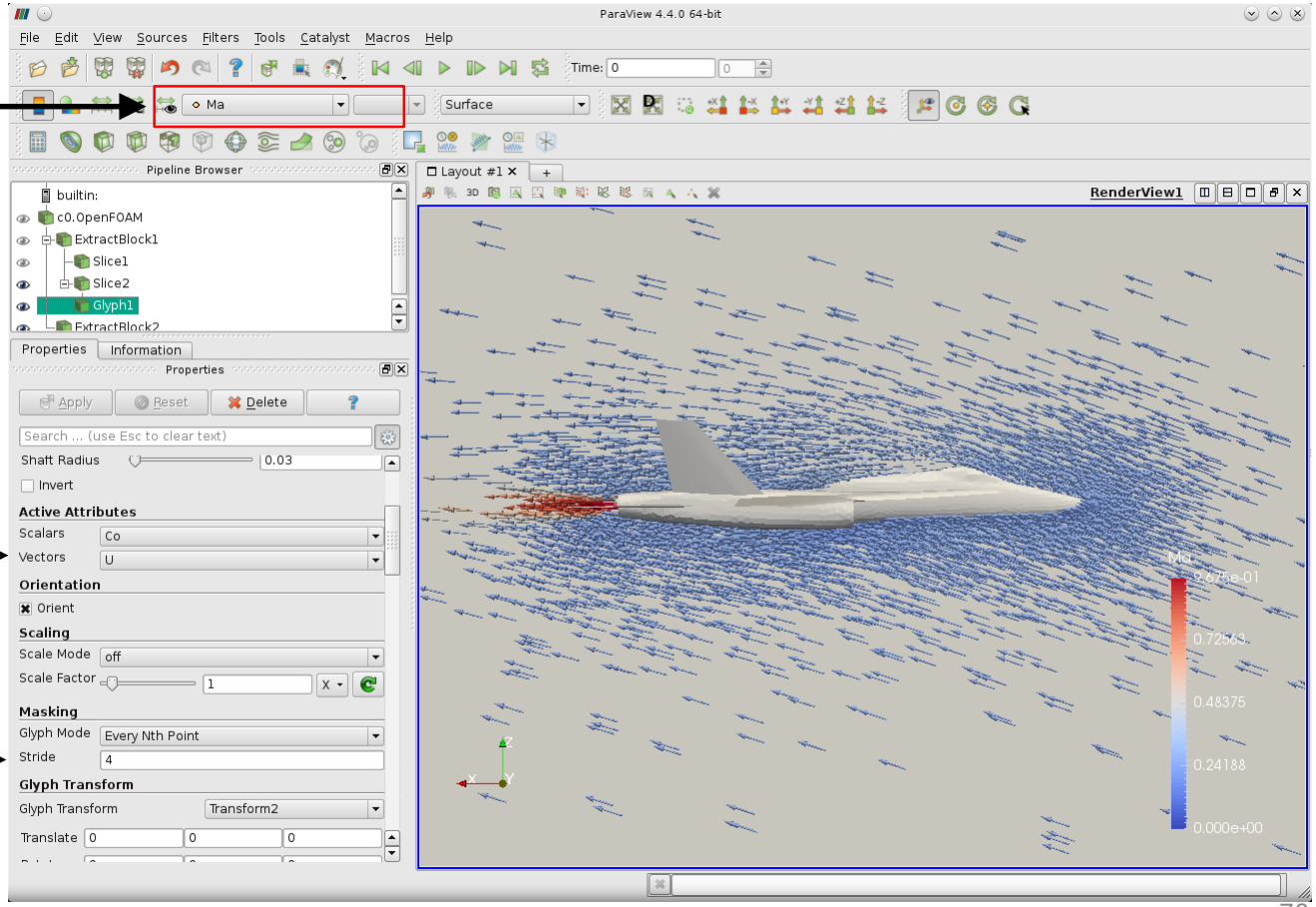
Glyphs (vectors)

- Select the object `ExtractBlock1`, and create two slices. One normal to X and the other normal to Y.
- Select slice normal to Y (in this case the object `Slice2`), and from the menu `Filters → Alphabetical` or the toolbar, select the `Glyph` filter 

Color the glyphs using `Ma`

Select `U` for `Vectors`

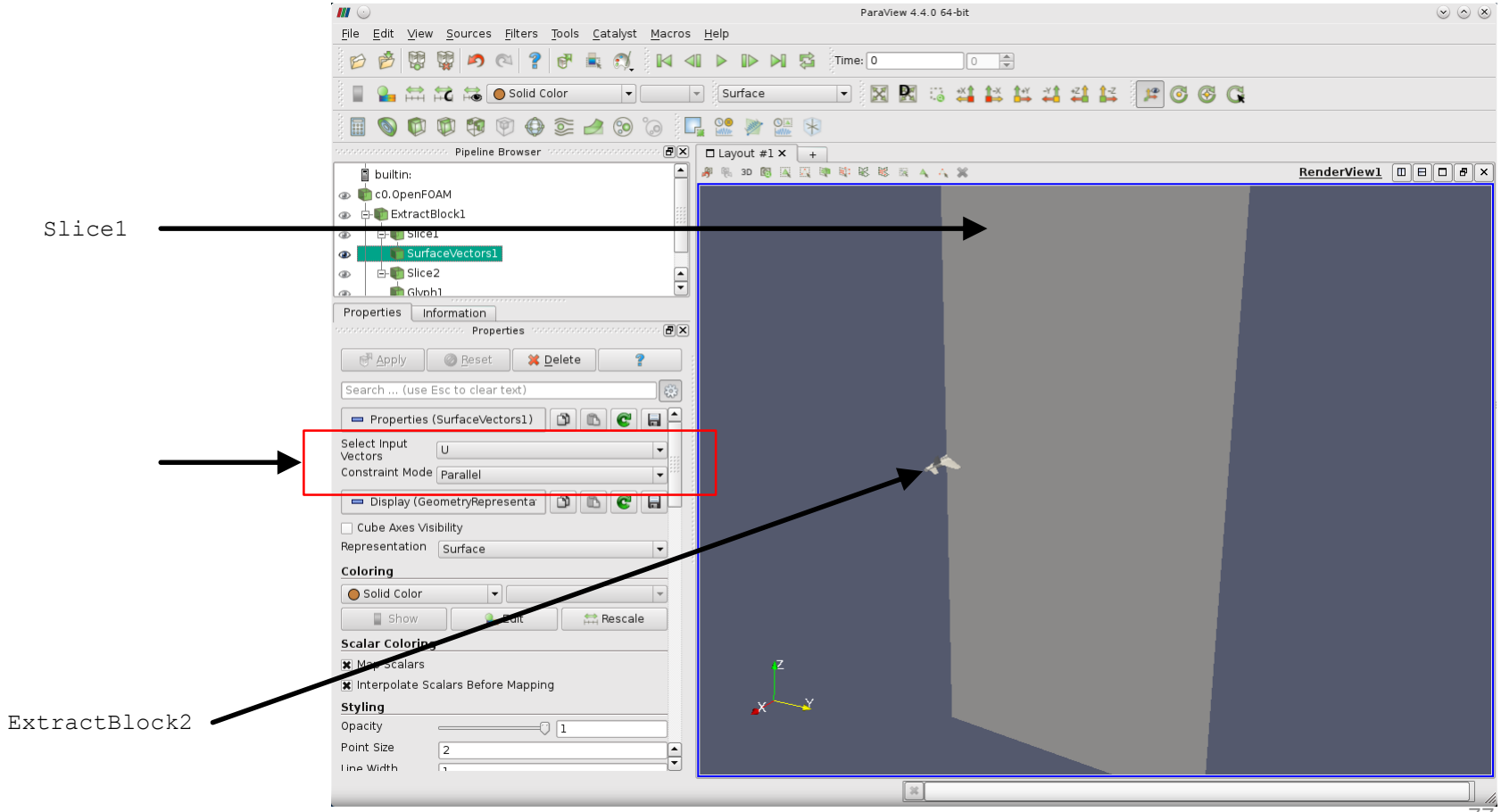
Adjust the glyphs options to obtain a similar result



Scientific visualization with paraFoam/ParaView

Glyphs (vectors)

- Select the slice normal to X (in this case the object `Slice1`) and from the menu `Filters → Alphabetical` apply the filter `Surface Vectors`.
- Select `U` as `input vectors` and `Parallel` as `Constraint Mode`.

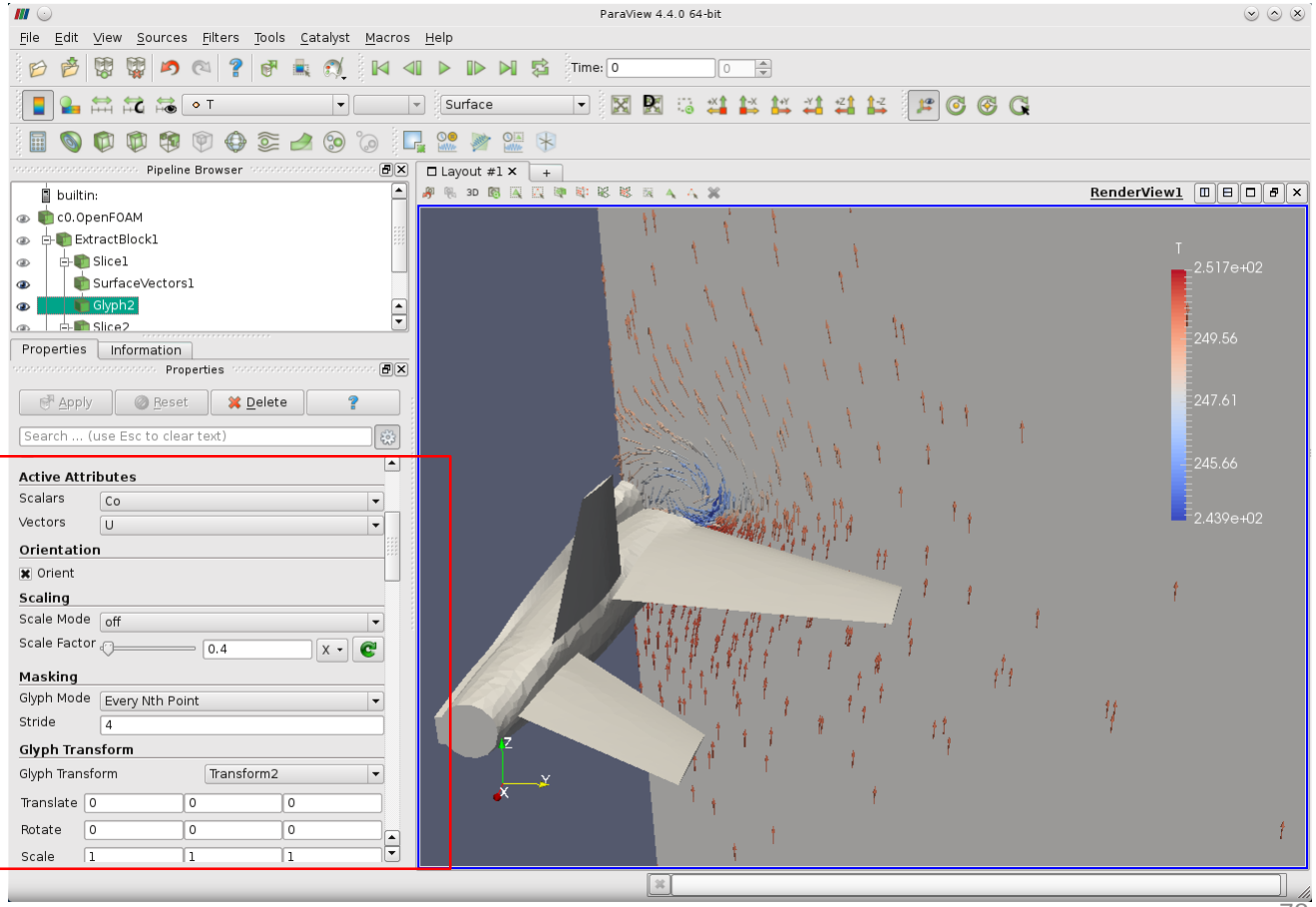


Scientific visualization with paraFoam/ParaView

Glyphs (vectors)

- Select the object `SurfaceVectors1` and from the menu `Filters → Alphabetical` or the toolbar, select the `Glyph` filter 
- Color the glyphs using the variable `T`.

Adjust the glyphs options to obtain a similar result



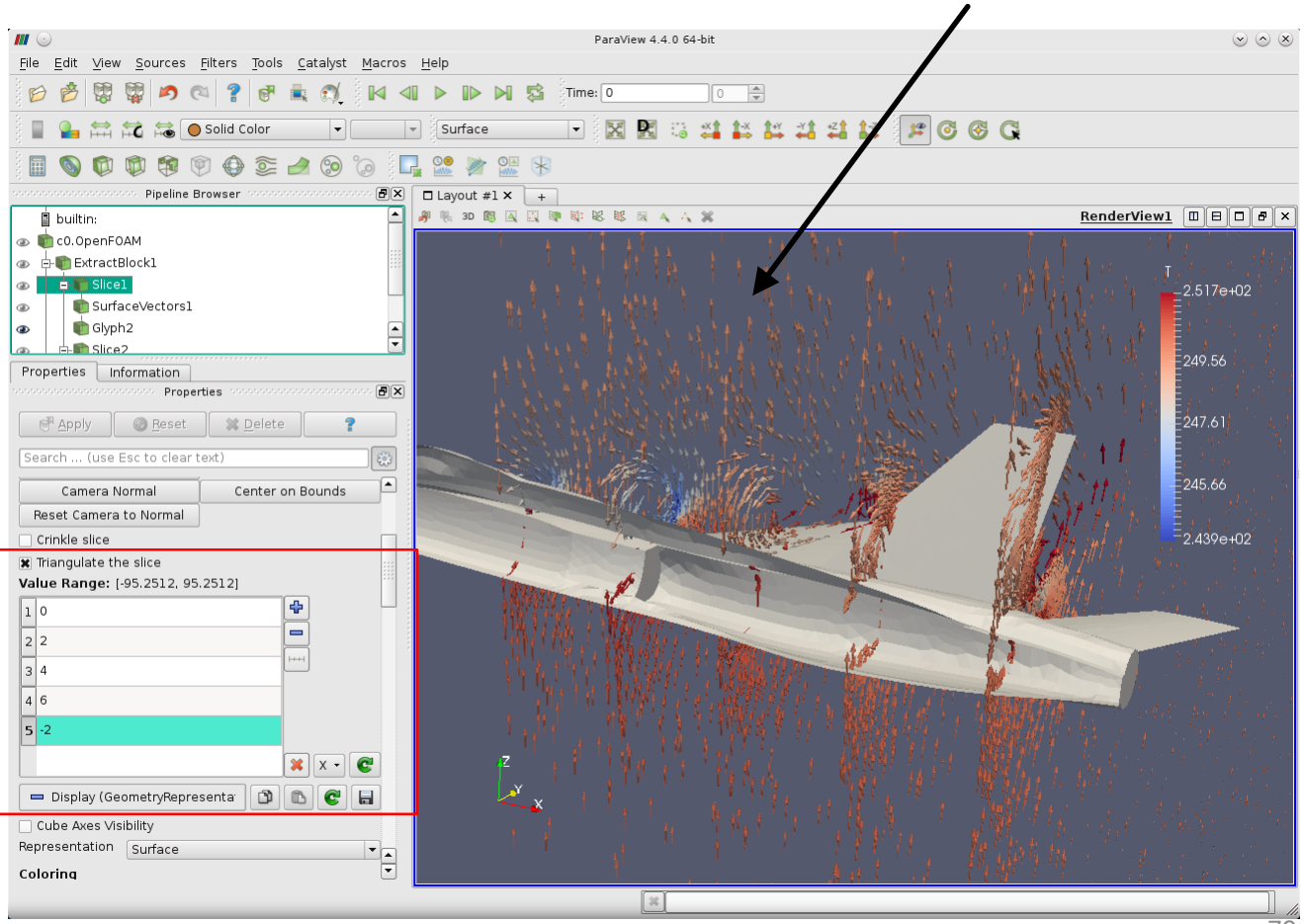
Scientific visualization with paraFoam/ParaView

Glyphs (vectors)

- Select the object `Slice1` and from the options in the `Properties` panel, add more planes.
- The 3D view is updated automatically and all the new planes will inherit the properties.

`Glyph2` inherit all the properties of `Slice1` and `SurfaceVectors1`

Add more planes



Scientific visualization with paraFoam/ParaView

Gradient of Unstructured Dataset

- Let's use the filter `Gradient of Unstructured Dataset` to compute the gradient of a field, and the Q-criterion.
- Select the object `ExtractBlock1` and from the menu `Filters → Alphabetical` select the `Gradient of Unstructured Dataset` filter.

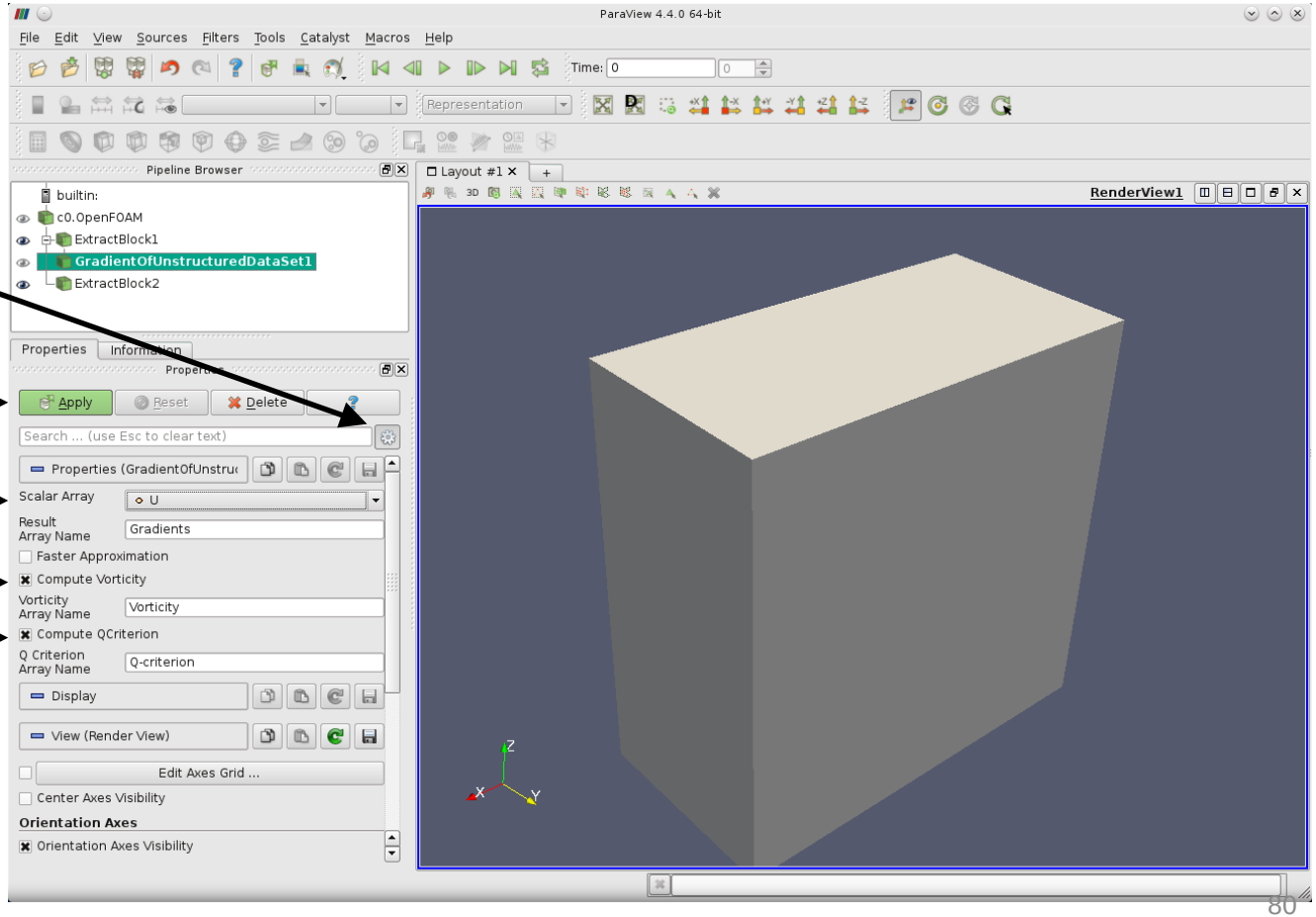
To compute the `Qcriterion` you need to enable the advanced properties

Remember to click the Apply button

Select the input variable, `U` in this case

Check Compute `Vorticity`

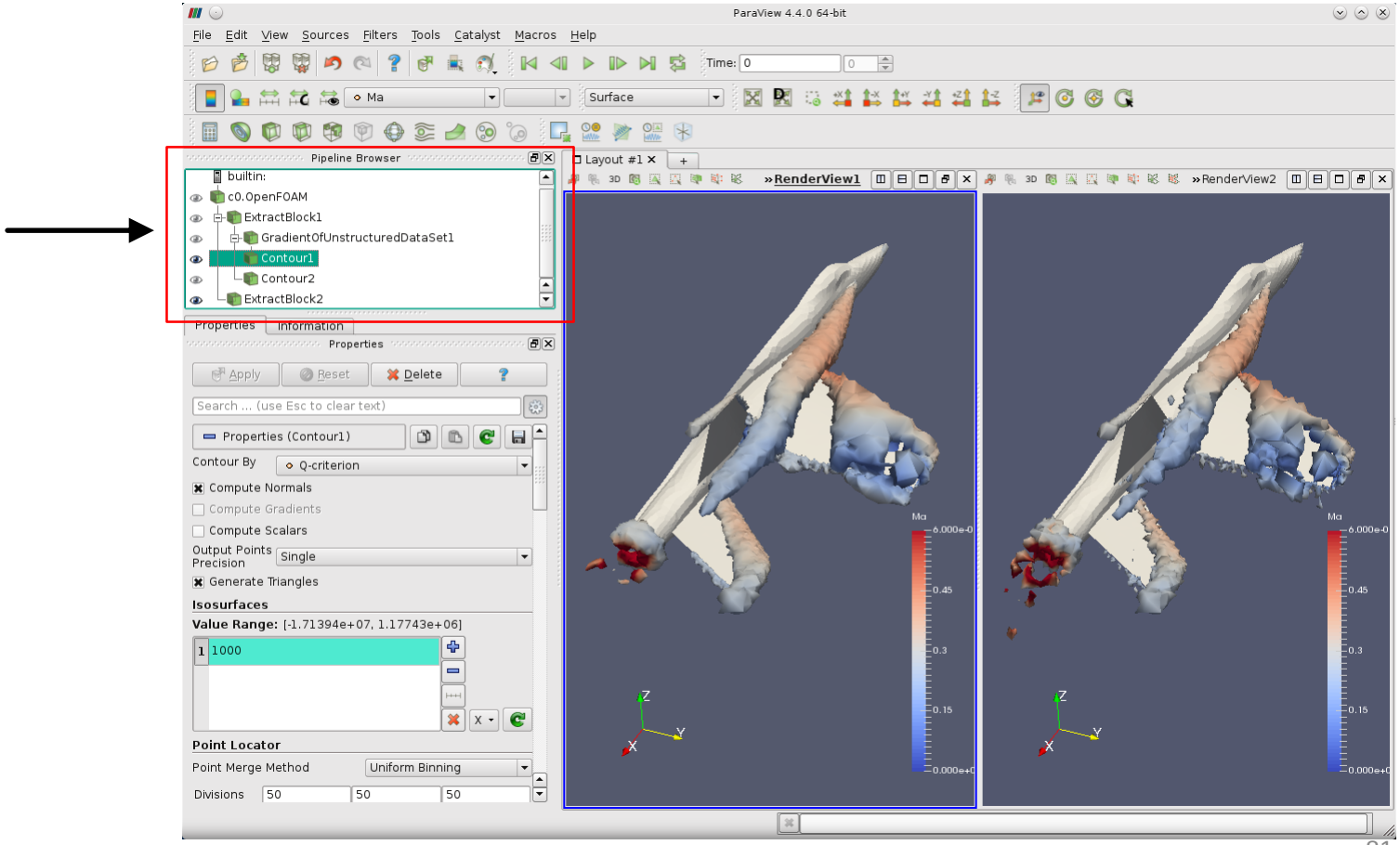
Check Compute `QCriterion`



Scientific visualization with paraFoam/ParaView

Gradient of Unstructured Dataset

- Create a multiview and compare the Q-criterion computed in ParaView with the Q-criterion computed using OpenFOAM®.
- Try to reproduce the same pipeline.

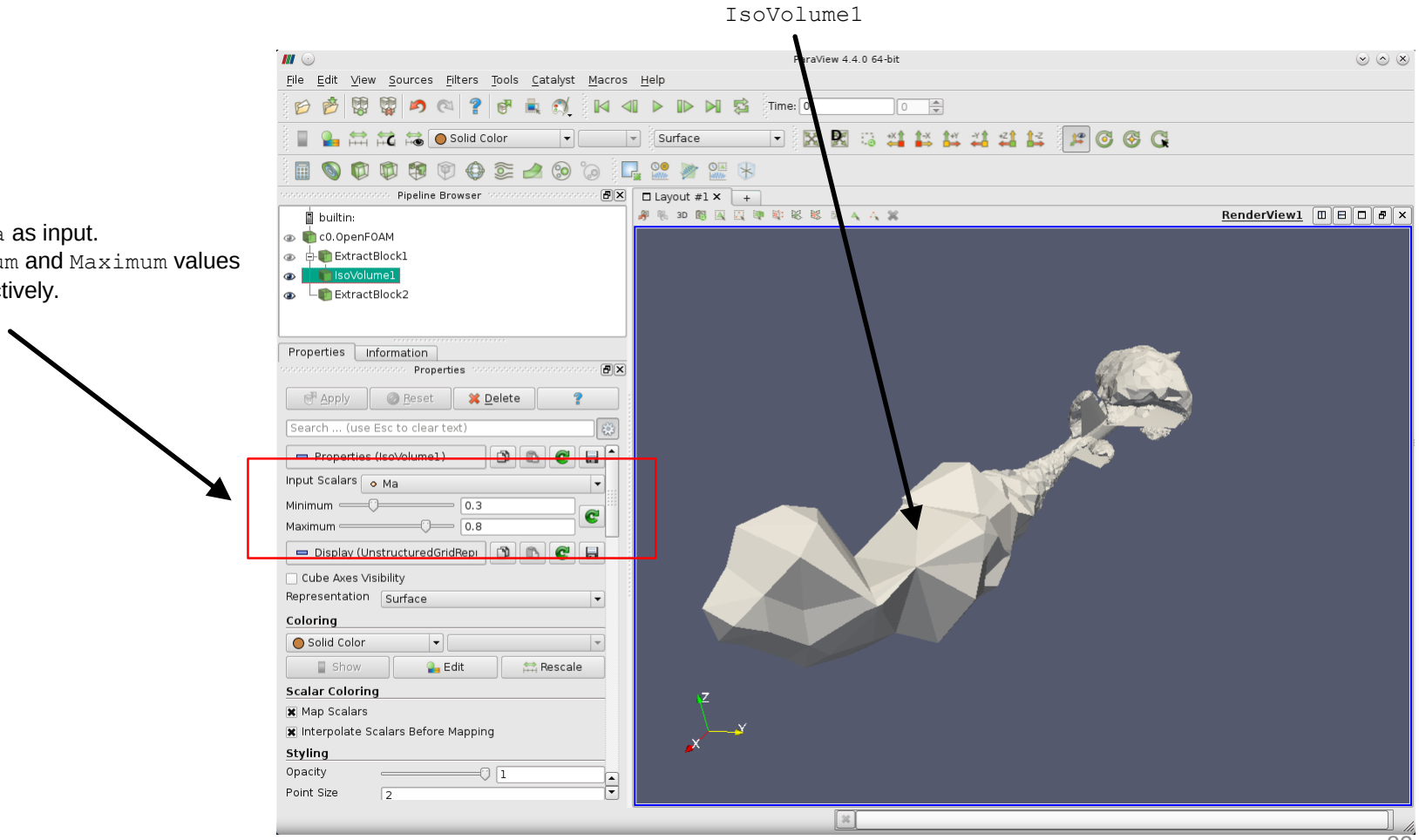


Scientific visualization with paraFoam/ParaView

Iso-Volume

- Let's use the filter `Iso Volume` to do volume rendering.
- Select the object `ExtractBlock1` and from the menu `Filters → Alphabetical` select the `Iso Volume` filter.

