

ParaView Workshop

Jonas
Lindemann,
LUNARC

Based in part on slides
and data sets by
Mike Bailey Oregon State
University

InfraVis ParaView workshop

- This is a participatory workshop
- You should have received links to the material and download links for ParaView
- I will present ParaView in a way that you can follow along
- If you want to actively participate please have ParaView running and have the examples in the Mike_Bailey folder available.
- It is perfectly ok to follow along passively

InfraVis ParaView workshop

- ParaView is complex application
- We will not have time to cover all functionality
- I will try to convey the workflow of the application
- A starting point for further work