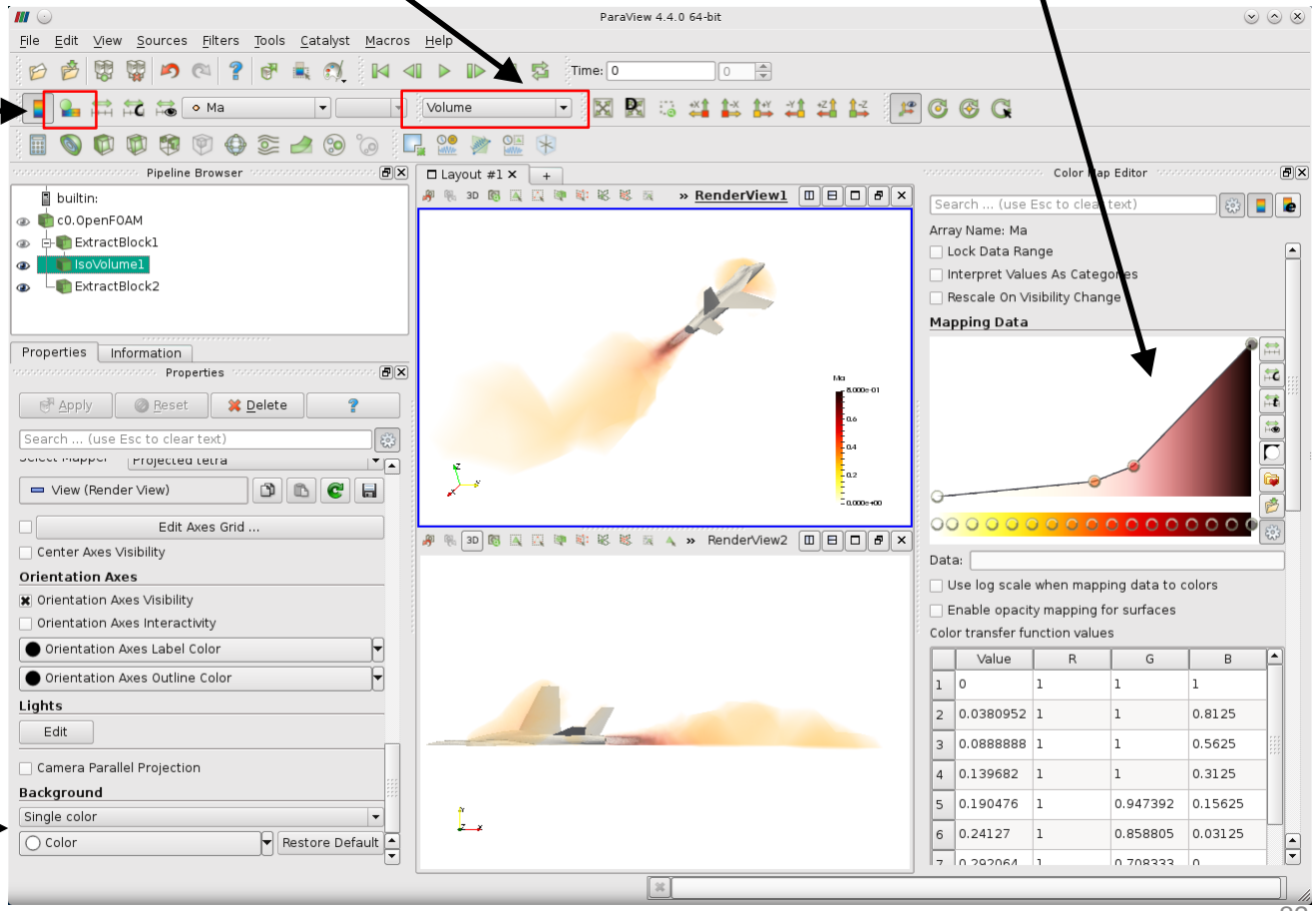
- Use the variable `Ma` as input.
- Choose as `Minimum` and `Maximum` values 0.3 and 0.8, respectively.



Scientific visualization with paraFoam/ParaView

Iso-Volume

- Create a multiview and plot the extracted iso-volume using `Volume` rendering.
- Try to reproduce the same pipeline.



- Depending on the mesh dimensions, this operation can be time consuming.
- You will need a good video card.

Scientific visualization with paraFoam/ParaView

The calculator

- Let's start from a clean pipeline.
- Select the object `c0.OpenFOAM`, and from the menu `Filters → Alphabetical` or the toolbar, select the `Calculator` filter 

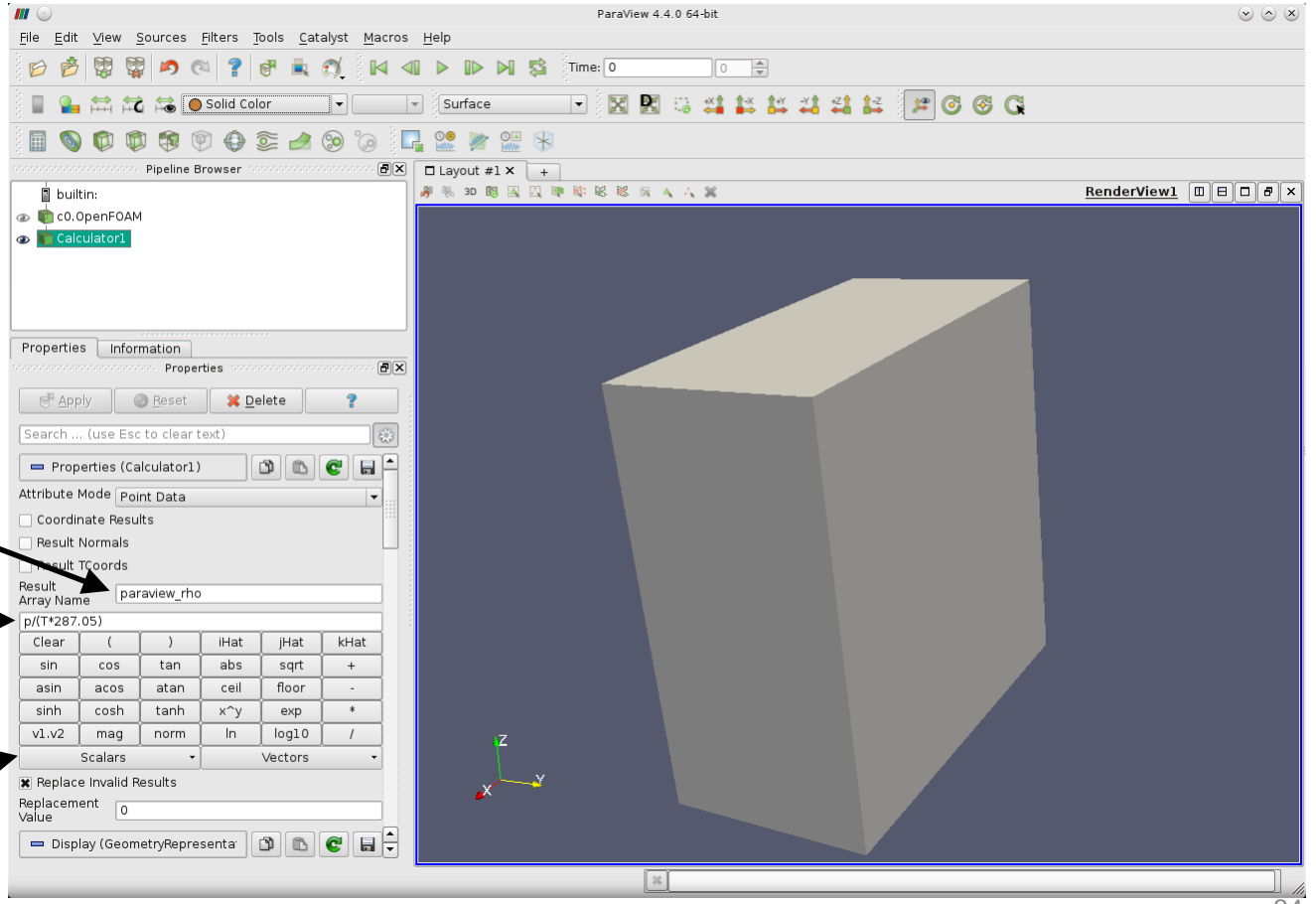
- Let's use the calculator filter to compute the density inside ParaView.
- To compute the density we can use the equation of state for an ideal gas,

$$\rho = \frac{p}{T \times R_{air}}$$

Give a name to the new variable

Enter the equation

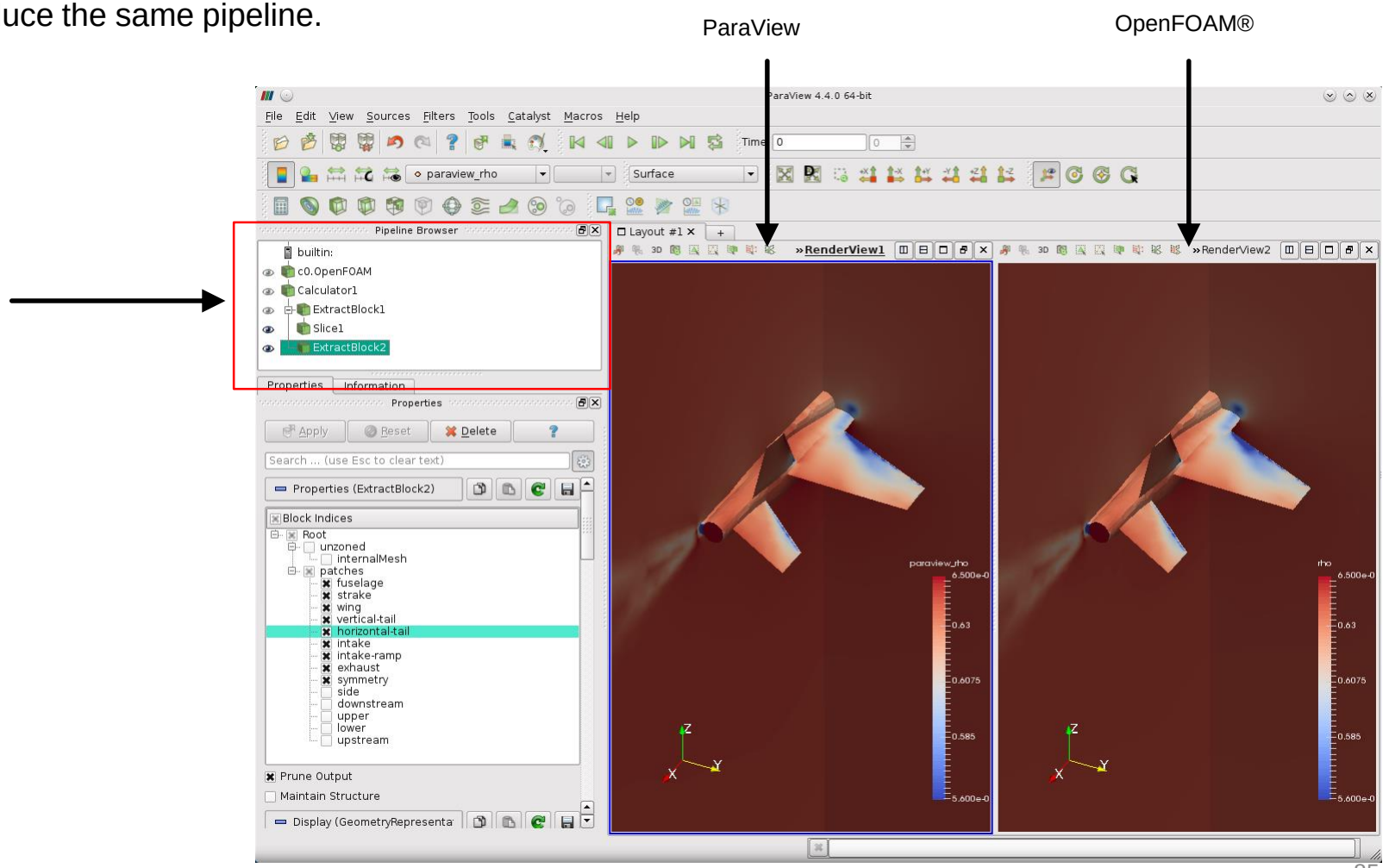
Select the valid fields from the `Scalars` and `Vectors` drop down menu.



Scientific visualization with paraFoam/ParaView

The calculator

- Create a multiview and compare the density value computed in ParaView with the density value computed in OpenFOAM®.
- Try to reproduce the same pipeline.



Scientific visualization with paraFoam/ParaView

The calculator

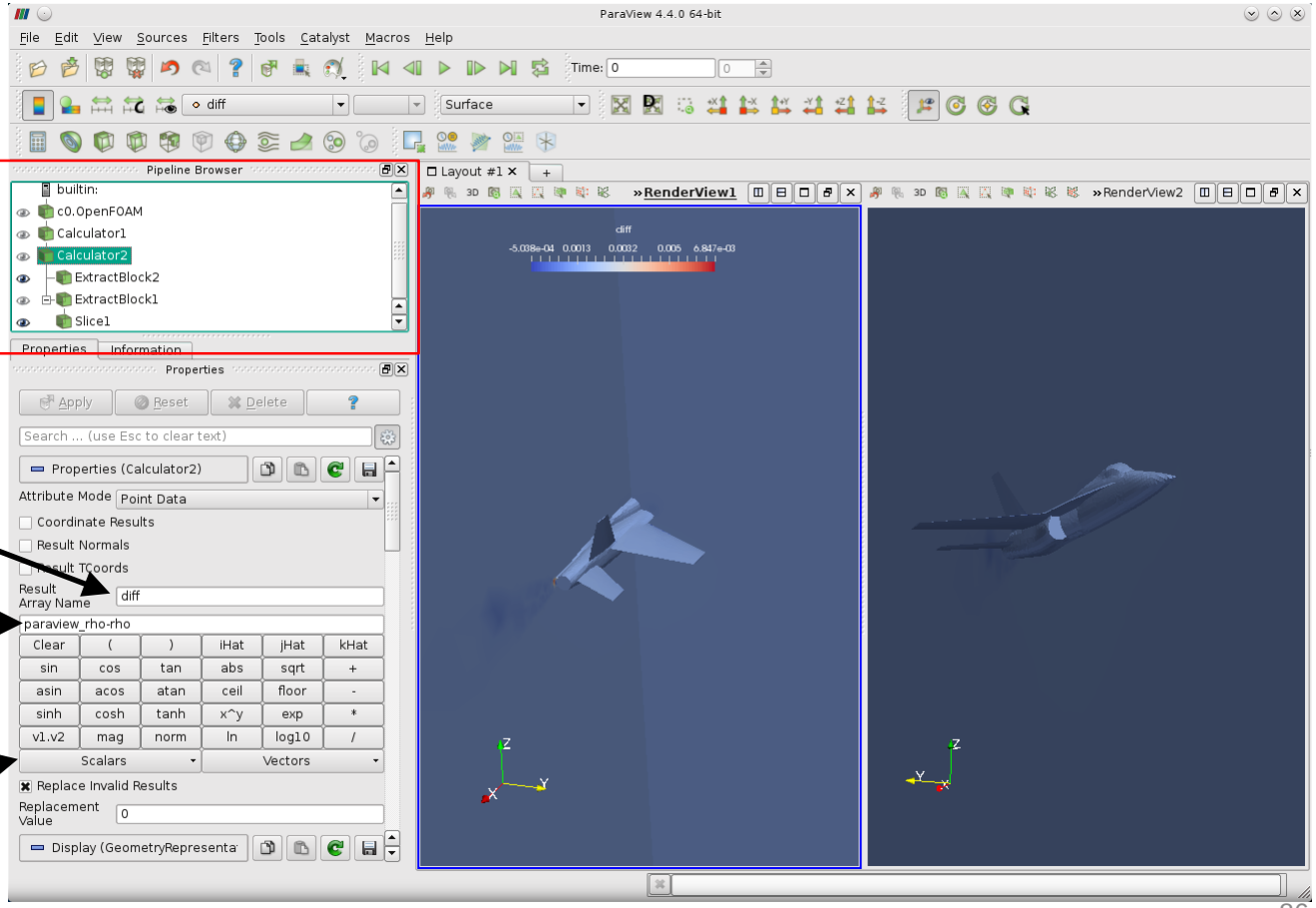
- Let's apply the calculator filter to the object `Calculator2`.
- Compute the difference between the density computed with ParaView and the density computed in OpenFOAM® and plot the results.

Try to reproduce the same pipeline.

Give a name to the new variable

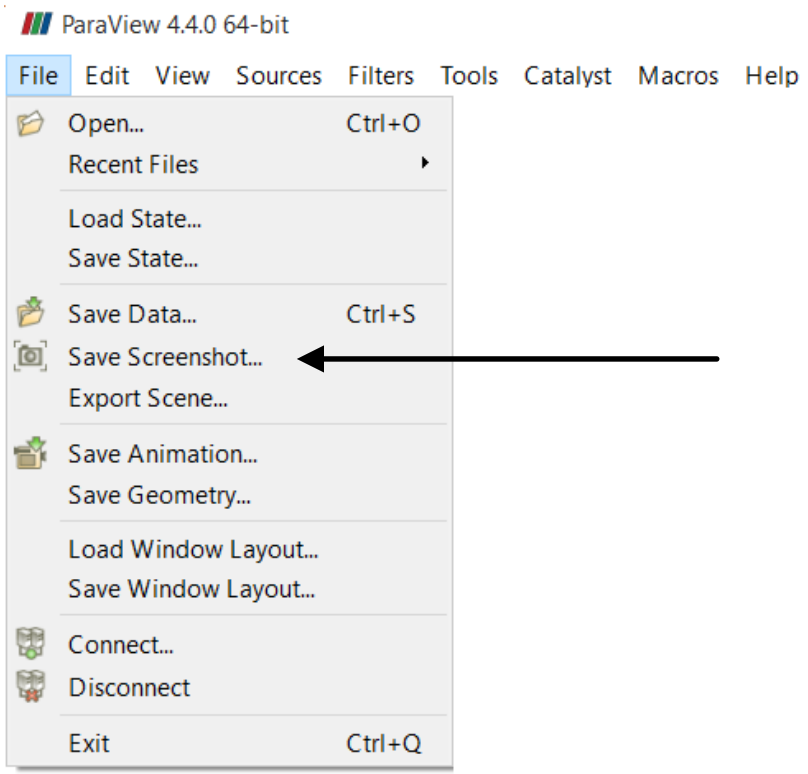
Input the following equation
diff = ParaView_rho - rho

Select the valid fields from the `Scalars` and `Vectors` drop down menu.



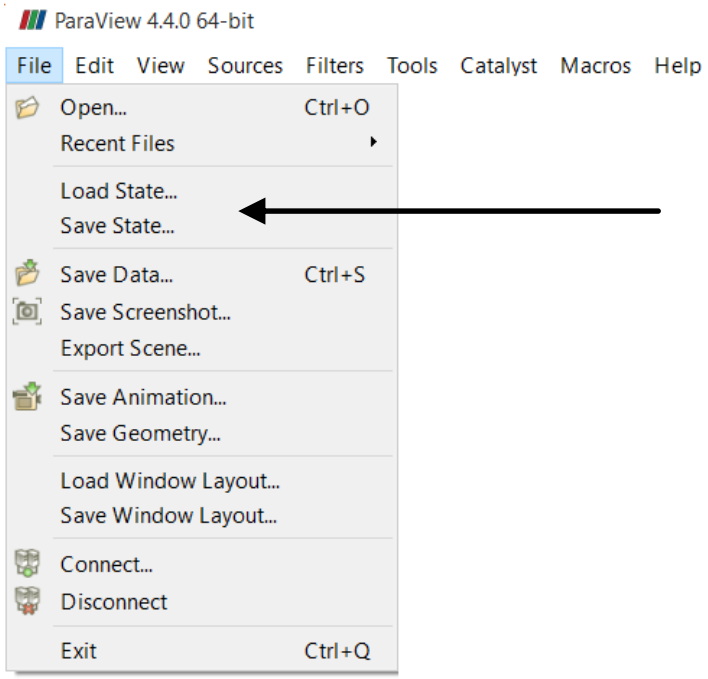
Saving screenshots

- Saving screenshots is straightforward.
- Select the menu `File → Save Screenshot`, set the save screenshots options, and save the file.
- `Export Scene` is equivalent to `Save Screenshot`, but it saves the screenshot in vector format.



Saving states

- States are session configurations used to restart the pipeline and GUI at any time.
- When you save a state you are saving the pipeline and the GUI configuration. Saving states is straightforward.
- Select the menu `File → Save State`, and save the state file.
- To load a state, select the menu `File → Load State`, and load the state file.
- When loading a state, is highly advisable to do it from a clean pipeline.



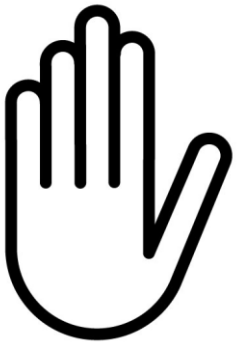
Saving states

- In the directory **`states`**, you will find the following states files:
  - *`view1.pvsm`*
  - *`view2.pvsm`*
  - *`view3.pvsm`*
  - *`view4.pvsm`*
  - *`view5.pvsm`*
  - *`view6.pvsm`*
  - *`view7.pvsm`*
- Try to reload each state and play around with the pipeline.

Scientific visualization with paraFoam/ParaView

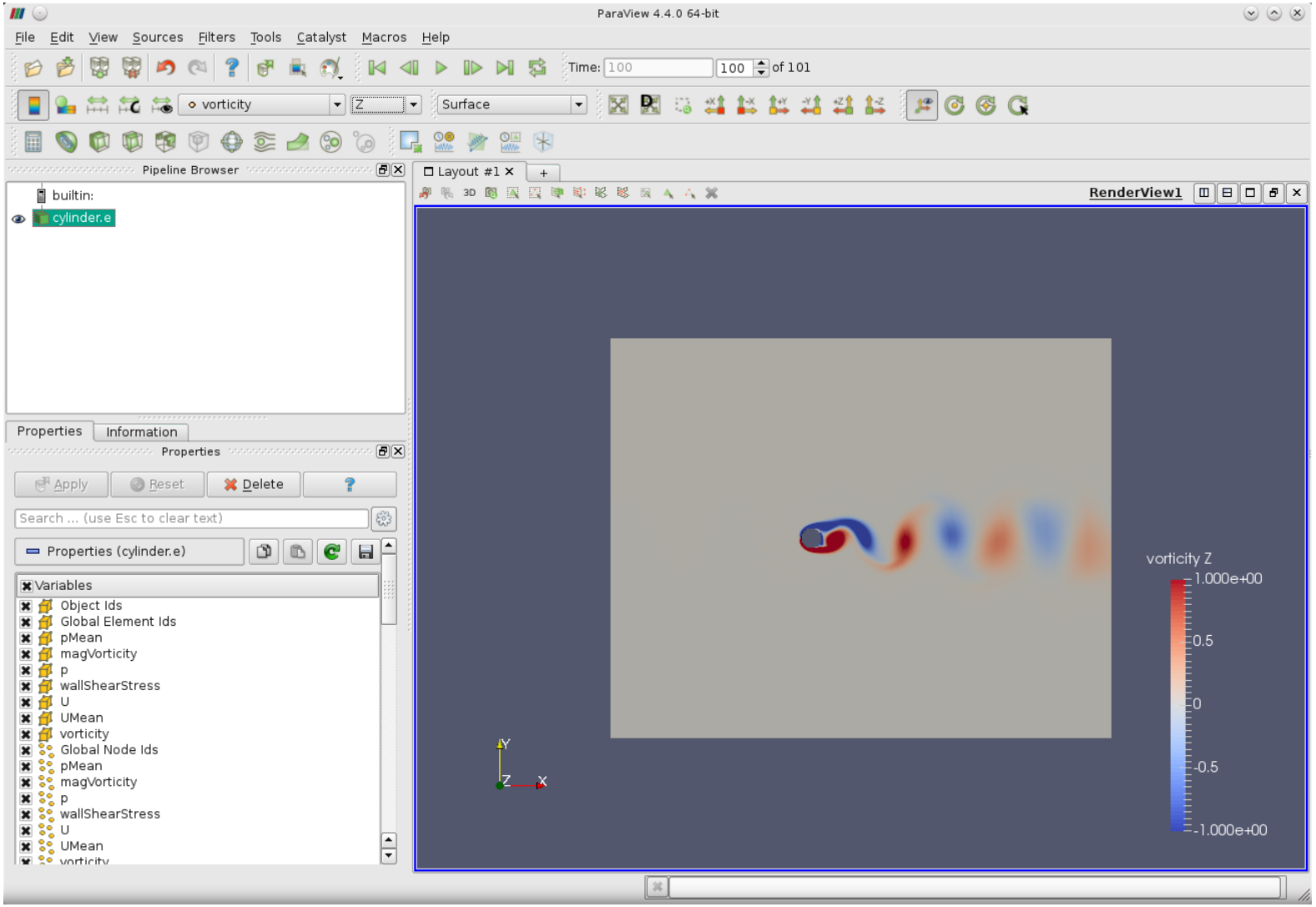
- Let’s do post processing using paraFoam. Go to the directory:

```
vortex_shedding
```



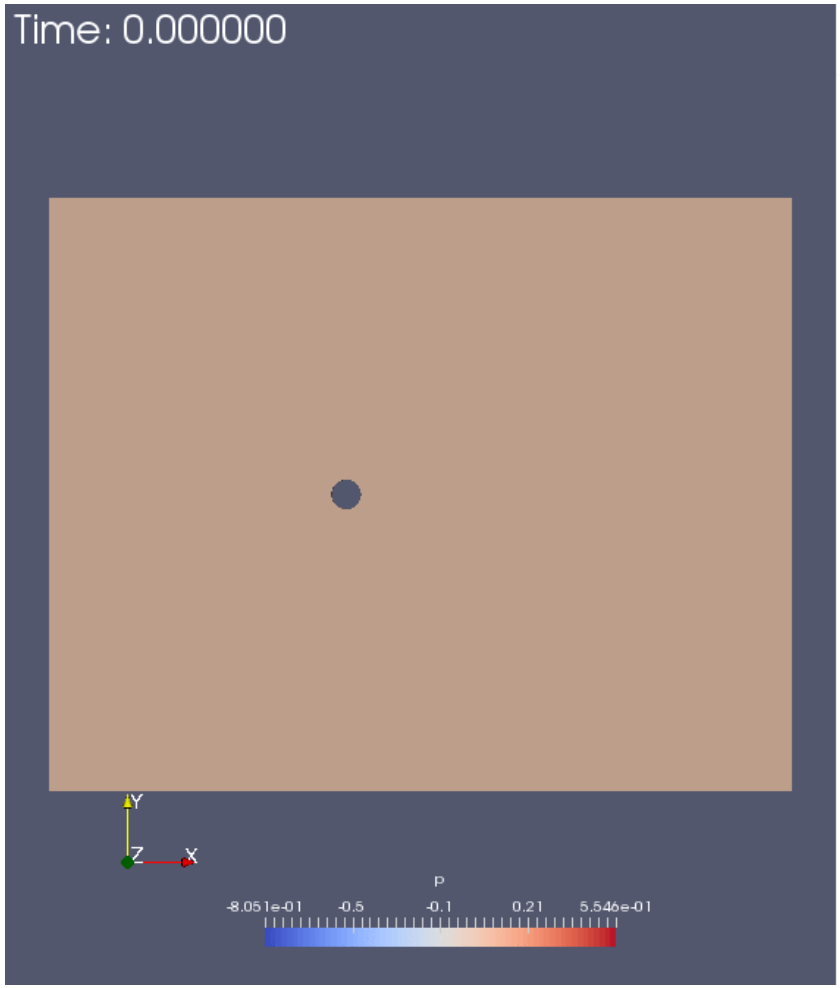
- From this point on, please follow me.
- We are all going to work at the same pace.
- Remember, `$PTOFC` is pointing to the path where you unpacked the tutorials.

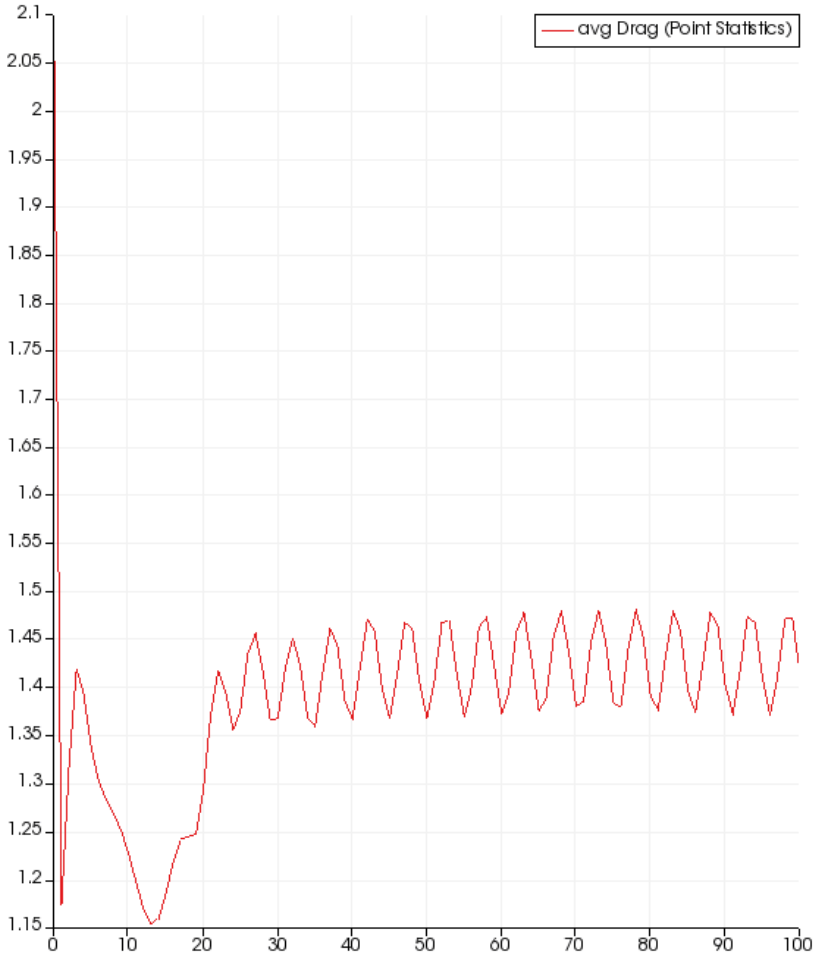
Scientific visualization with paraFoam/ParaView



Some pretty pictures

Scientific visualization with paraFoam/ParaView





Some pretty pictures

Scientific visualization with paraFoam/ParaView

What are we going to do?

- We will use this case to introduce paraFoam/ParaView.
- We will do some sampling directly in paraFoam/ParaView.
- We will work with transient data.
- We do not need to run the simulation, the solution is saved in exodus2 format.
- To find the numerical solution we used the solver `icoFoam`.
- This is the 2D cylinder case at Re = 200.

Visualizing the case

- Go to the directory **`$PTOFC/101PARAVIEW/vortex_shedding`** and type in the terminal:

```
1. $> tar –xzvf cylinder.tar.gz
```

- We will do the post processing using paraFoam, in the terminal type

```
1. $> paraFoam
```

Ignore the warning and use the default option. Then open the file *`cylinder.e`* in paraFoam.

- If you want to use ParaView instead, in terminal type

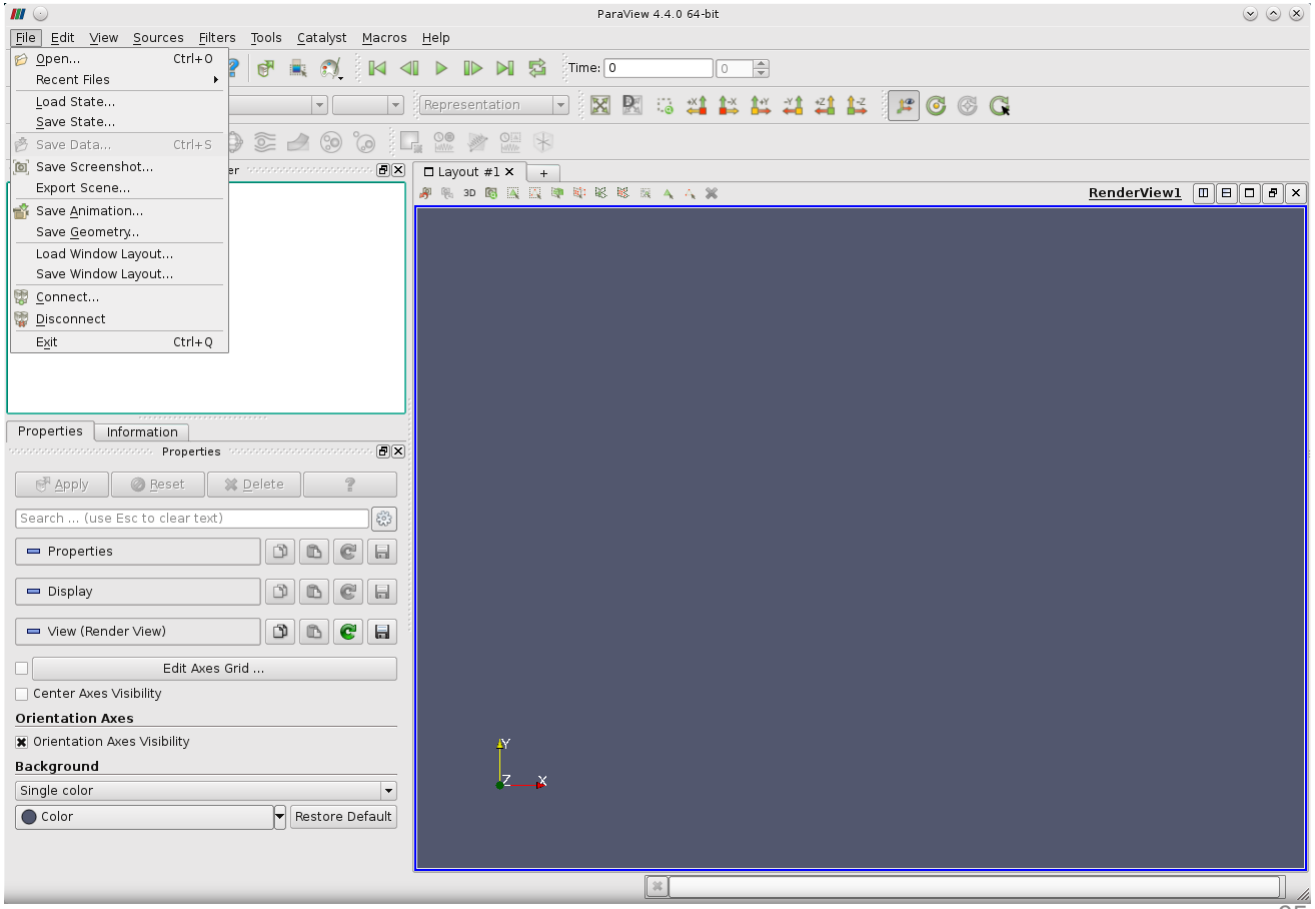
```
1. $> paraview5
```
Paraview5 is an alias to call paraview version 5.0.1

Then open the file *`cylinder.e`* in ParaView.

Scientific visualization with paraFoam/ParaView

Reading in data

- Select the menu `File → Open`.
- Select the file *`cylinder.e`*, and click Apply 

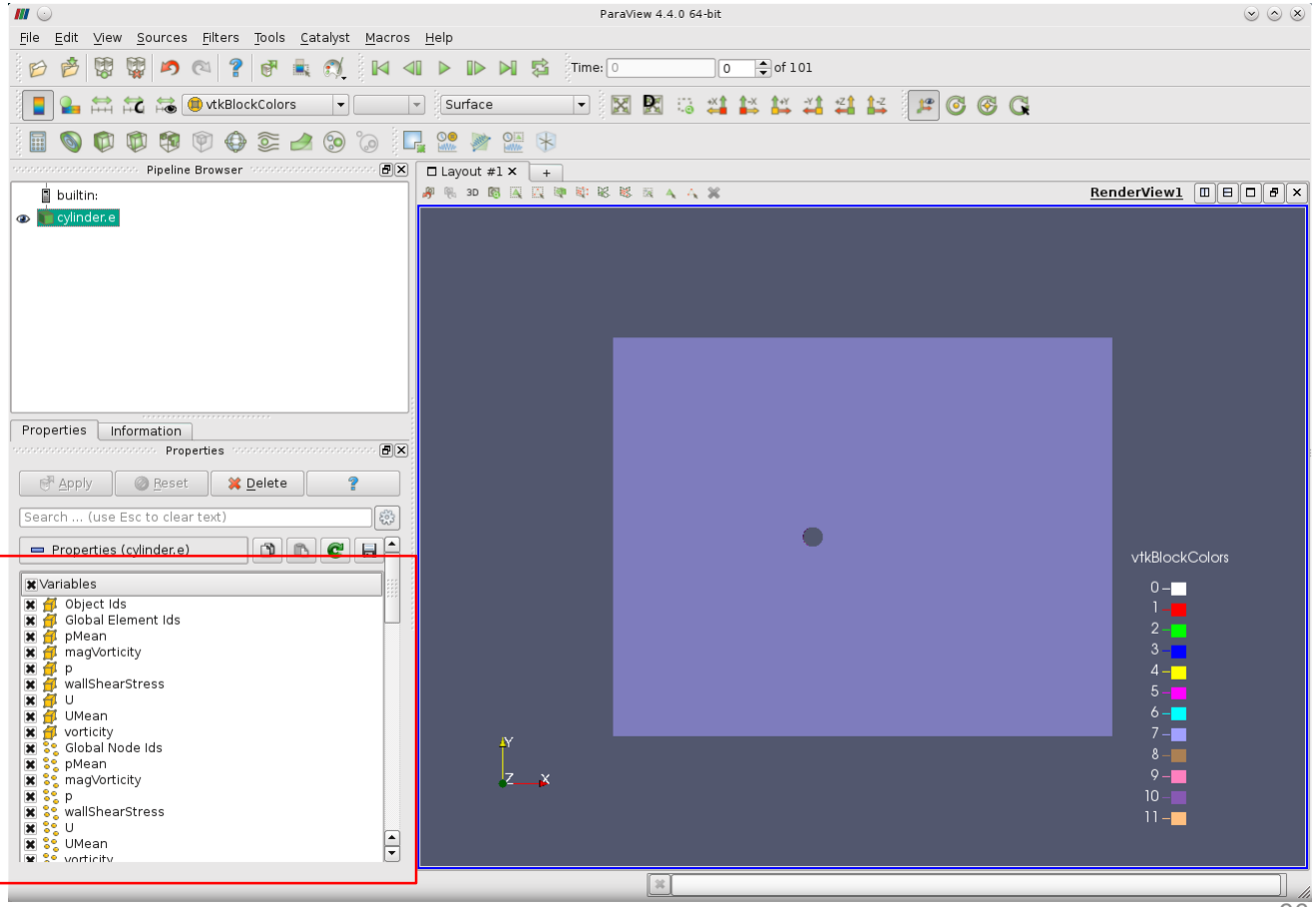


Scientific visualization with paraFoam/ParaView

Data representation and field coloring

- In the properties panel, select all the variables.

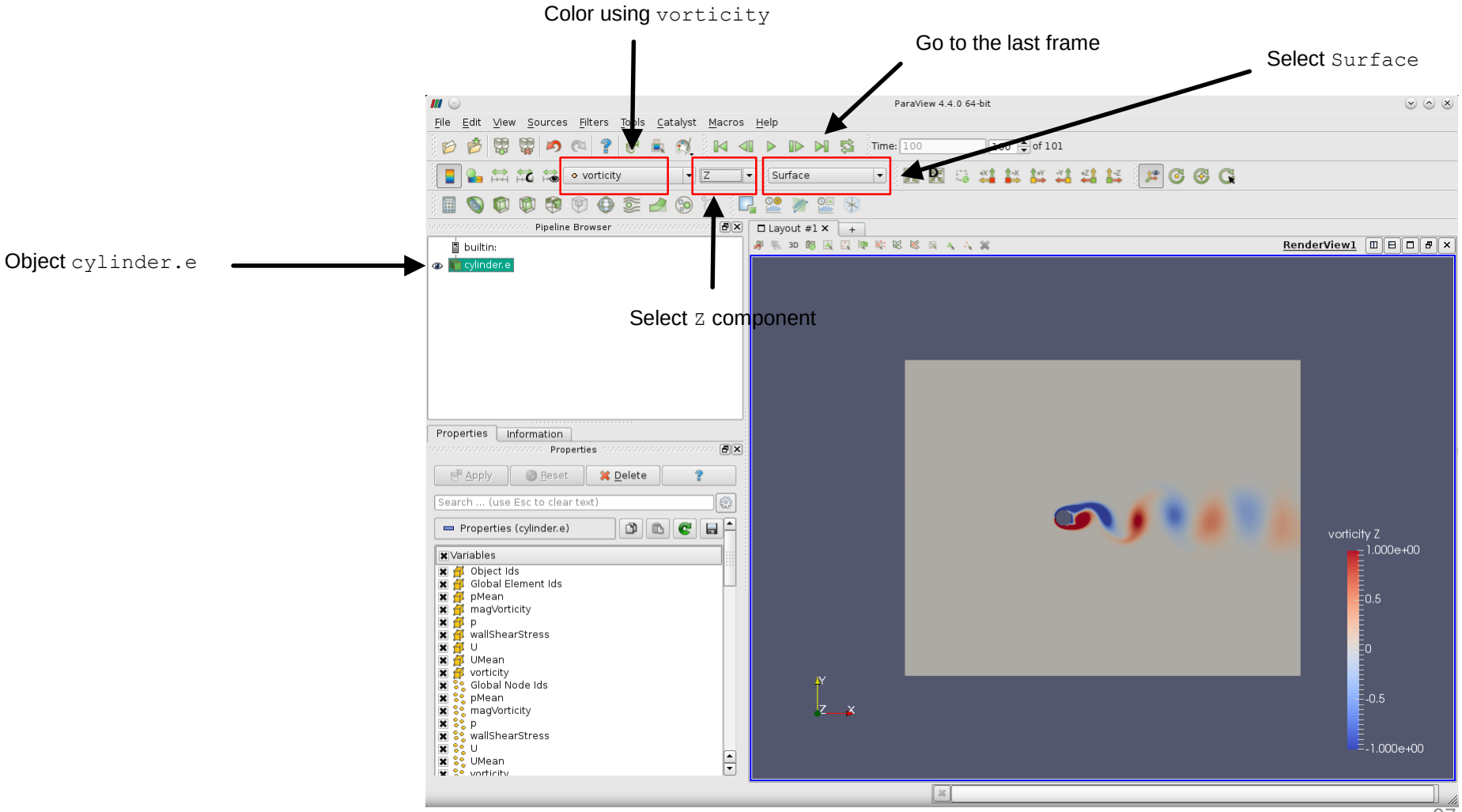
Select all variables



Scientific visualization with paraFoam/ParaView

Data representation and field coloring

- Plot Z-vorticity, adjust the range, and go to the last frame.



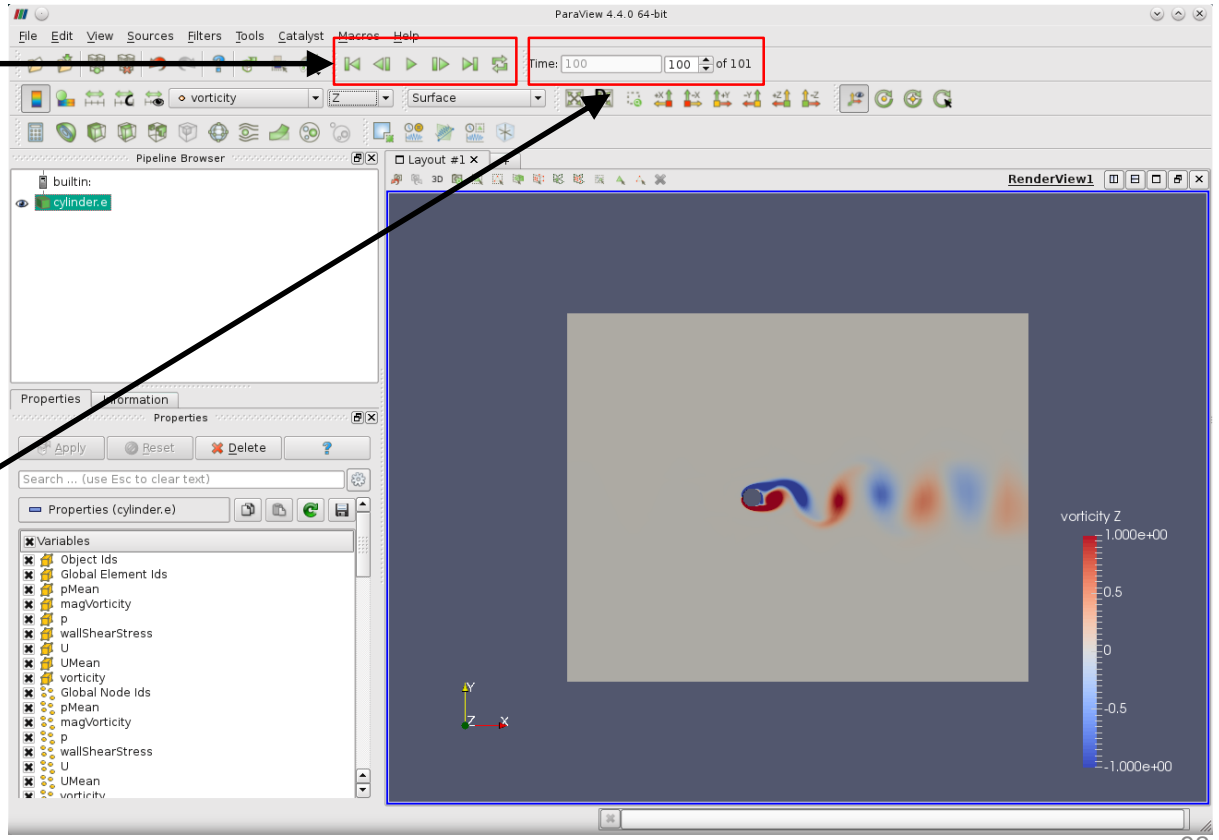
Scientific visualization with paraFoam/ParaView

Animating the data

- Press the play  button in the `VCR Controls` toolbar 
- Enjoy the animation.
- Nothing change when we work with transient data, everything we have done so far remains the same.
- The only new feature is that we have control over the time.

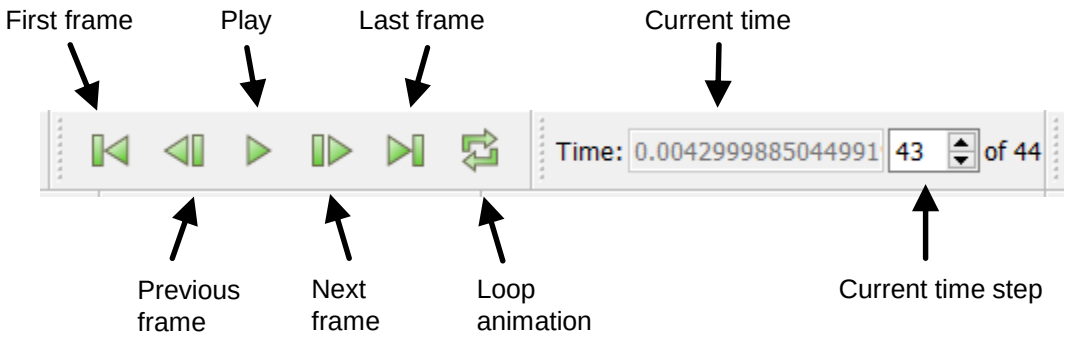
`VCR Controls`

`Current Time Controls`



Animating the data

- At this point let's review the `VCR Controls` and `Current Time Controls` toolbars.

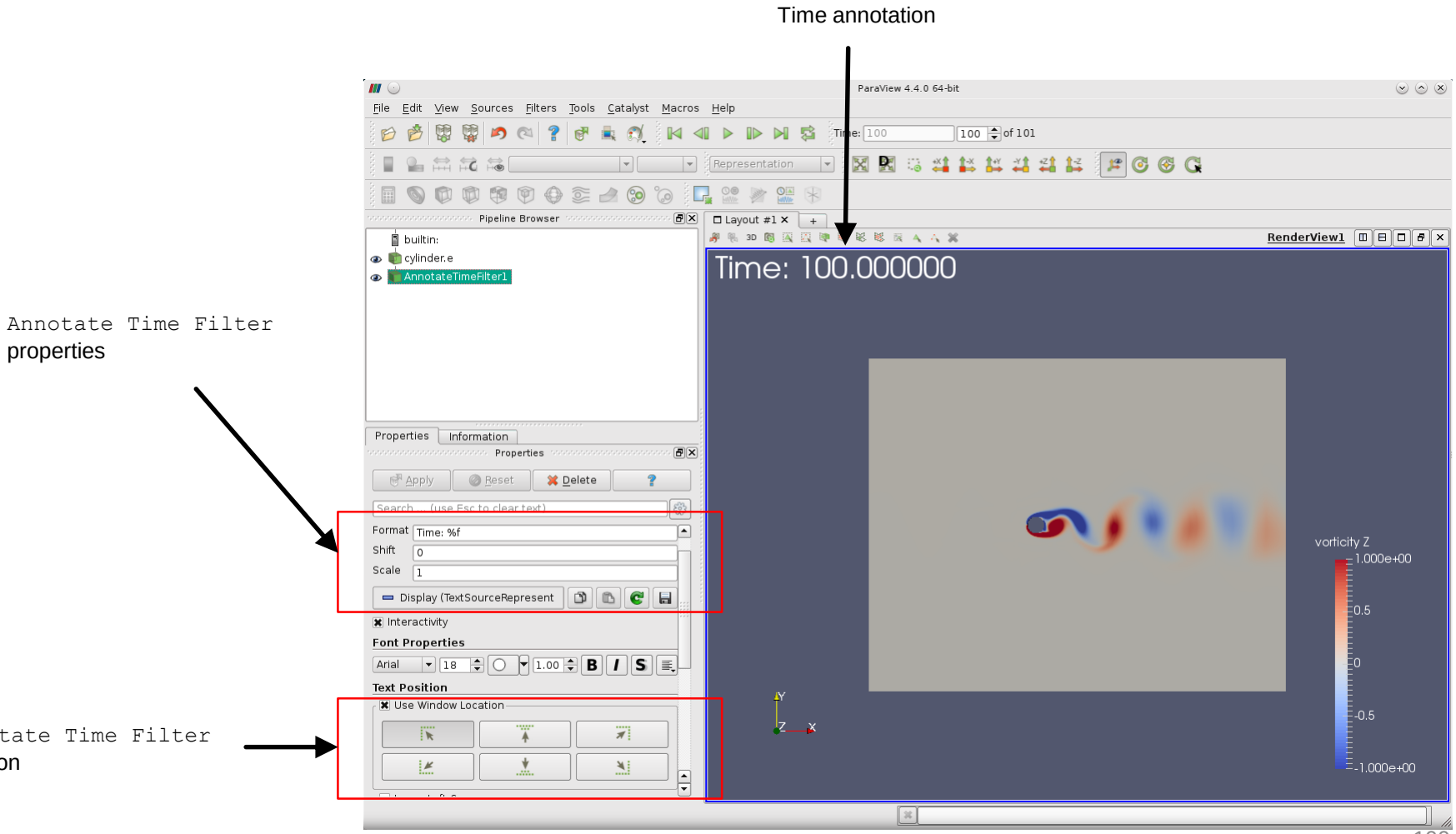


- And this is all we need to know to deal with transient data.

Scientific visualization with paraFoam/ParaView

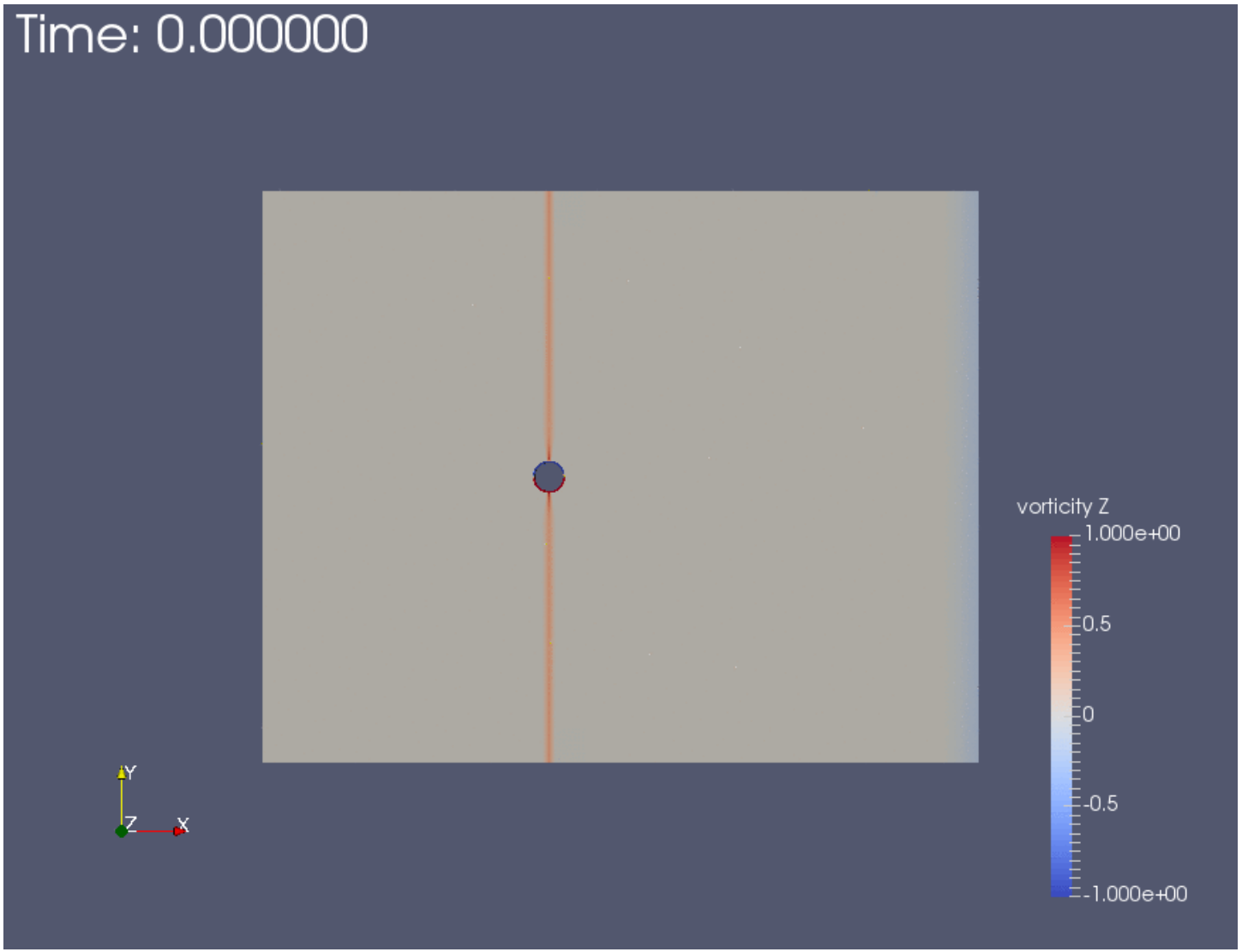
Annotate Time Filter

- Select the object `cylinder.e` and from the menu `Filters → Alphabetical` select `Annotate Time Filter`.



Scientific visualization with paraFoam/ParaView

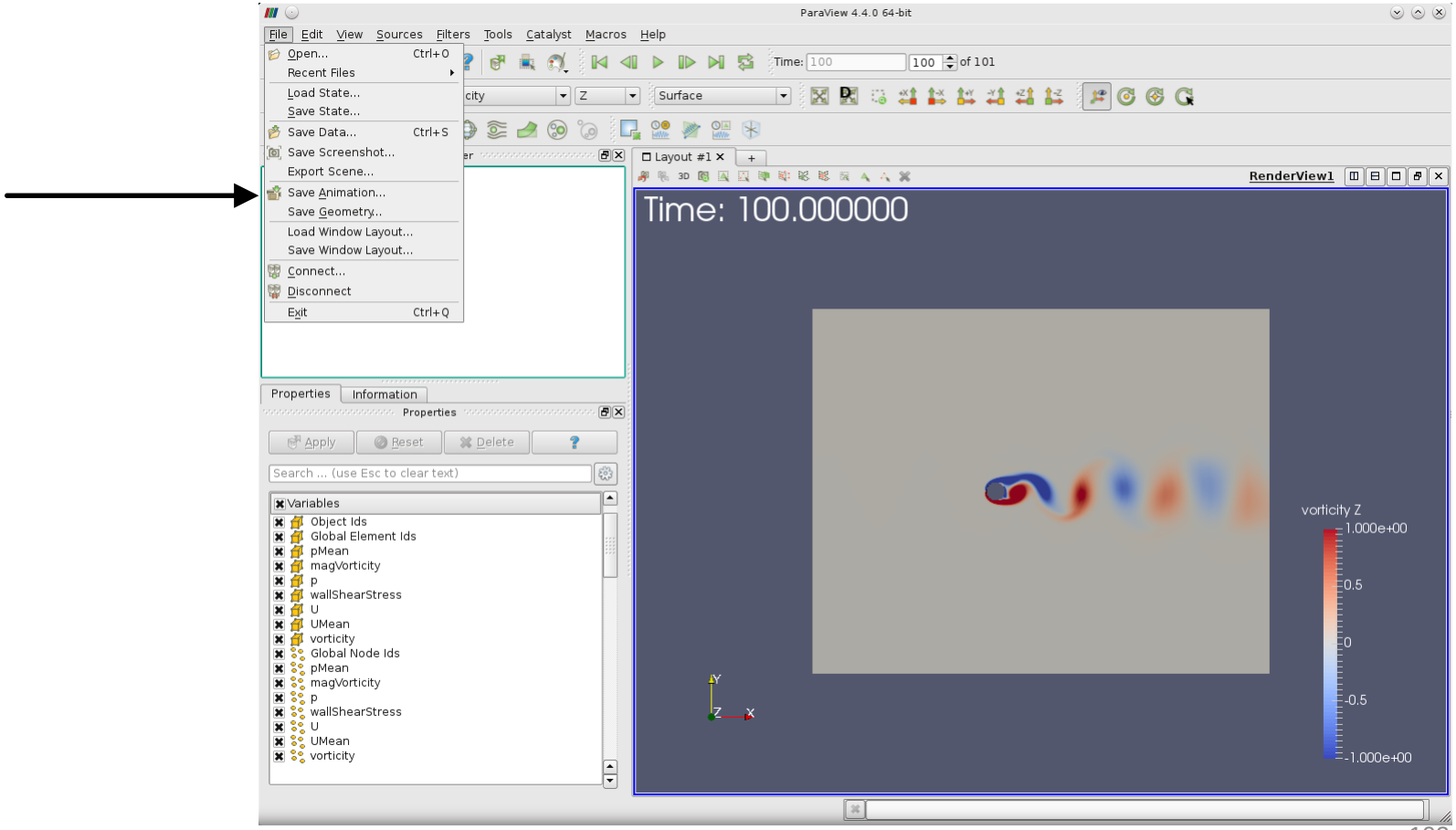
Animation



Scientific visualization with paraFoam/ParaView

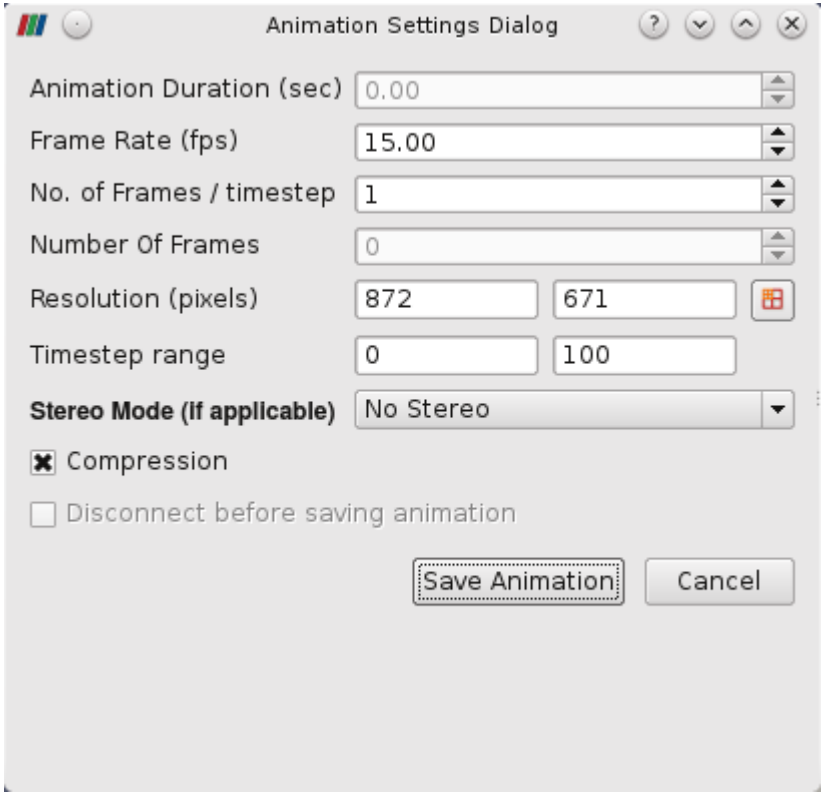
Saving animations

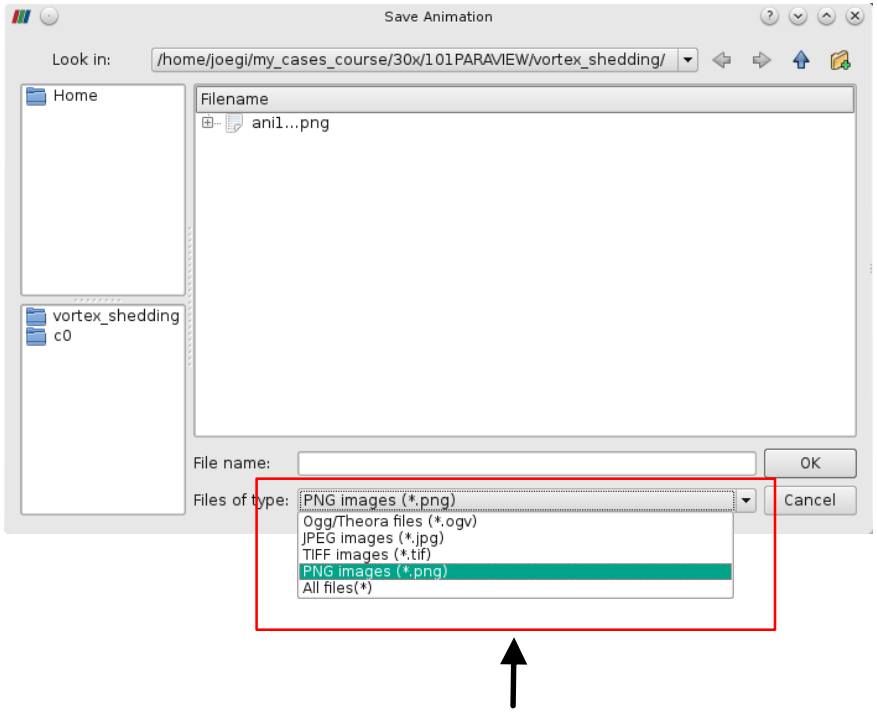
- Saving animations is straightforward.
- Select the menu `File → Save Animation`, set the animation settings, and save the animation.



Saving animations

- Saving animations is straightforward.
- Select the menu `File → Save Animation`, set the animation settings, and save the animation.





- Select a format.
- If you select a raster format (*.jpg, *.tif, *.png), it will save a series of screenshots.
- Then you can create an animated gif or a mpeg using the Linux command `convert`.

Creating an animated gif

- If you select a raster format (*.jpg, *.tif, *.png), ParaView will save a series of screenshots.
- You can create an animated gif or a mpeg using the Linux command `convert`.
- Assuming that you named the output as **out** and used the format ***.png**, ParaView will save a series of screenshots named *out.0000.png*, *out.0001.png*, … , *out.0100.png*
- To create the animated gif from the saved screenshots, type in the terminal

```
1. $> convert out.0*.png animation.gif
```

- With `convert` you can control compression format, orientation, delay time, size of the image and many more options. Feel free to read the documentation.

```
1. $> convert -help
```

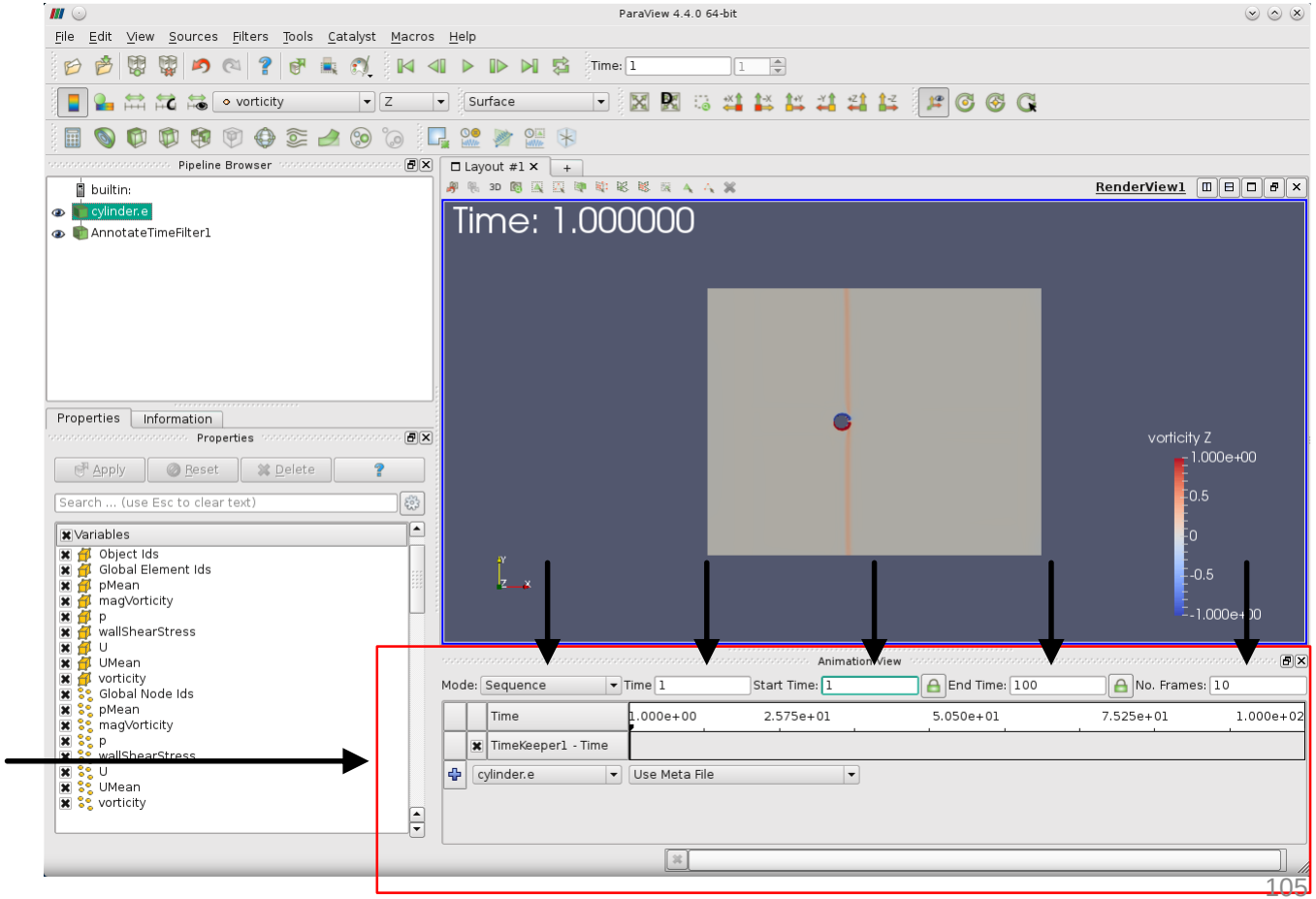
- If you use Windows, just find an equivalent open-source utility (there are many).

Scientific visualization with paraFoam/ParaView

Animation controls

- Select the menu `View → Animation View`.
- With the animation controls, you can control the speed of the animation.
- You can also animate cameras, filters and more.

The default animation mode is `Snap To TimeSteps`. It reads all the saved time-steps.



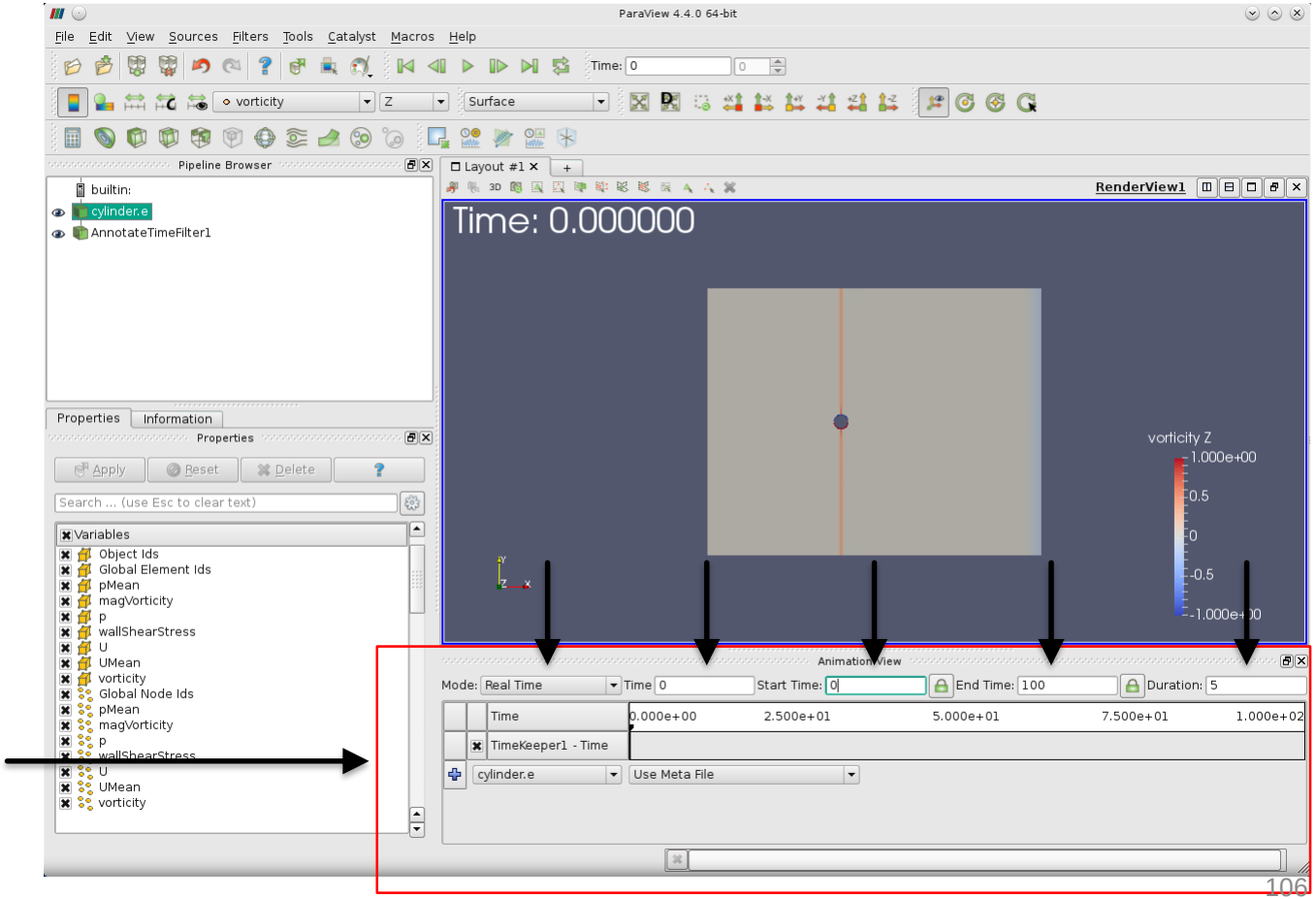
- Control animation by `Sequence`.
- The animation will start from frame 1 (`Start Time`), and will end at frame 100 (`End Time`).
- It will do the animation using 10 frames (`No. Frames`).

Scientific visualization with paraFoam/ParaView

Animation controls

- Select the menu `View → Animation View`.
- With the animation controls, you can control the speed of the animation.
- You can also animate cameras, filters and more.

The default animation mode is `Snap To TimeSteps`. It reads all the saved time-steps.

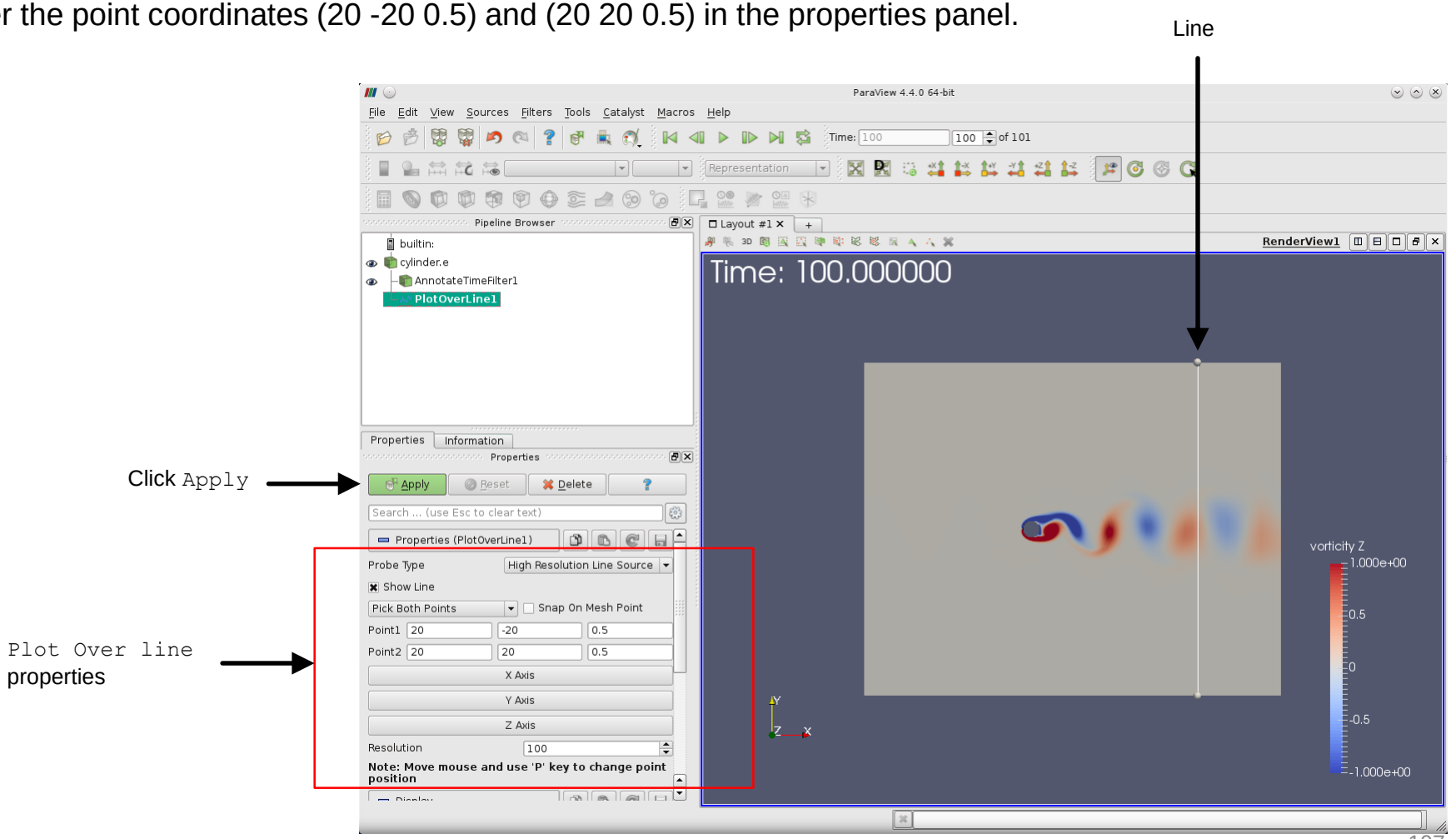


- Control animation by `Real Time`.
- The animation will start from time 0 (`Start Time`), and will end at time 100 (`End Time`).
- It will do the animation in 5 seconds (`Duration`).

Scientific visualization with paraFoam/ParaView

Sampling over a line

- Select the object `cylinder.e` and from the menu `Filters → Alphabetical` or the toolbar, select the `Plot Over Line` filter 
- Enter the point coordinates (20 -20 0.5) and (20 20 0.5) in the properties panel.

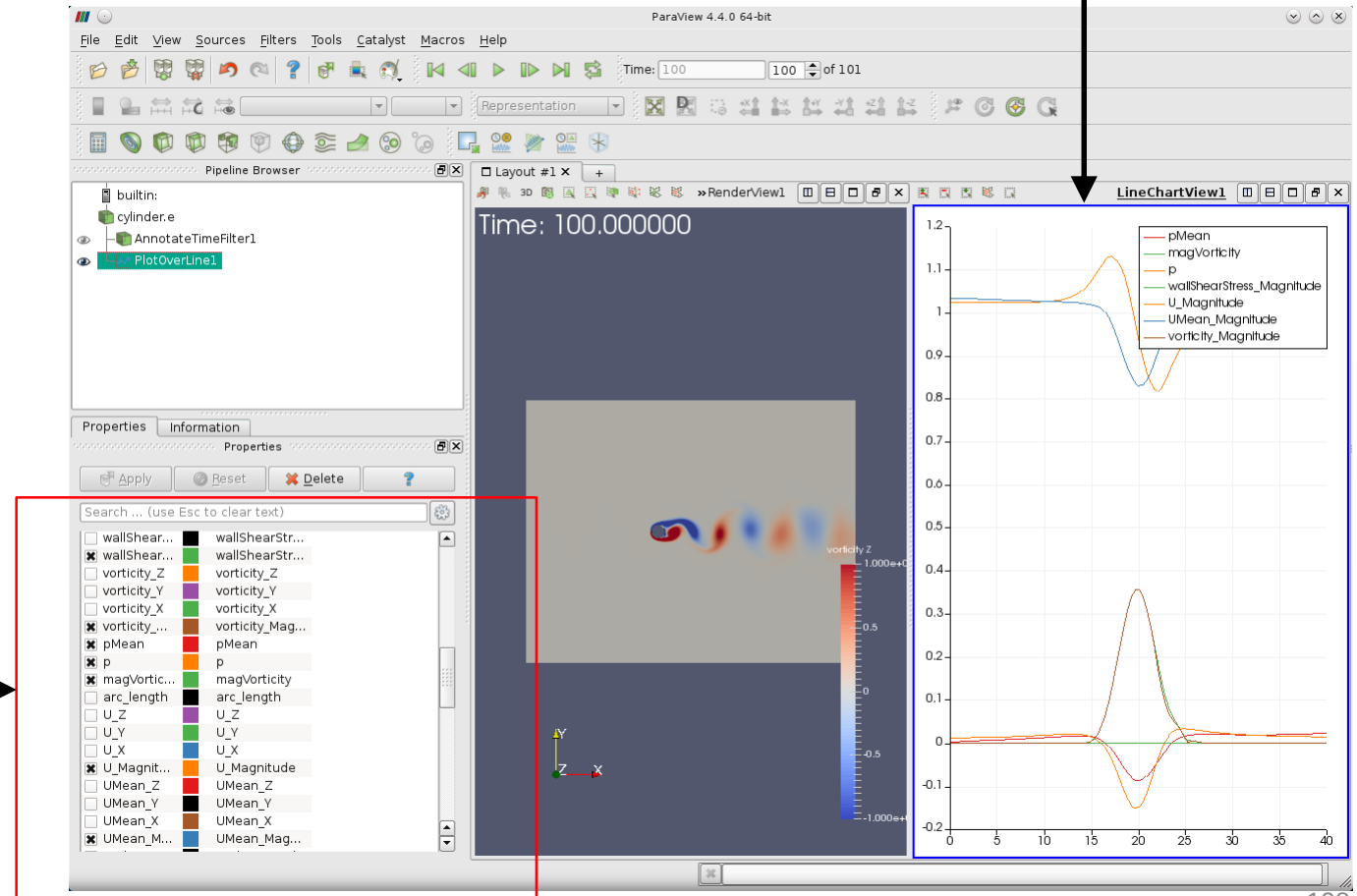


Scientific visualization with paraFoam/ParaView

Sampling over a line

- After clicking the Apply button, ParaView will open another view with the line plots.
- Set the line chart view active and select the variables to plot.

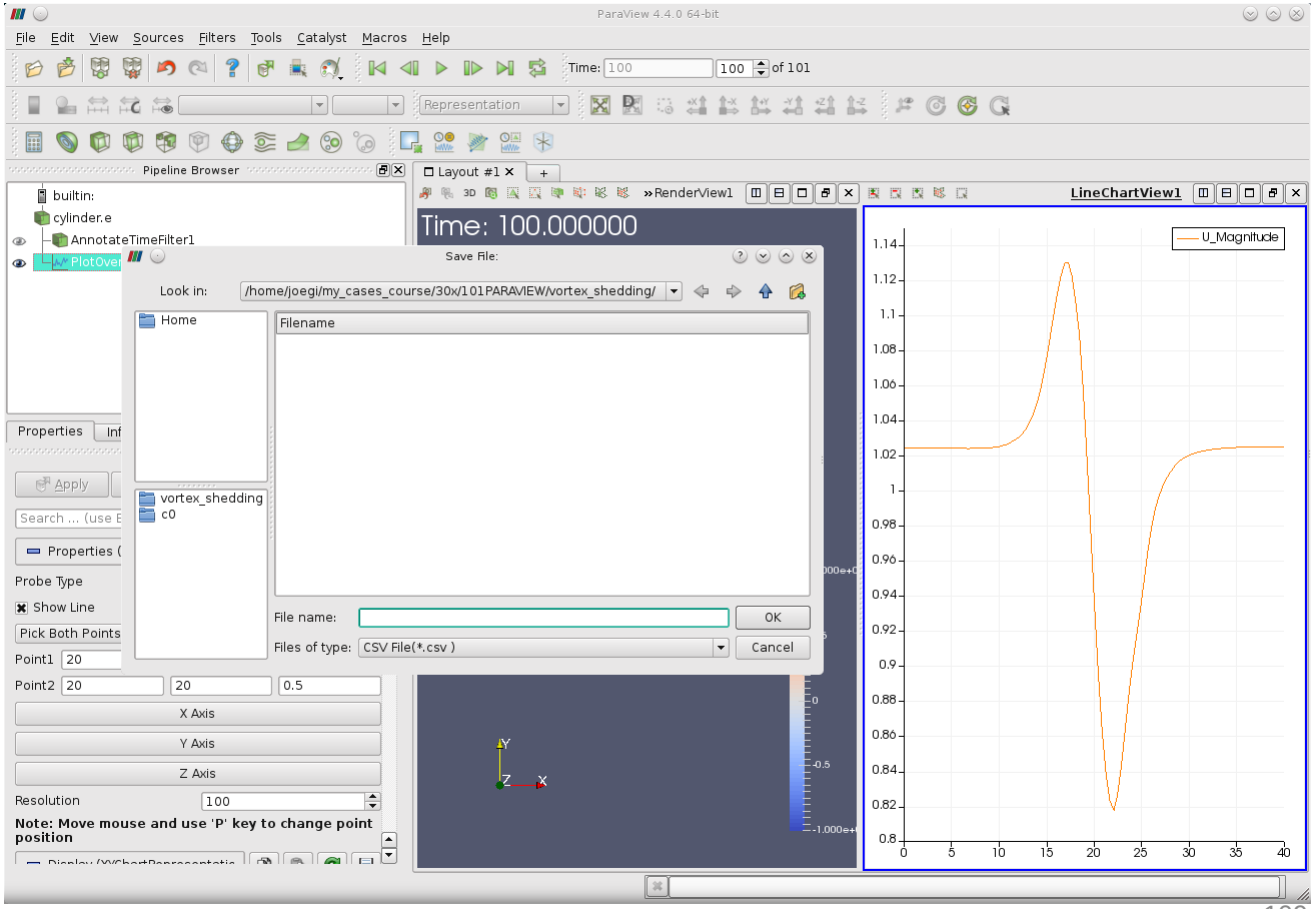
Active view
`LineChartView1`



- `Plot Over line` properties.
- Select the variables to plot.

Sampling over a line

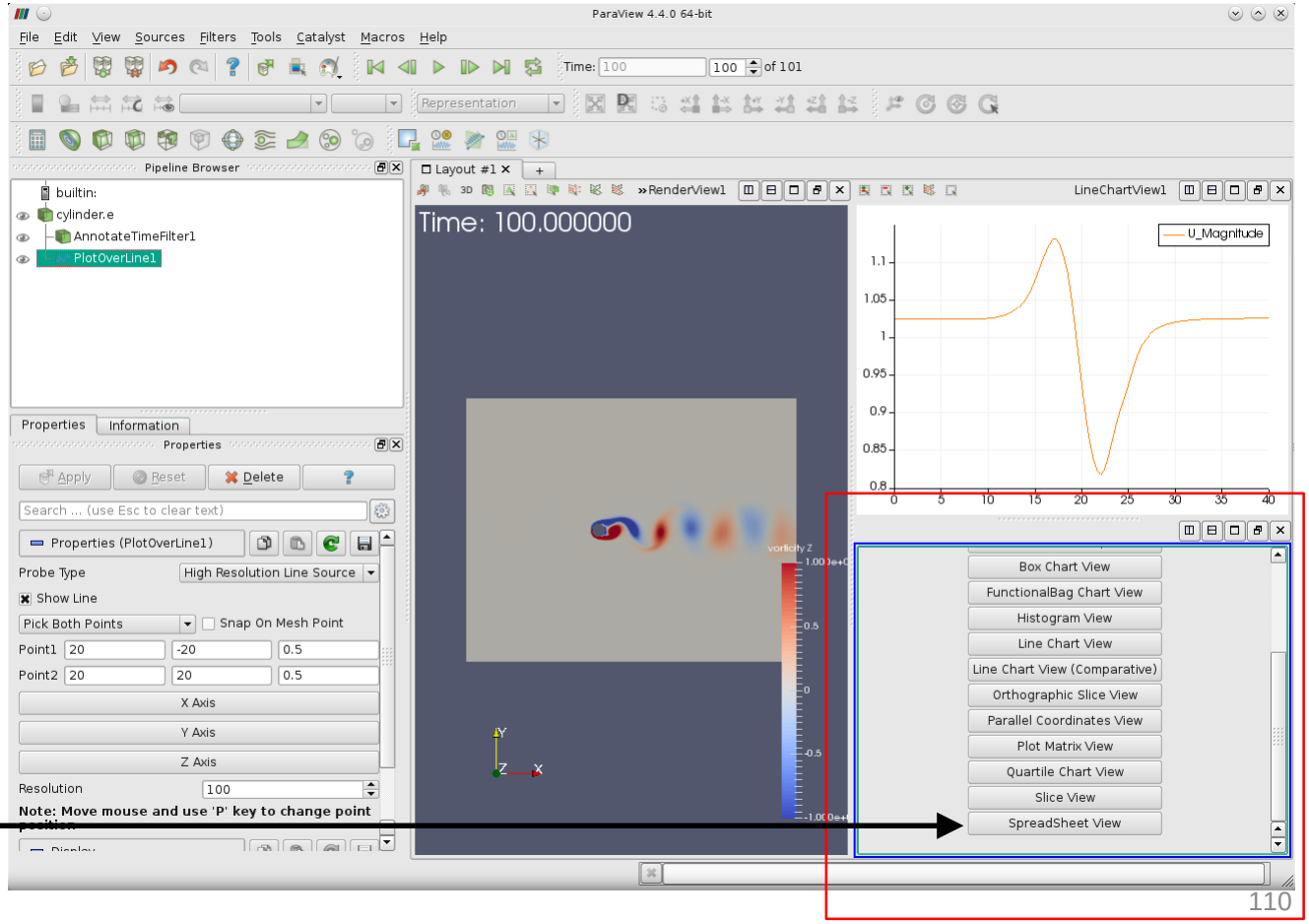
- Saving the sampled values is straightforward.
- Set the view `LineChartView1` active, select the menu `File → Save Data`, and save the data in CSV format.



Scientific visualization with paraFoam/ParaView

Sampling over a line

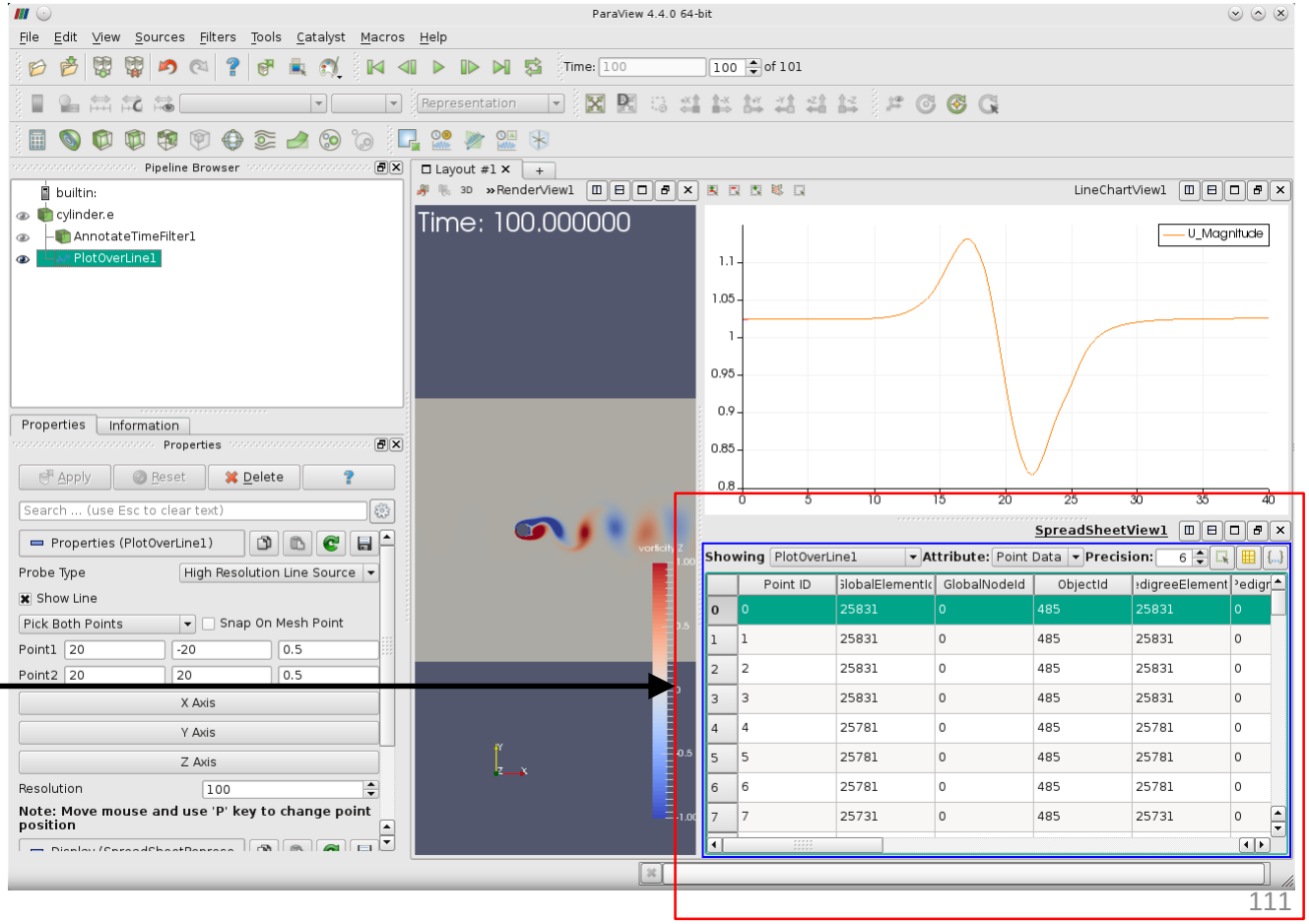
- To display the tabular values, create a new view and select `SpreadSheet View`.



Scientific visualization with paraFoam/ParaView

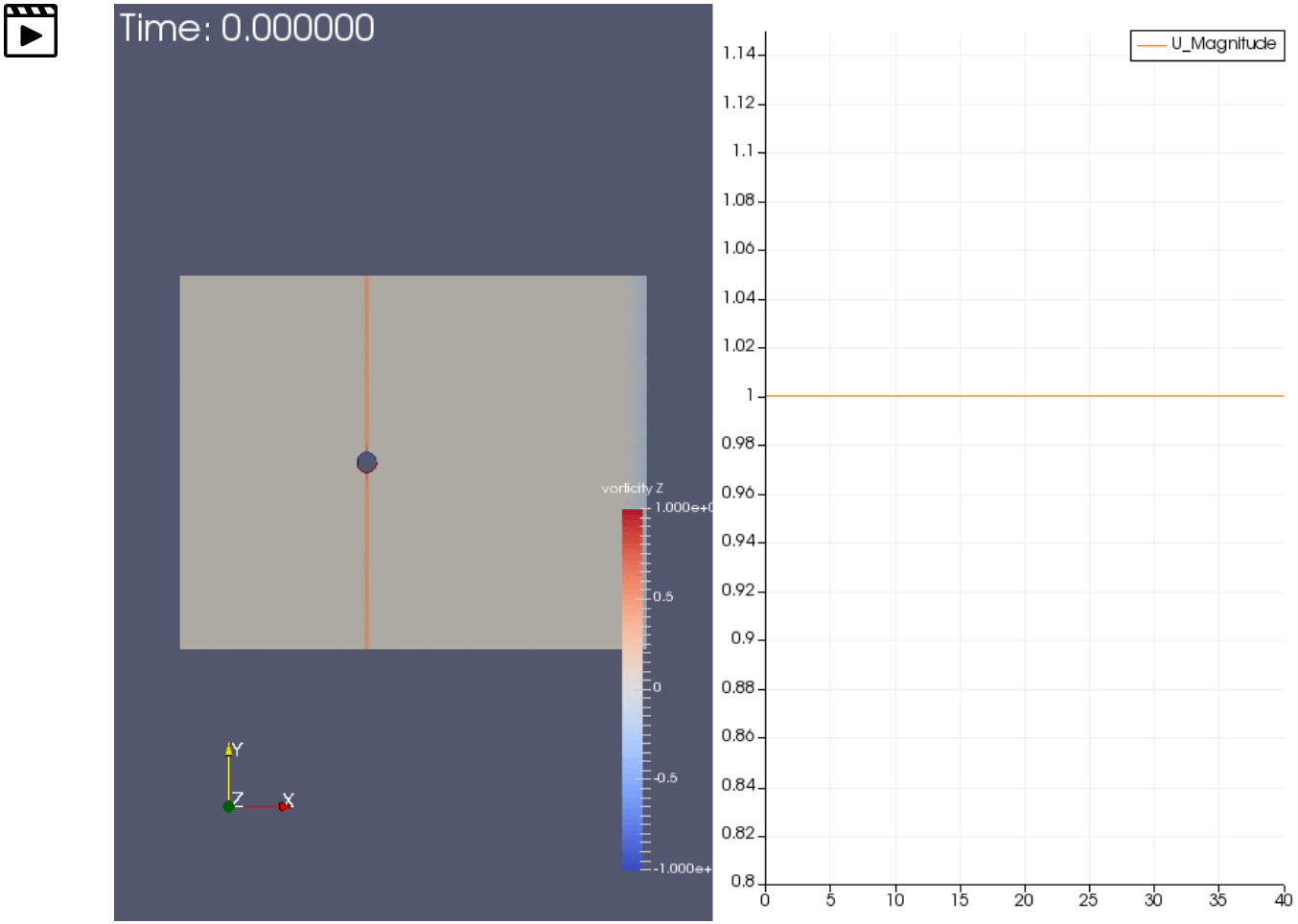
Sampling over a line

- To display the tabular values, create a new view and select `SpreadSheet View`.
- The tabular values of the sampled data are displayed in the view `SpreadSheetView1`.
- To save the data, select the menu `File → Save Data` and save the data in CSV format.



Sampling over a line

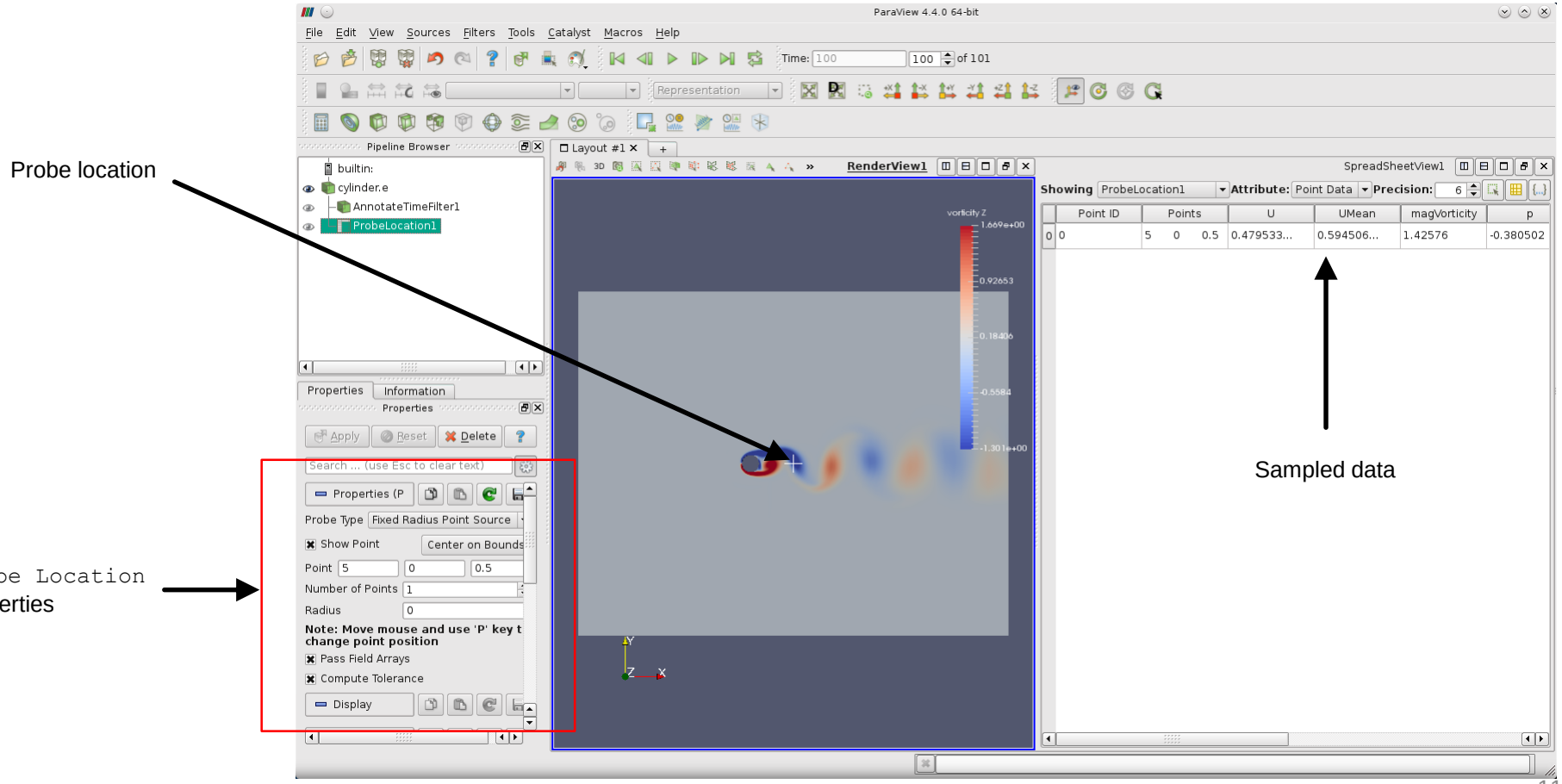
- To animate the sampled data, just press the play button in the `VCR Controls` toolbar.



Scientific visualization with paraFoam/ParaView

Probe a location

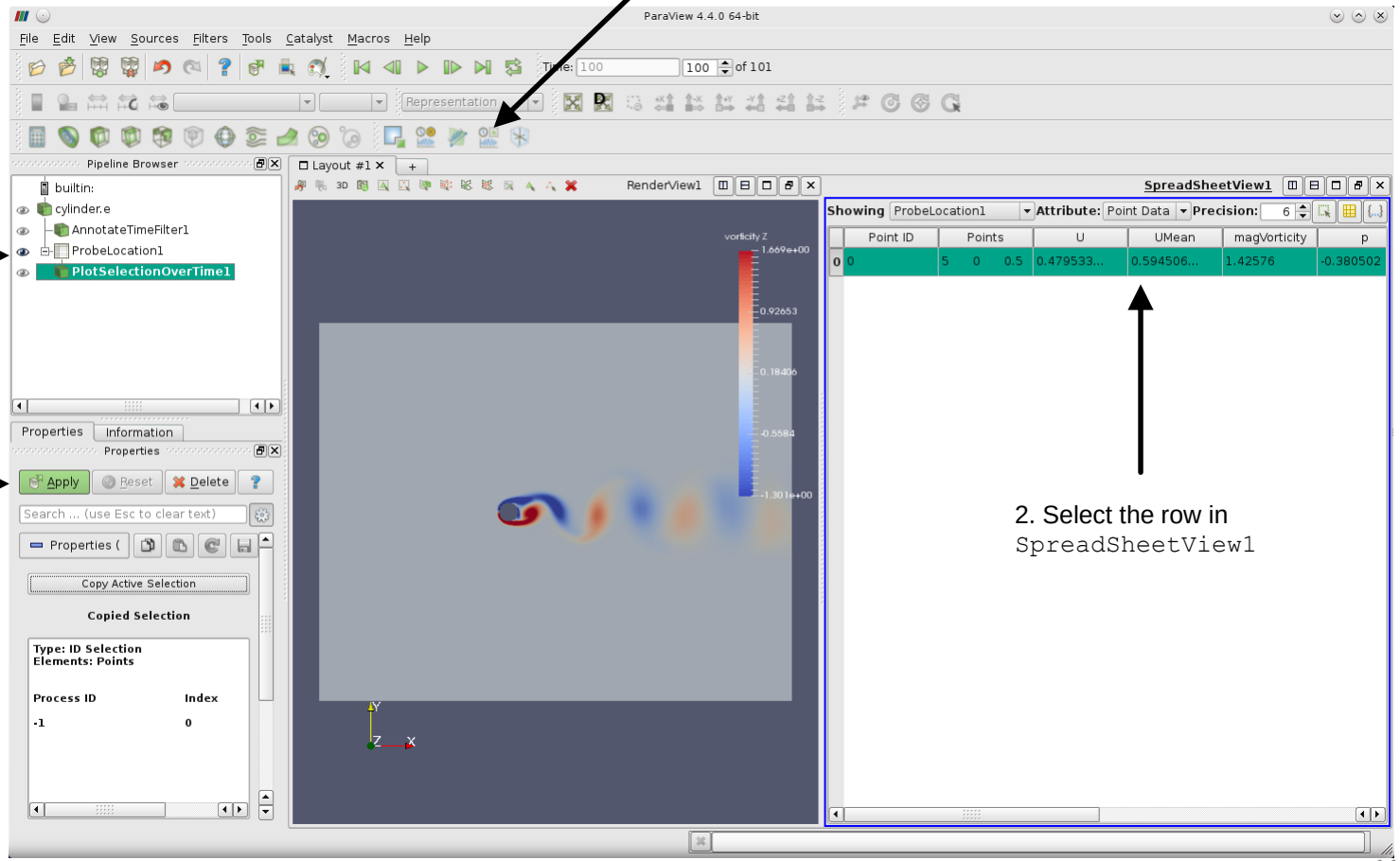
- Select the object `cylinder.e` and from the menu `Filters → Alphabetical` or the toolbar, select the `Probe Location` filter 
- Enter the point coordinates (5 0 0) in the properties panel.



Scientific visualization with paraFoam/ParaView

Plotting probed data

- Select the object `ProbeLocation1` and from the menu `Filters → Alphabetical` or the toolbar, select the `Plot Selection Over Time` filter 

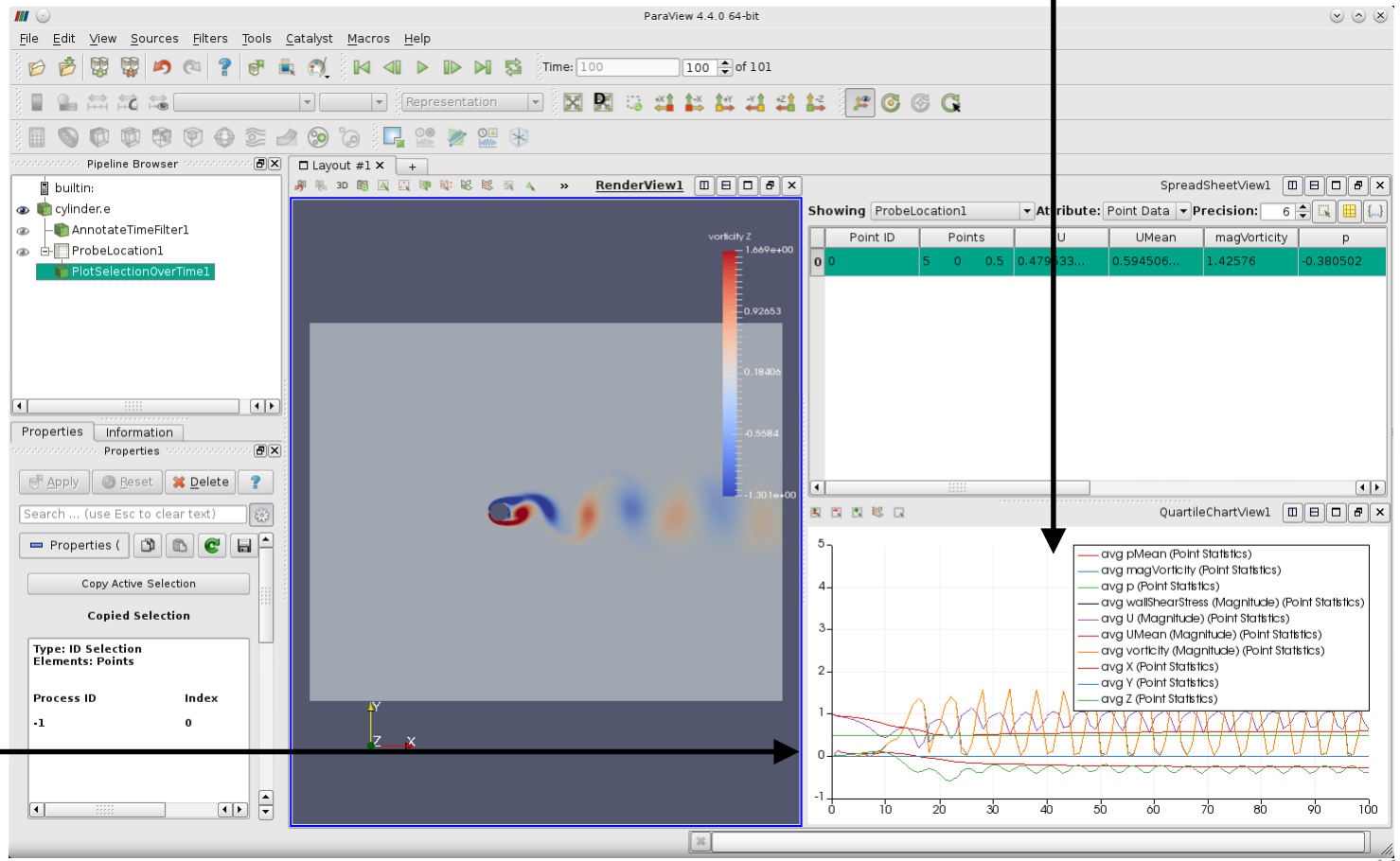


Scientific visualization with paraFoam/ParaView

Plotting probed data

- Select the object `ProbeLocation1` and from the menu `Filters → Alphabetical` or the toolbar, select the `Plot Selection Over Time` filter 

Click on the view to set it to active

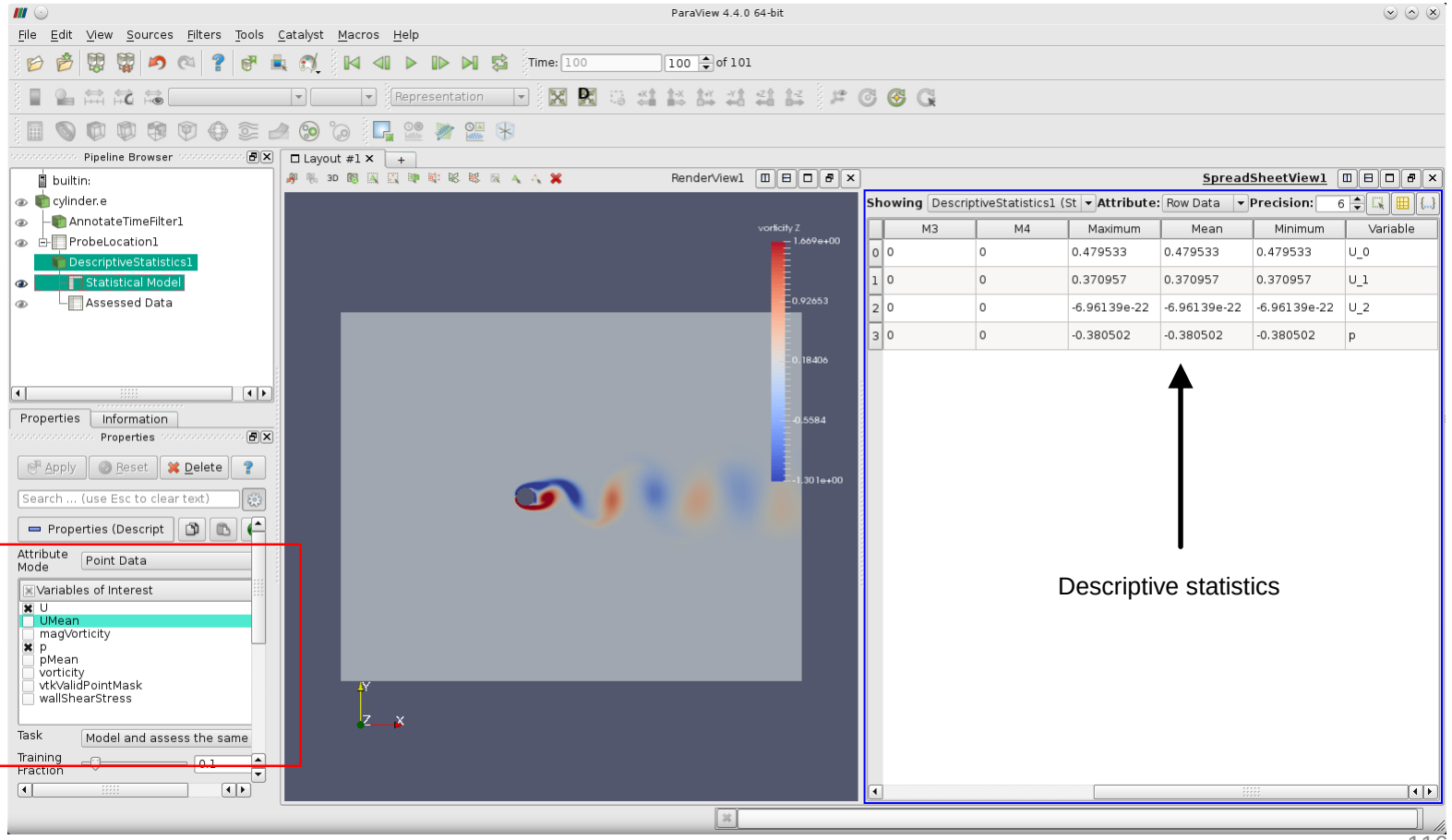


- Plot of selection
- To access the properties set the view to active

Scientific visualization with paraFoam/ParaView

Computing descriptive statistics of probed data

- Select the object `ProbeLocation1` and from the menu `Filters → Alphabetical` select the `Descriptive Statistics` filter

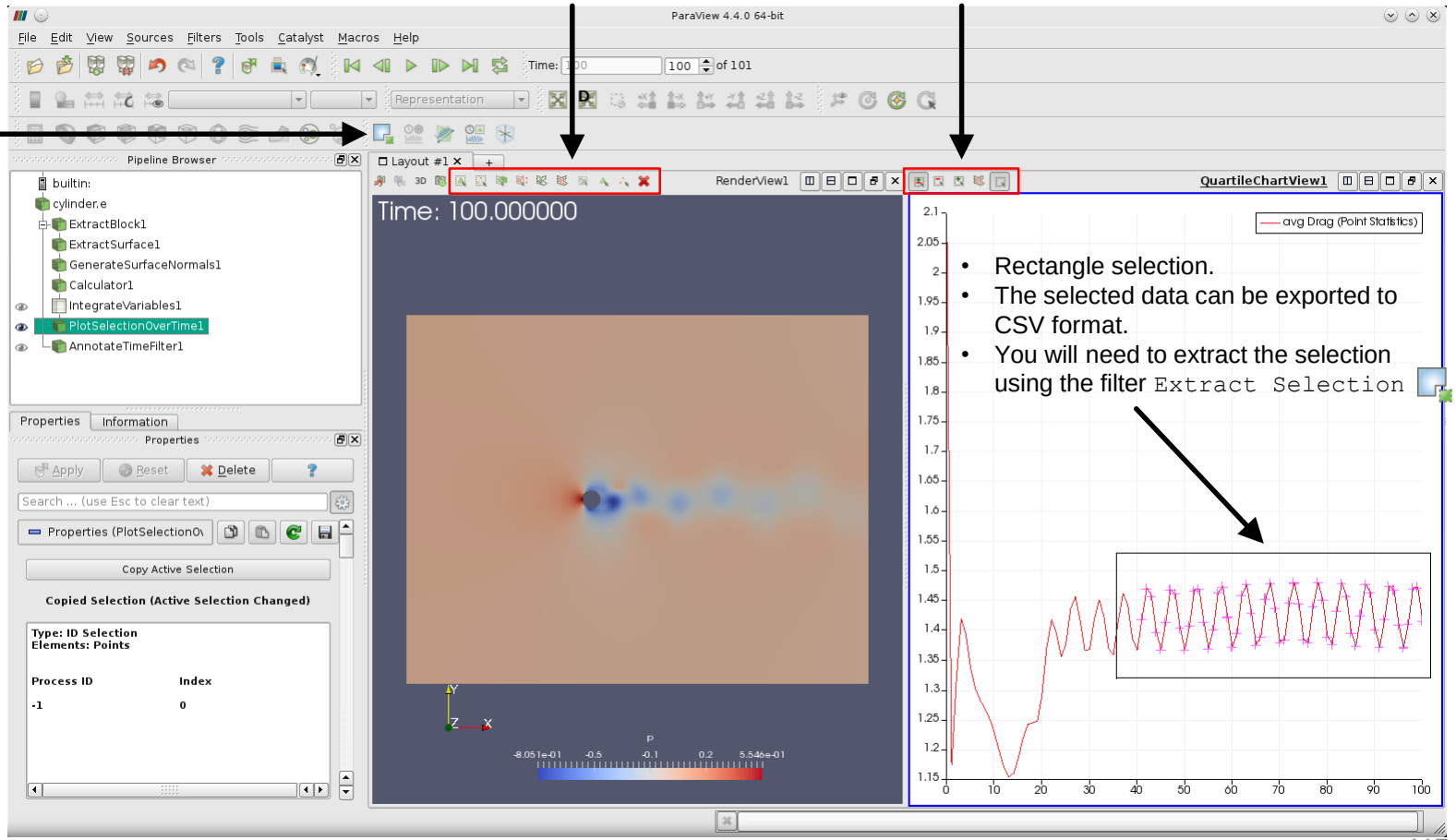


Scientific visualization with paraFoam/ParaView

Selections

- To select data (in the 3D view, line chart view or spread sheet view), use the selection buttons in the small toolbar in the render view.
- Play around with the different methods available. In this case we used rectangle selection in the `QuartileChartView1` view.

`Extract Selection` filter



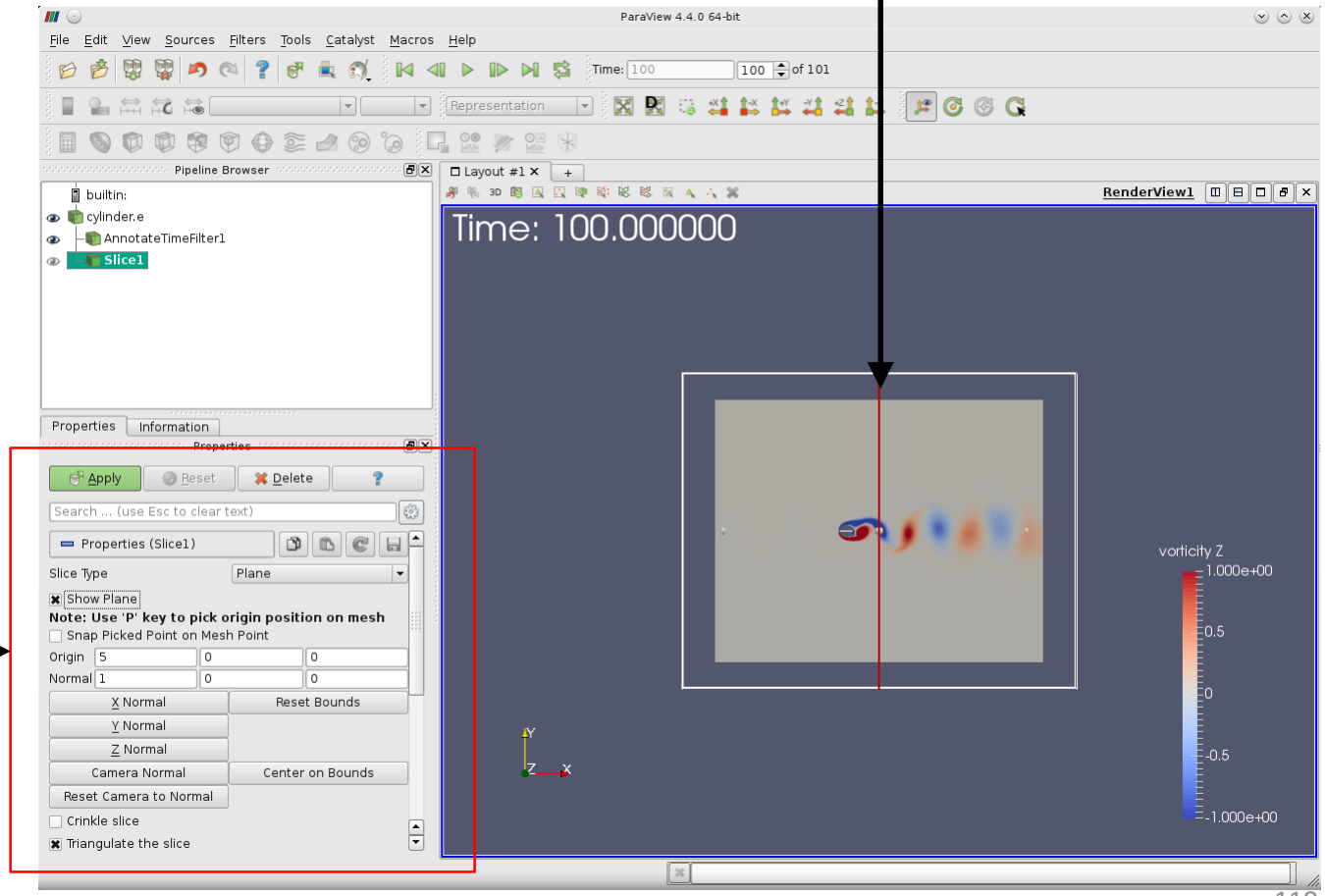
Scientific visualization with paraFoam/ParaView

Computing mass flow through a plane

- Select the object `cylinder.e`, and from the menu `Filters → Alphabetical` or the toolbar, select the `Slice` filter 
- Create a plane normal to X and set the origin to (5 0 0).

`Slice1`

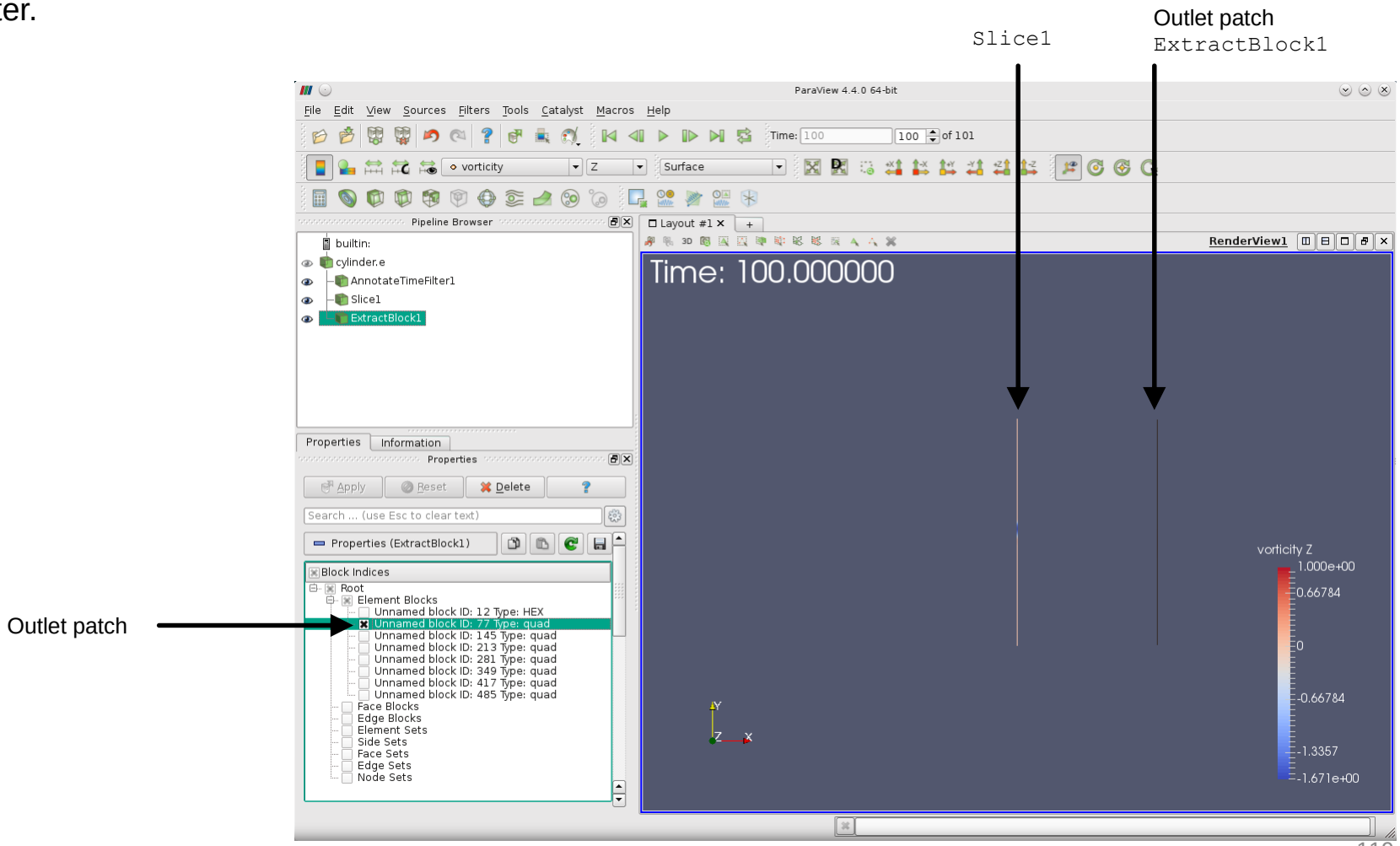
`Slice` filter properties



Scientific visualization with paraFoam/ParaView

Computing mass flow through a plane

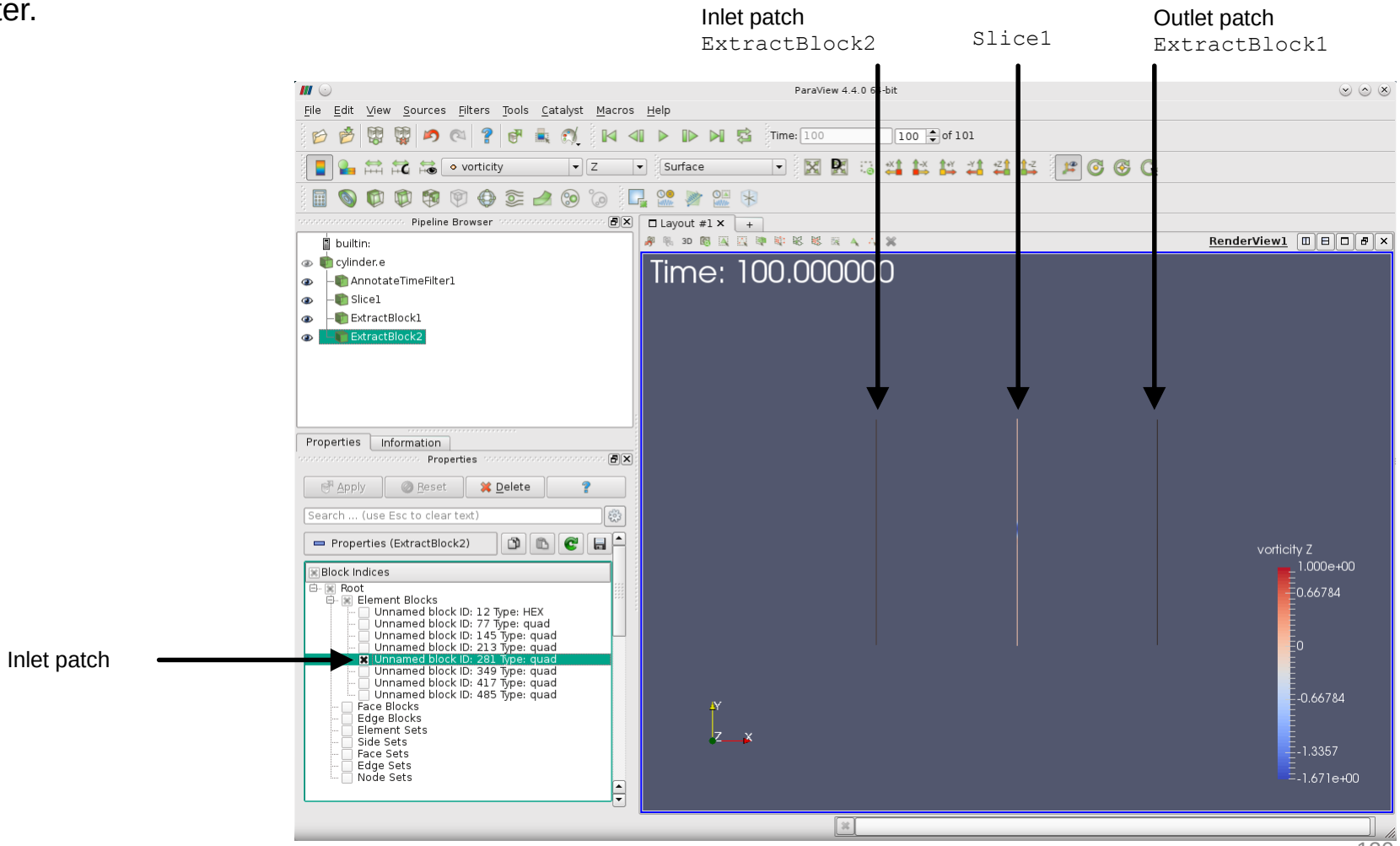
- Use the filter `Extract Block` to extract the boundary patch outlet.
- Select the object `cylinder.e`, and from the menu `Filters → Alphabetical` select the `Extract Block` filter.



Scientific visualization with paraFoam/ParaView

Computing mass flow through a plane

- Use the filter `Extract Block` to extract the boundary patch inlet.
- Select the object `cylinder.e`, and from the menu `Filters → Alphabetical` select the `Extract Block` filter.

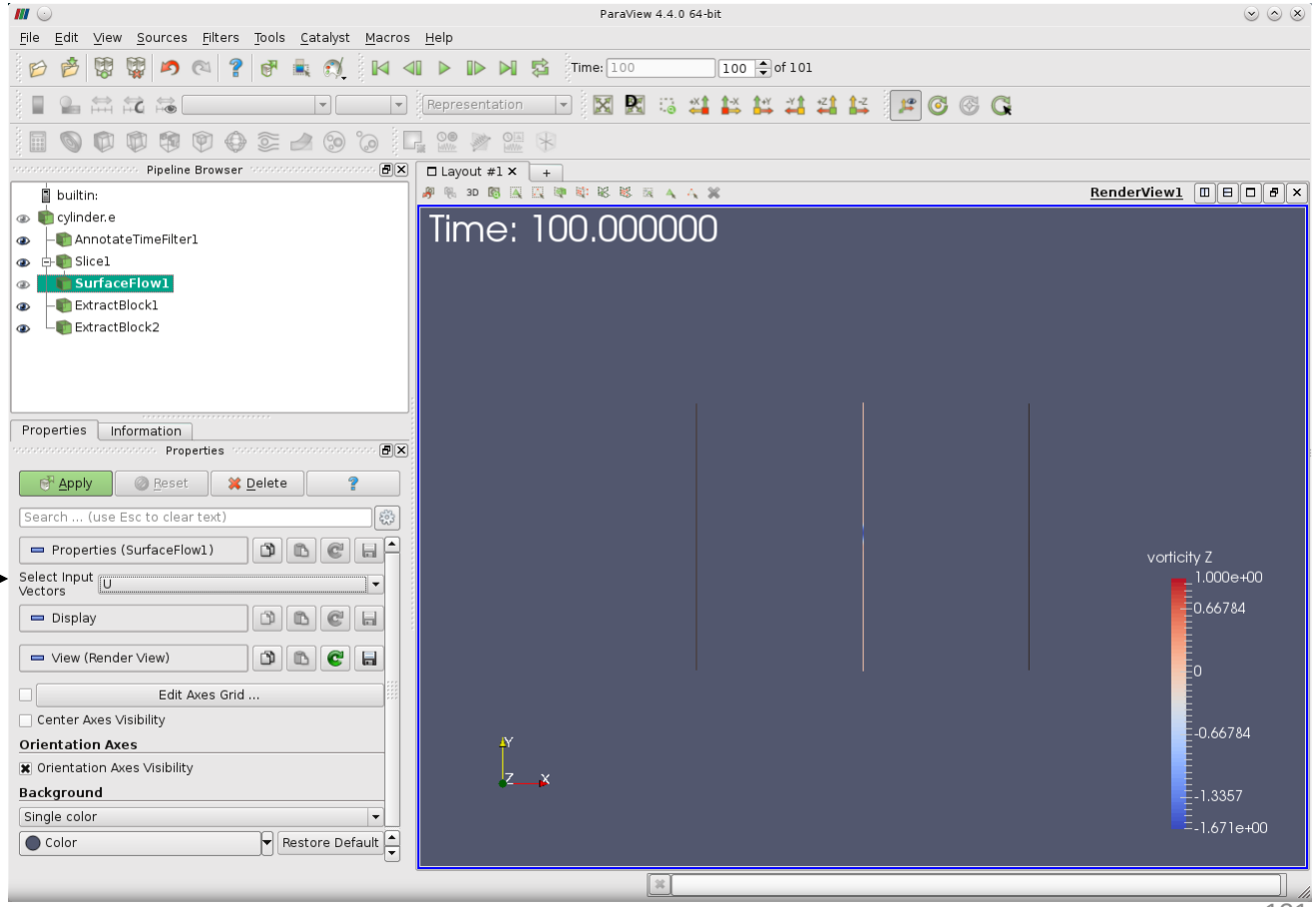


Scientific visualization with paraFoam/ParaView

Computing mass flow through a plane

- Select the object `Slice1`, and from the menu `Filters → Alphabetical` select the `Surface Flow` filter and select `U` as input vector.

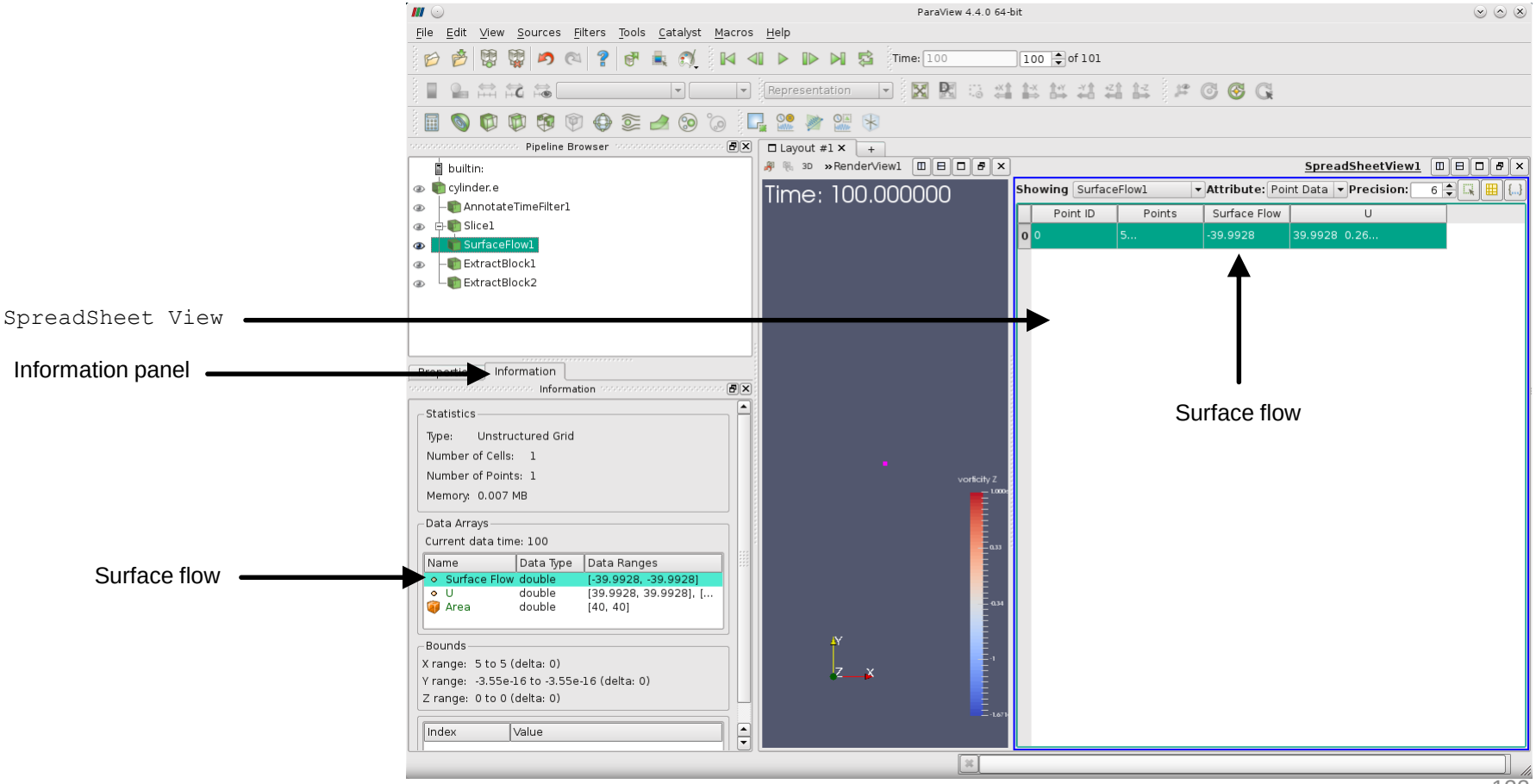
Select `U`



Scientific visualization with paraFoam/ParaView

Computing mass flow through a plane

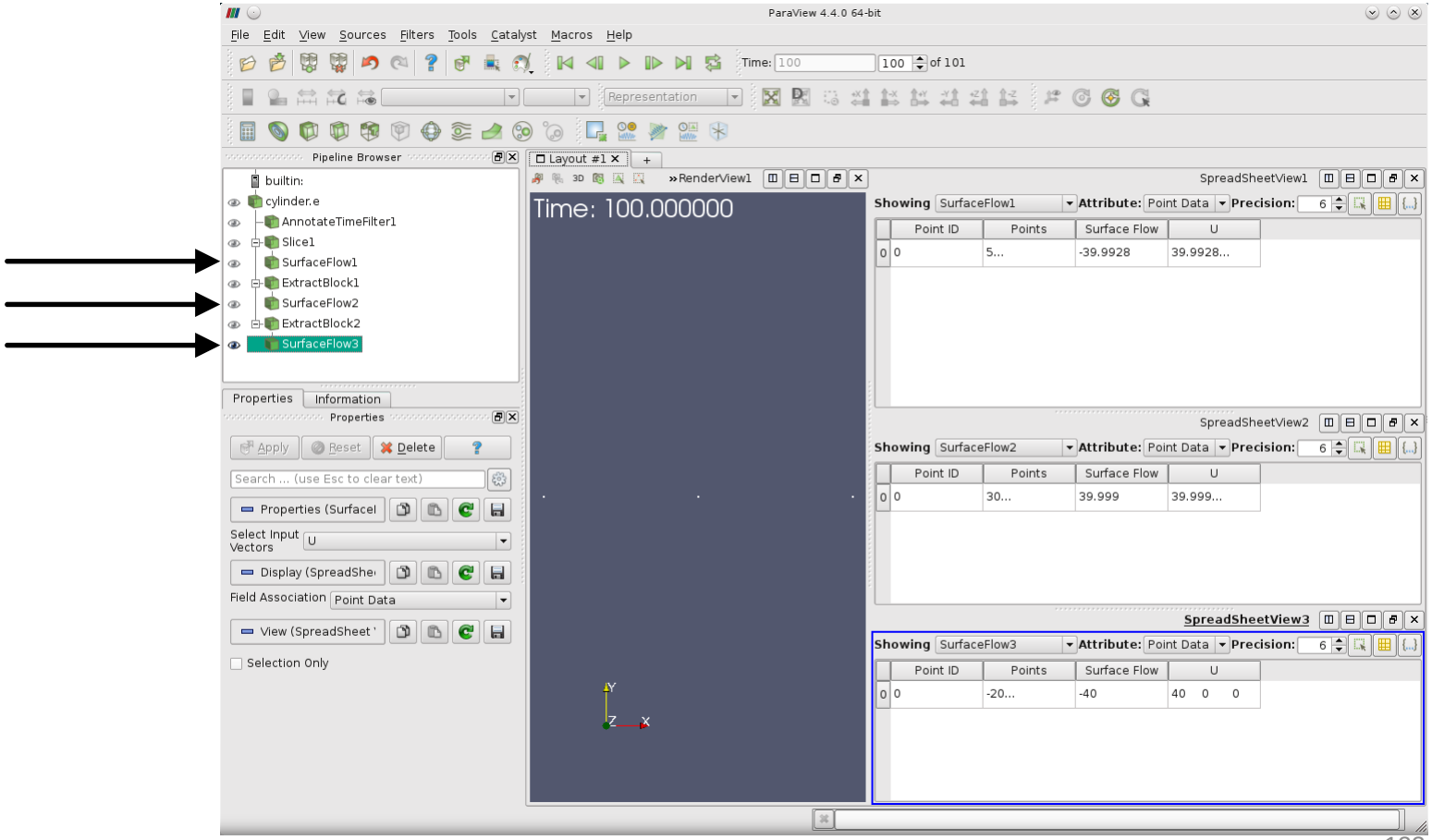
- Select the object `Slice1`, and from the menu `Filters → Alphabetical` select the `Surface Flow` filter and select `U` as input vector.
- The mass flow information can be accessed via the `SpreadSheet View` or the information panel.



Scientific visualization with paraFoam/ParaView

Computing mass flow through a plane

- At this point, compute the mass flow for the objects `ExtractBlock2` and `ExtractBlock3`.
- The filter `Surface Flow` works with arbitrary planes and shapes. It is a very powerful alternative to **functionObjects** in OpenFOAM®.



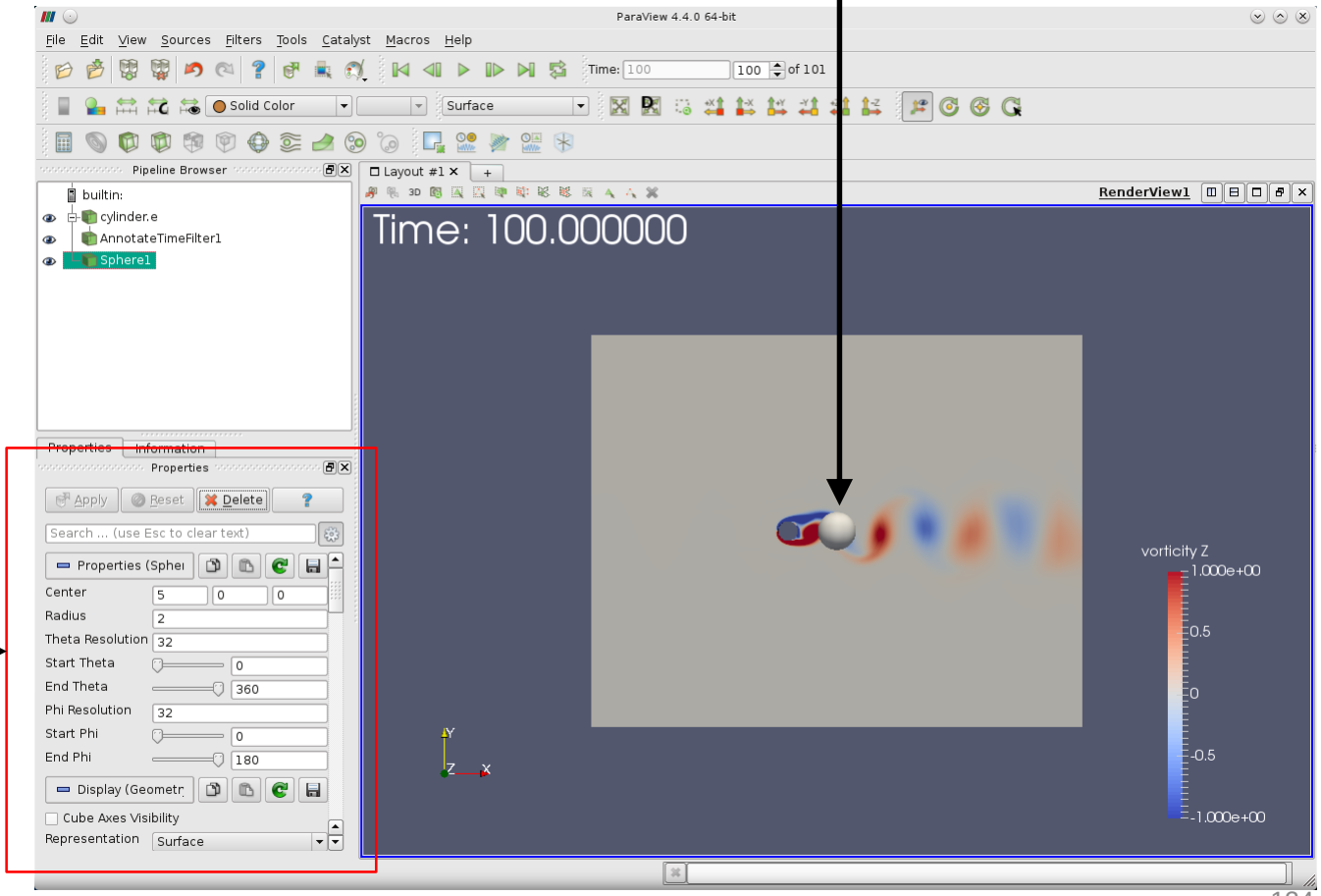
Scientific visualization with paraFoam/ParaView

Mapping the solution to a source

- Go to the `Sources` menu and create a `Sphere`.
- An item named `Sphere1` is added to and selected in the pipeline browser.
- Use the values shown in the properties panel.

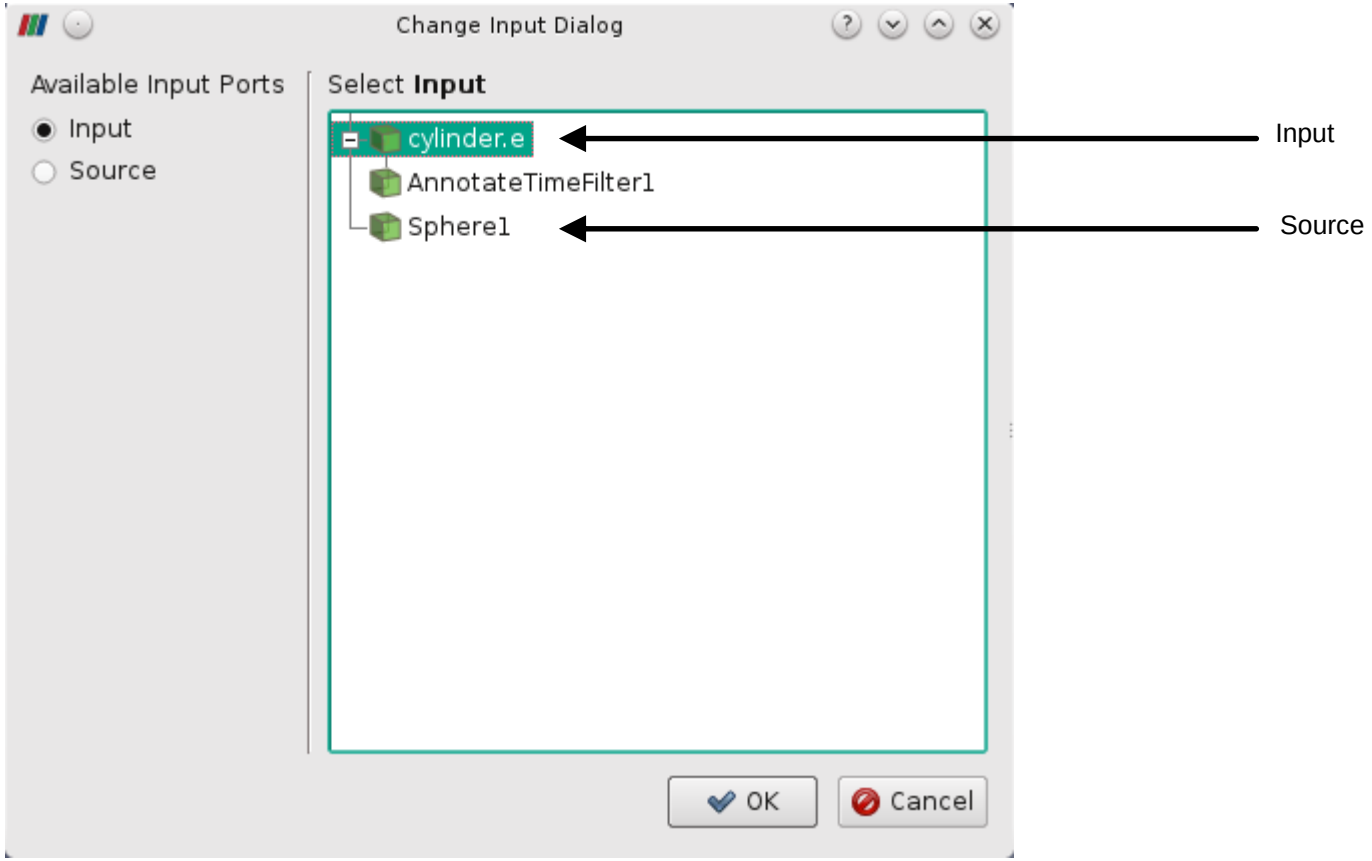
`Sphere1`

`Sphere` source properties



Mapping the solution to a source

- From the menu `Filters → Alphabetical` select the `Resample With Dataset` filter.
- As Input use the object `cylinder.e` and as a source use the object `Sphere1`.
- We are mapping the solution from `cylinder.e` to `Sphere1`.



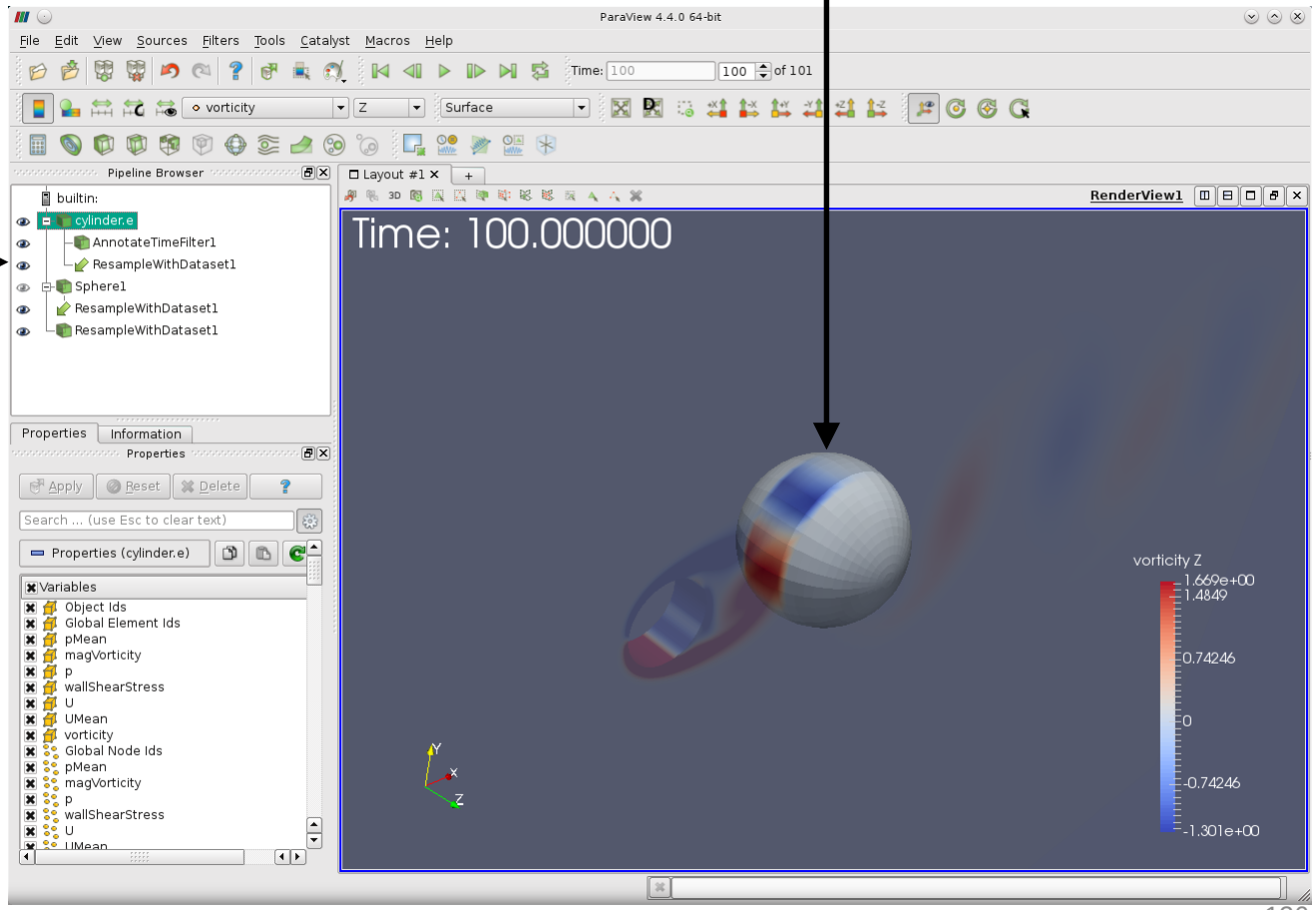
Scientific visualization with paraFoam/ParaView

Mapping the solution to a source

- The filter `Resample With Dataset` mapped the solution from the object `cylinder.e` to the object `Sphere1`.

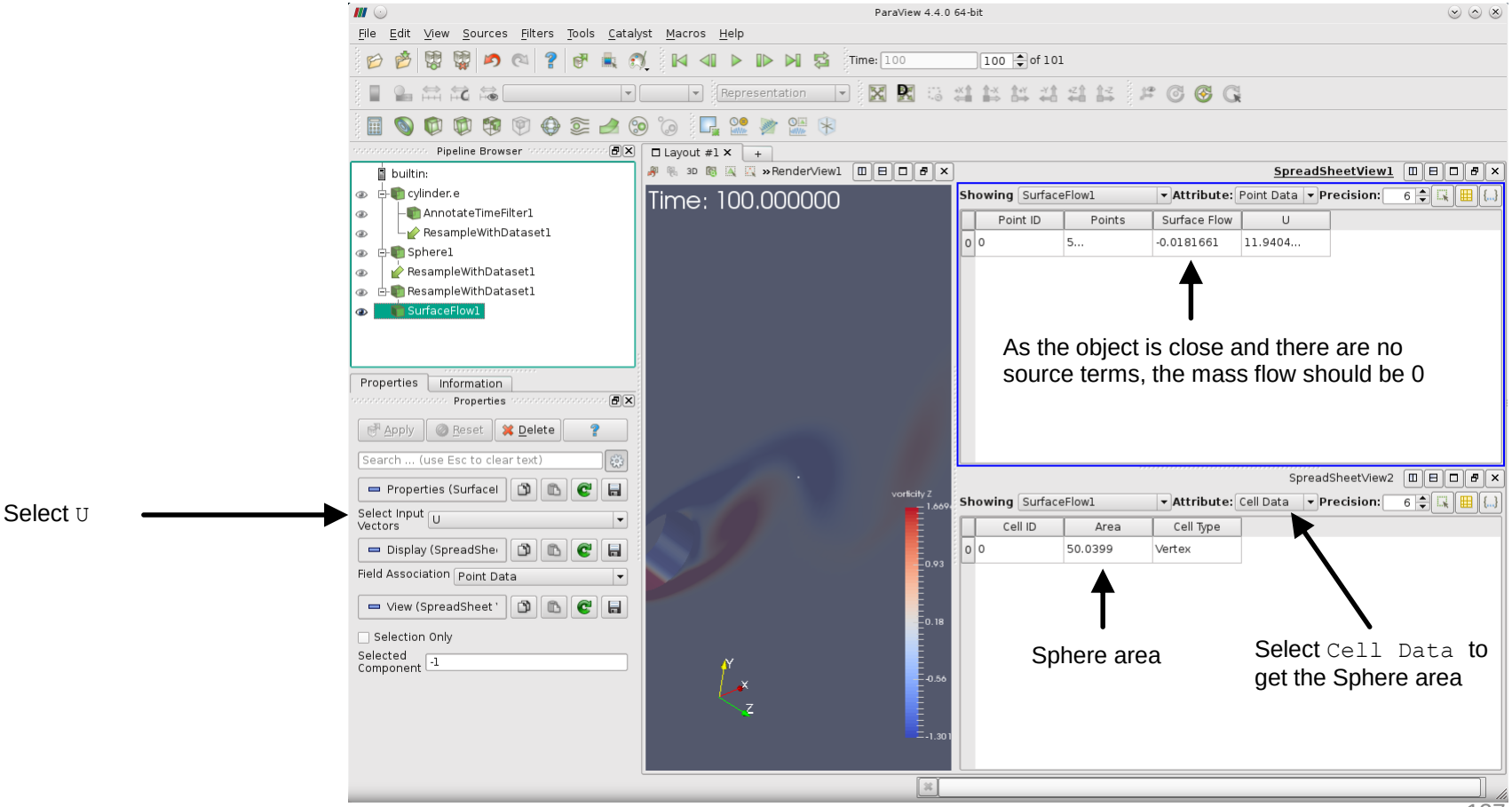
`Sphere1` with the mapped solution

The pipeline hierarchy is automatically generated



Mapping the solution to a source

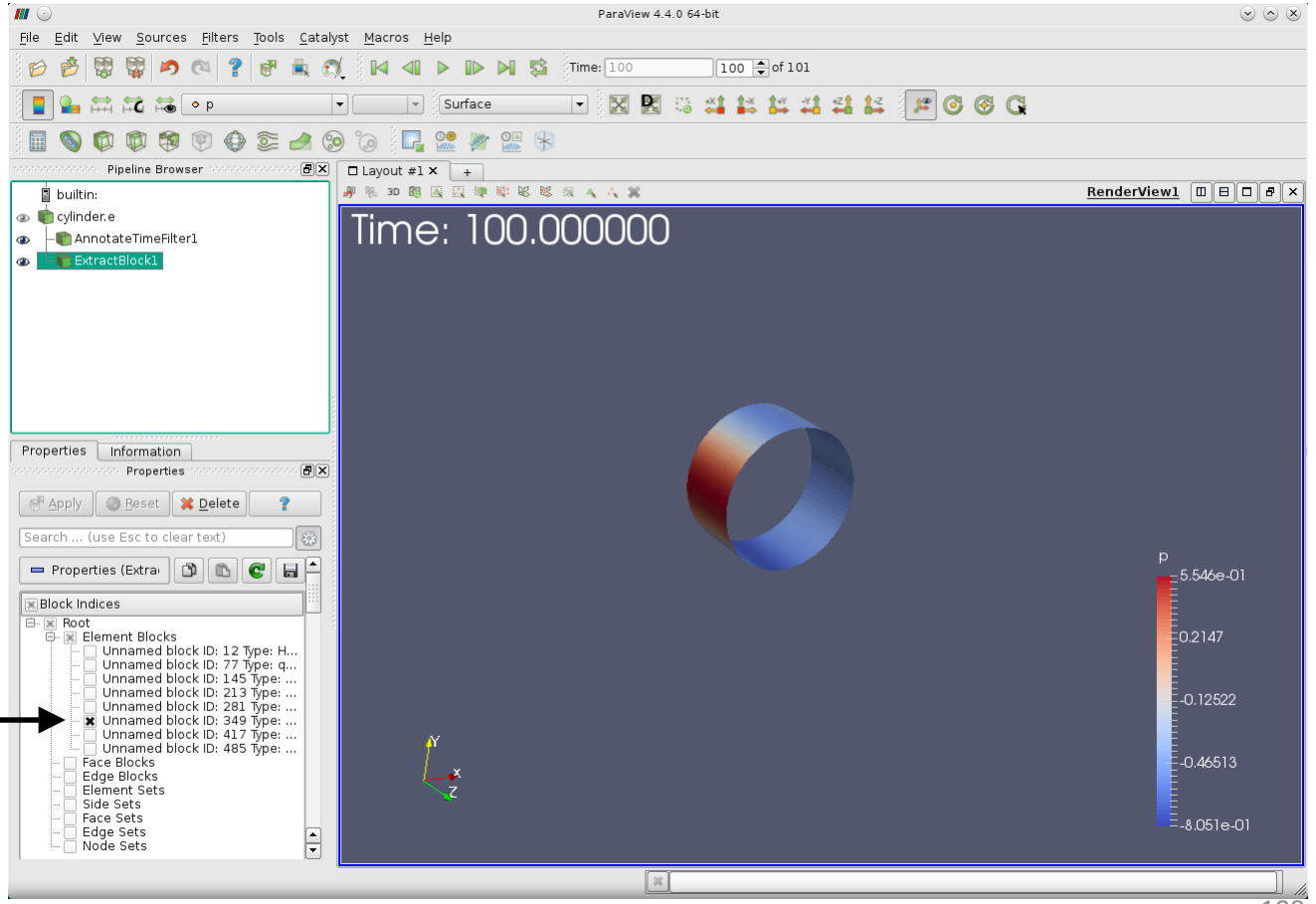
- At this point, let's compute the mass flow on the newly created object .
- Select the object `ResampleWithDataset1`, and from the menu `Filters → Alphabetical` select the `Surface Flow` filter.



Scientific visualization with paraFoam/ParaView

Computing the forces on the cylinder

- Use the filter `Extract Block` to extract the cylinder patch.
- Select the object `cylinder.e`, and from the menu `Filters → Alphabetical` select the `Extract Block` filter.



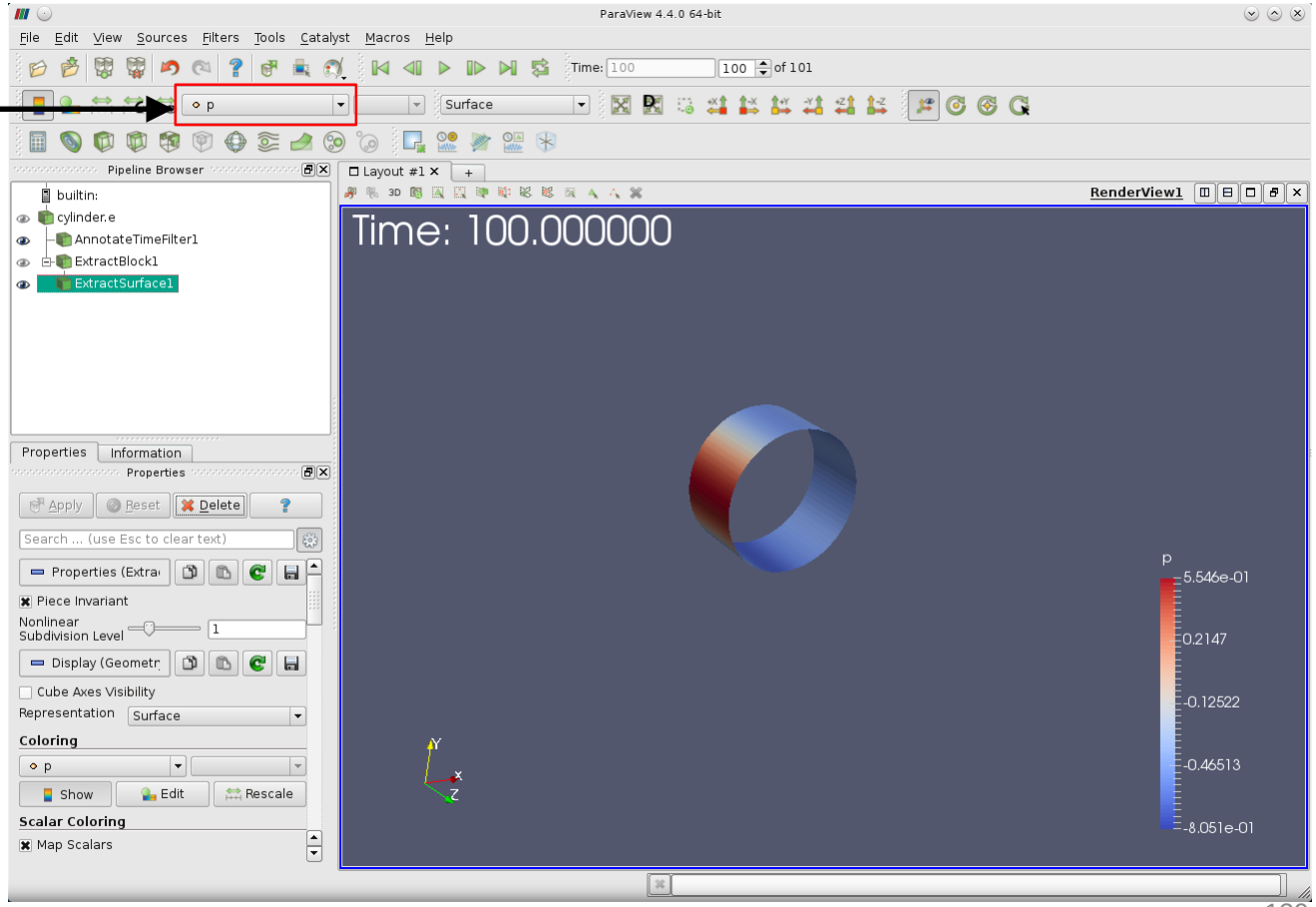
Cylinder

Scientific visualization with paraFoam/ParaView

Computing the forces on the cylinder

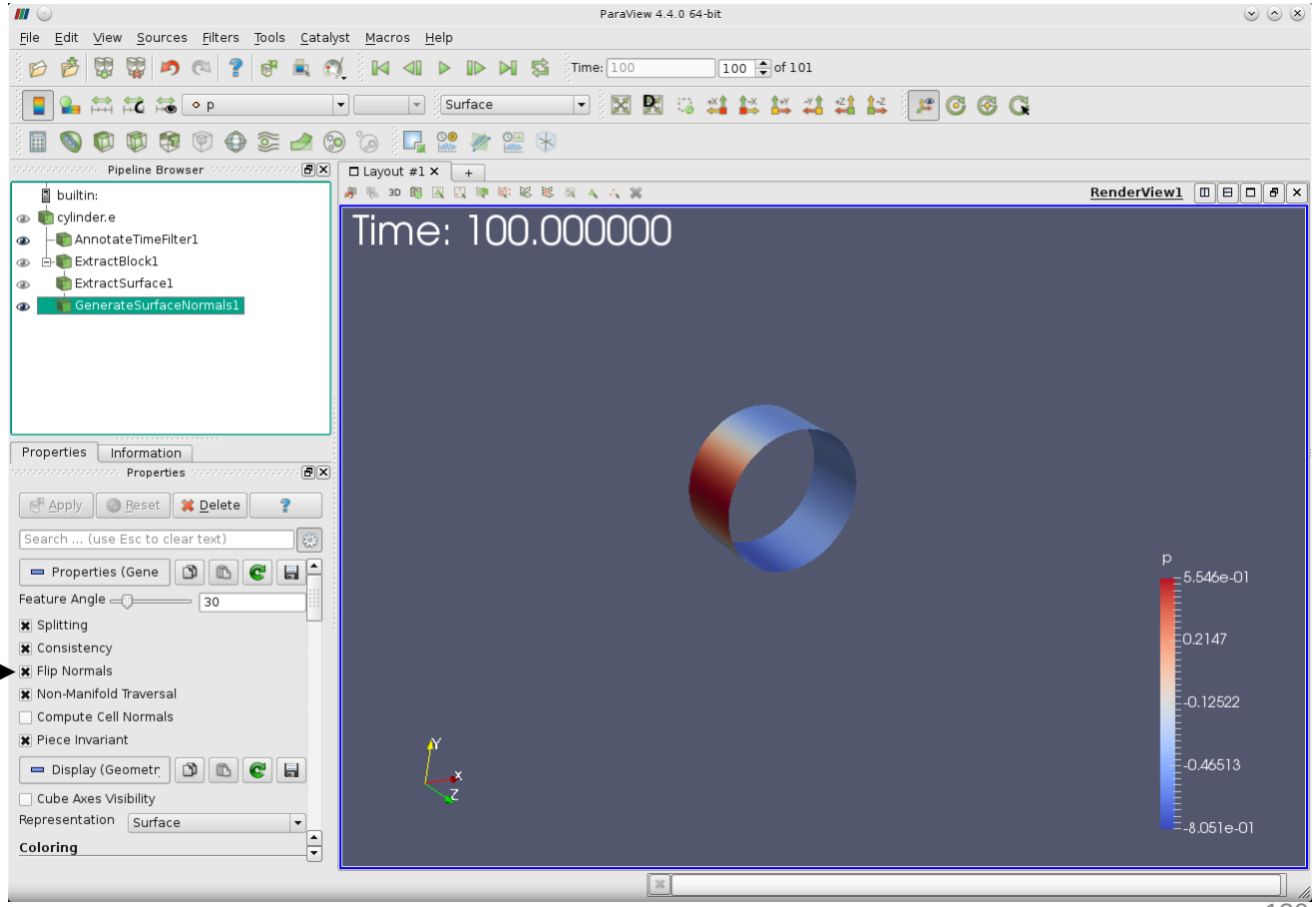
- Select the object `ExtractBlock1`, and from the menu `Filters → Alphabetical` select the `Extract Surface` filter to extract the boundary surface.
- This will convert the selection into **vtkPolyData** format and then we will be able to generate the surface normals.

Color using `p` (pressure)



Computing the forces on the cylinder

- Select the object `ExtractSurface1`, and from the menu `Filters → Alphabetical` select the `Generate Surface Normals` filter.
- This will generate the surface normals.



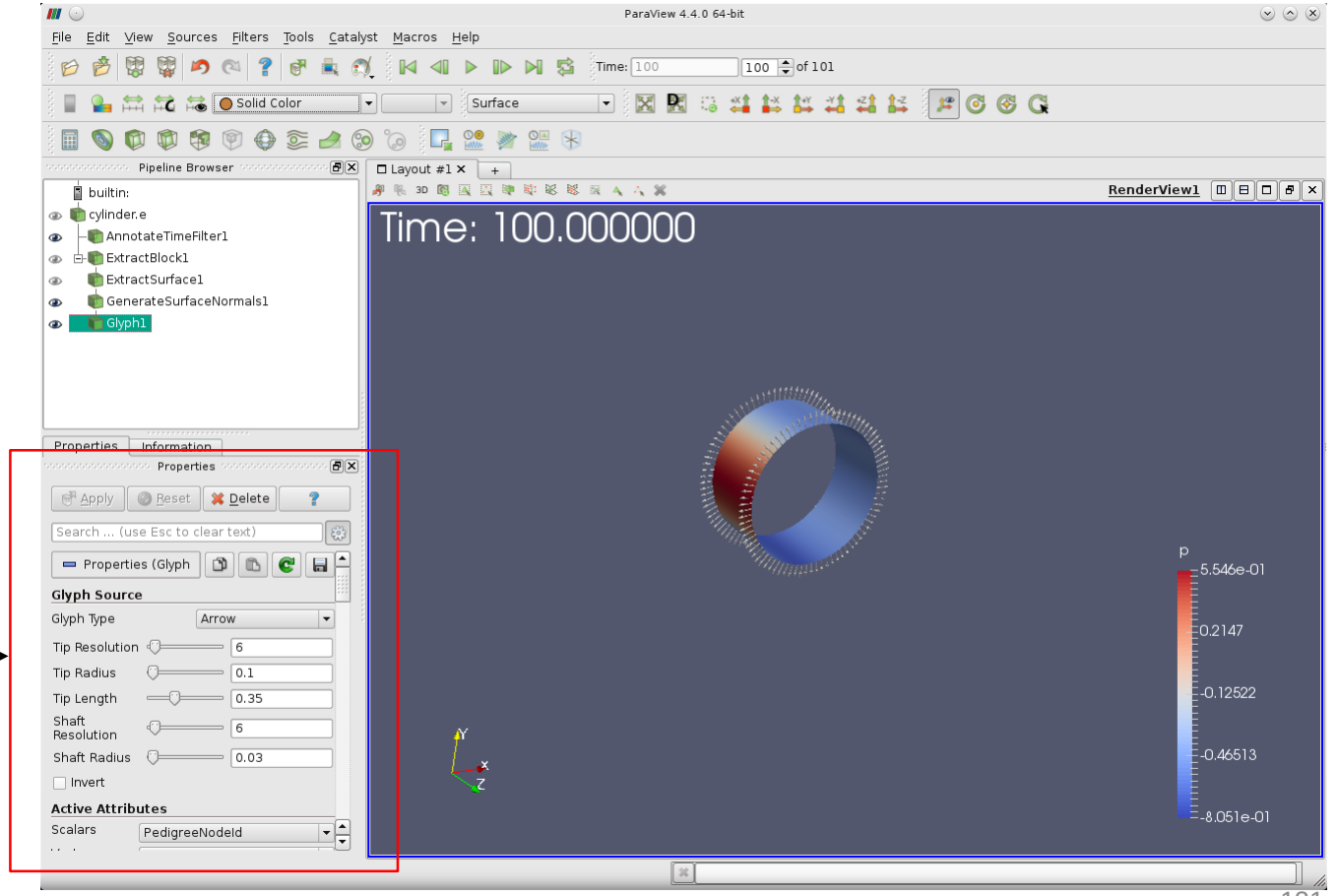
Check to flip normals

Scientific visualization with paraFoam/ParaView

Computing the forces on the cylinder

- To plot the surface normals, select the object `GenerateSurfaceNormals1`, and from the menu `Filters → Alphabetical` or the toolbar, select the `Glyph` filter 
- The normals are pointing outwards because in the previous step we fliped the normals.

Adjust the glyphs options to obtain a similar result



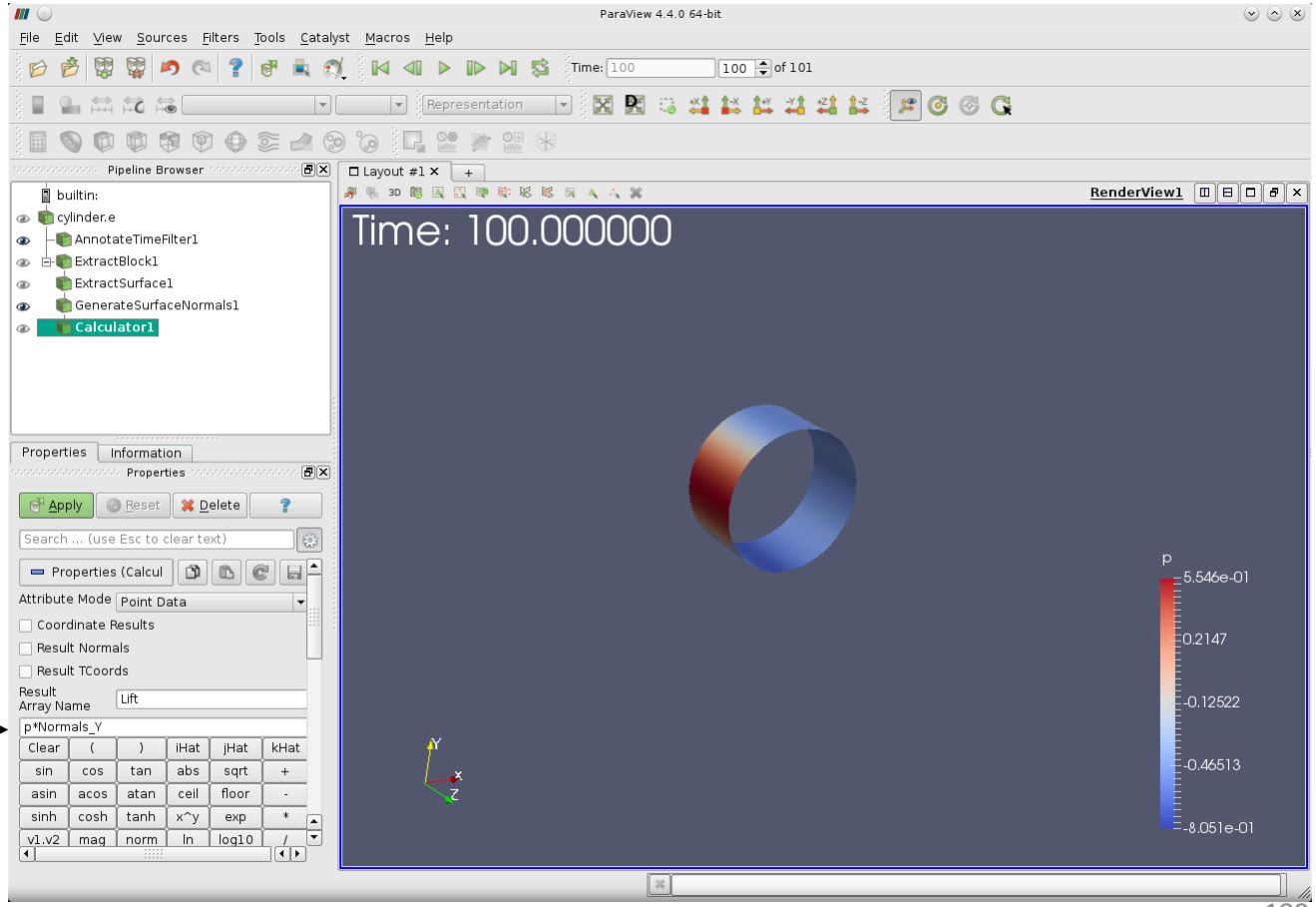
Scientific visualization with paraFoam/ParaView

Computing the forces on the cylinder

- Select the object `GenerateSurfaceNormals1`, and from the menu `Filters → Alphabetical` or the toolbar, select the `Calculator` filter 
- Compute the vertical force (lift) on the cylinder surface using the pressure and normals.

Remember, as we are using an incompressible solver, the pressure coming from OpenFOAM® is the modified pressure, so you need to multiply it by the reference density value, which is one in this case.

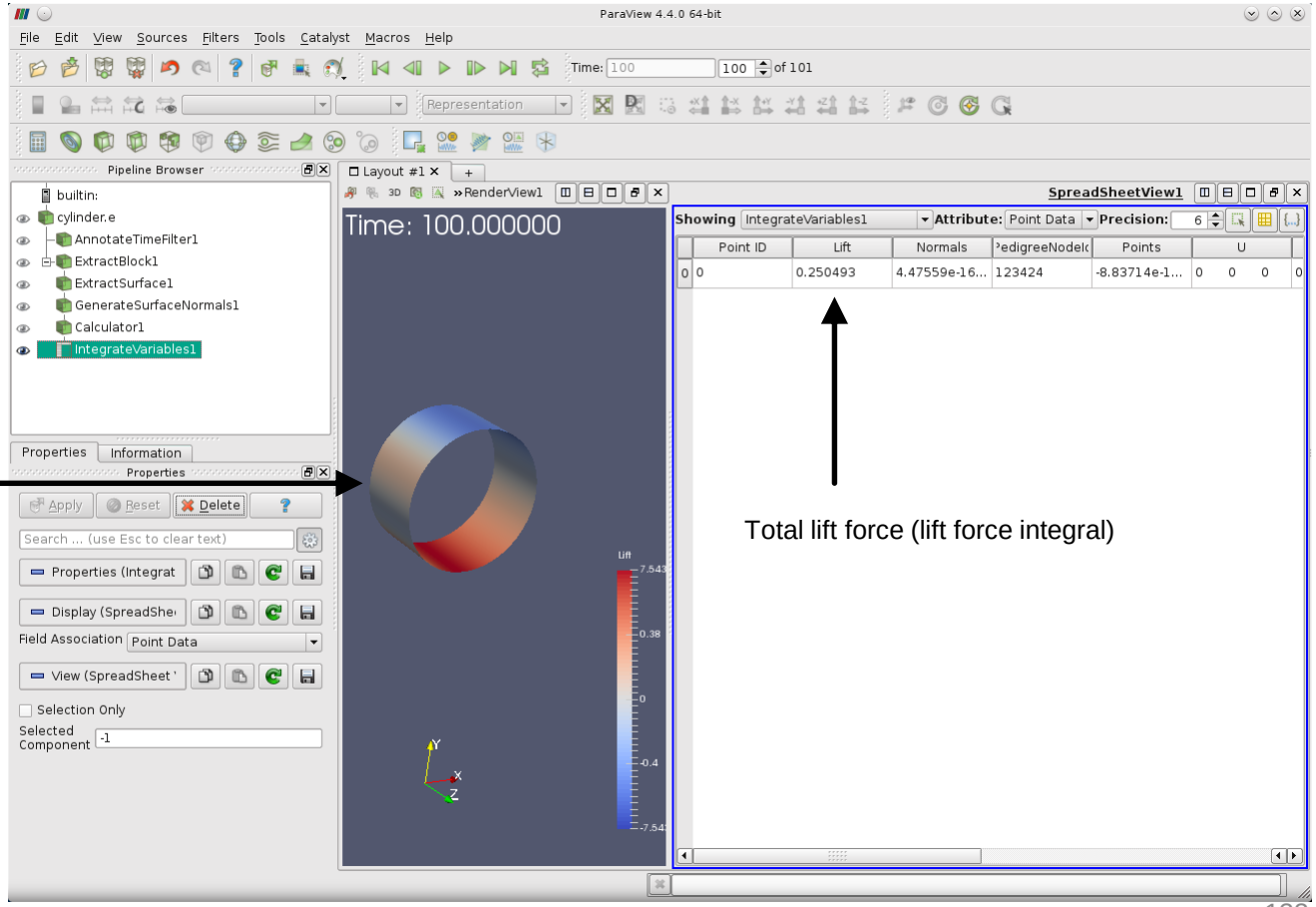
Lift = p * Normal_Y →



Computing the forces on the cylinder

- Select the object `Calculator1`, and from the menu `Filters → Alphabetical`, select the `Integrate Variables` filter.
- This will integrate the lift force over the cylinder.

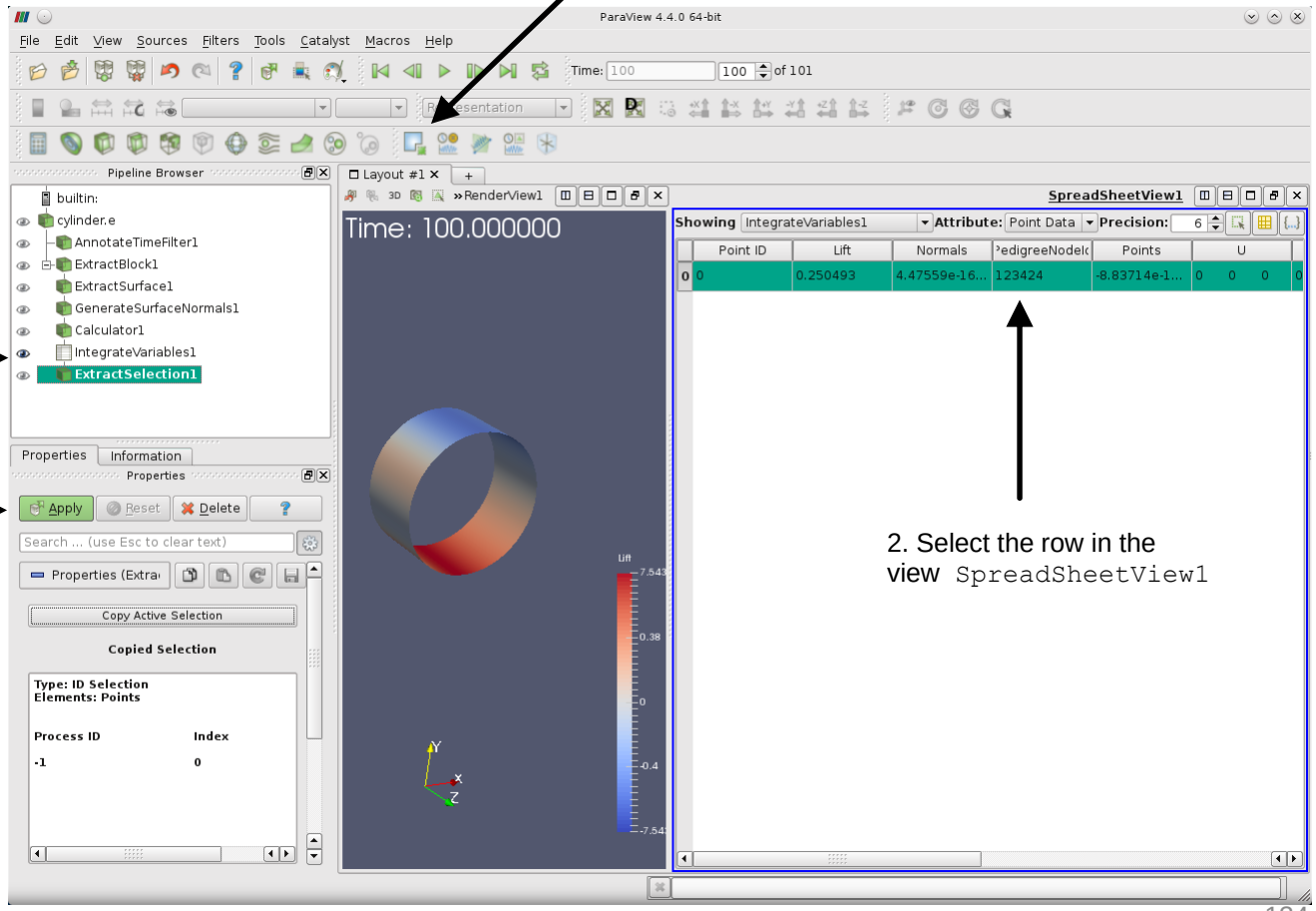
Cylinder colored by lift force distribution



Scientific visualization with paraFoam/ParaView

Computing the forces on the cylinder

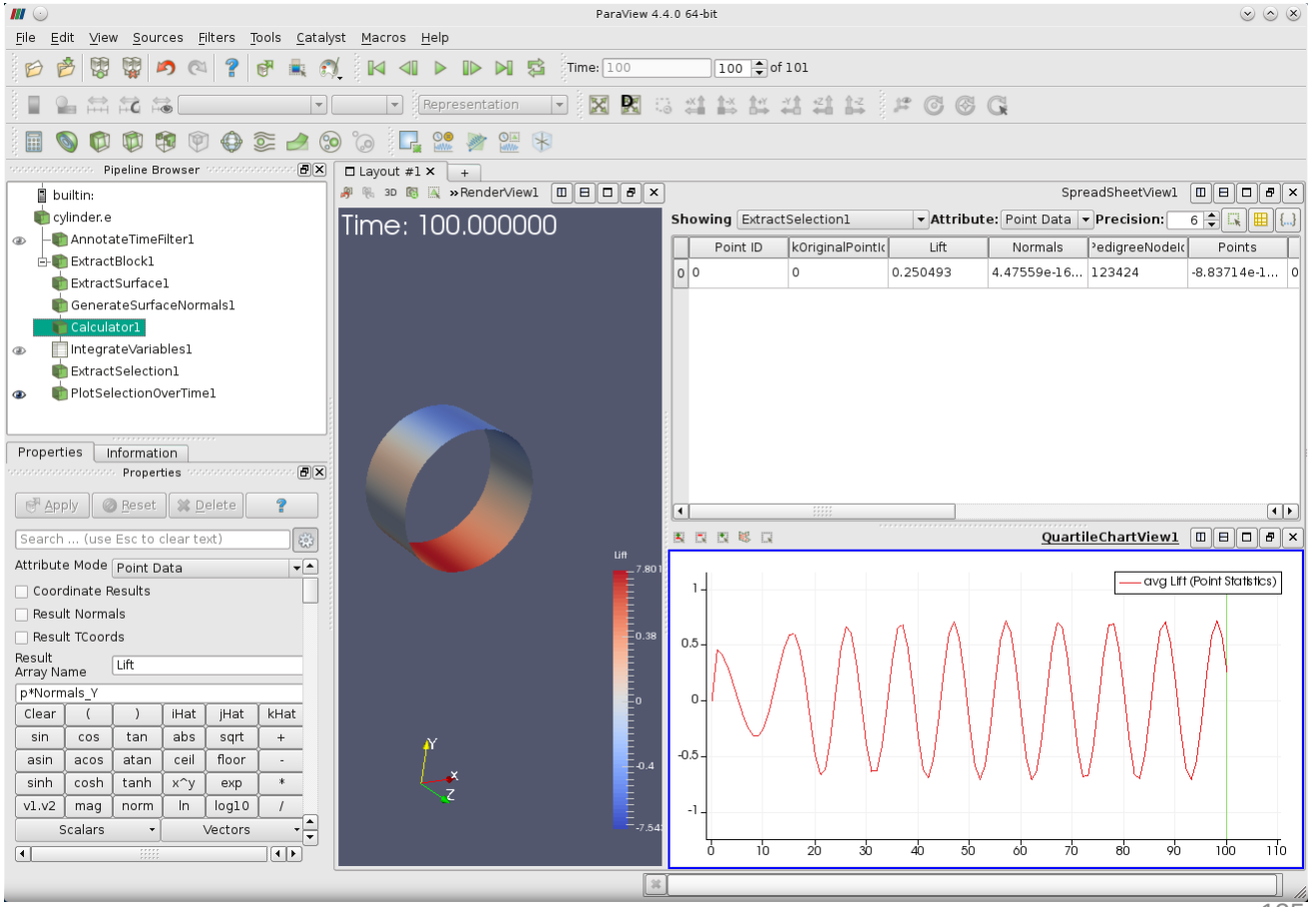
- Select the object `IntegrateVariables1`, and from the menu `Filters → Alphabetical` or the toolbar, select the `Extract Selection` filter 



Scientific visualization with paraFoam/ParaView

Computing the forces on the cylinder

- Select the object `ExtractSelection1`, and from the menu `Filters → Alphabetical` or the toolbar, select the `Plot Selection Over Time` filter 



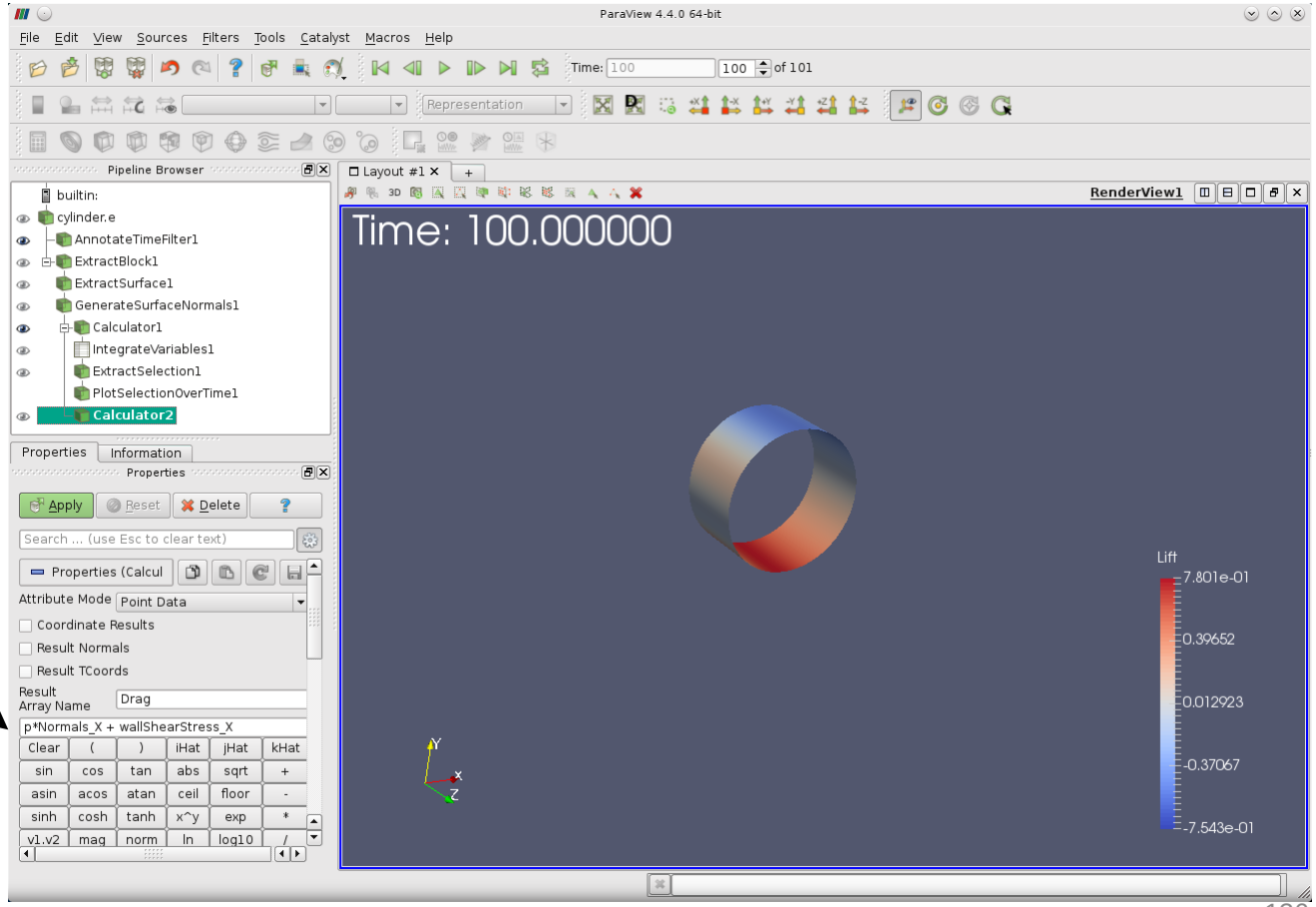
Scientific visualization with paraFoam/ParaView

Computing the forces on the cylinder

- Compute the horizontal force (drag) on the cylinder surface using the pressure and normals.
- Proceed in the same way as for lift.
- In this case we also add the wall shear stress in the X direction.

Remember, as we are using an incompressible solver, the pressure coming from OpenFOAM® is the modified pressure, so you need to multiply it by the reference density value, which is one in this case.

Drag = p * Normal_X + wallShearSress_X

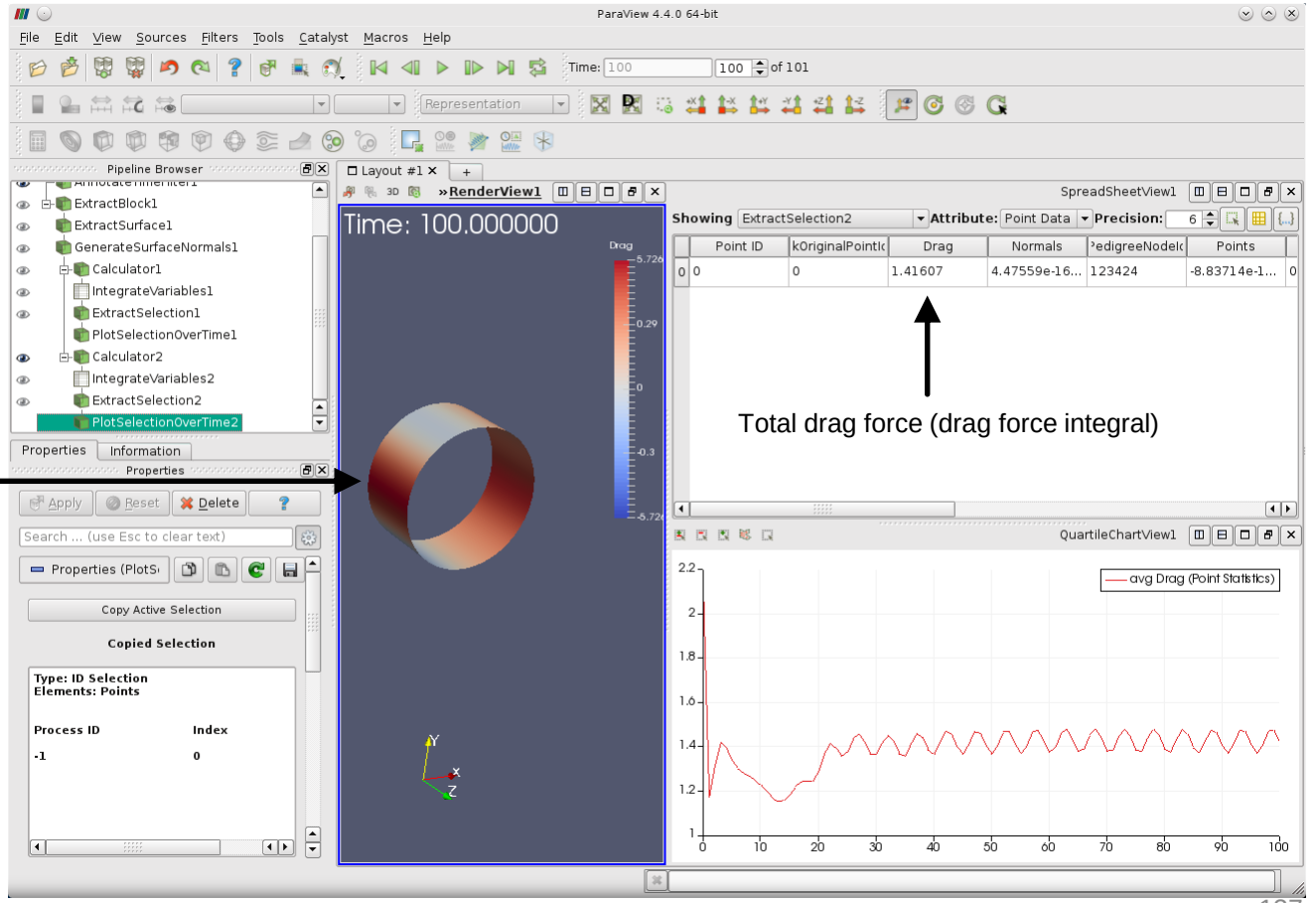


Scientific visualization with paraFoam/ParaView

Computing the forces on the cylinder

- Plot the computed drag force.
- In this case, you need to change the sign of the drag force in order to obtain the same result.

Cylinder colored by drag force distribution



Scientific visualization with paraFoam/ParaView

Animation of pressure field and drag force on the cylinder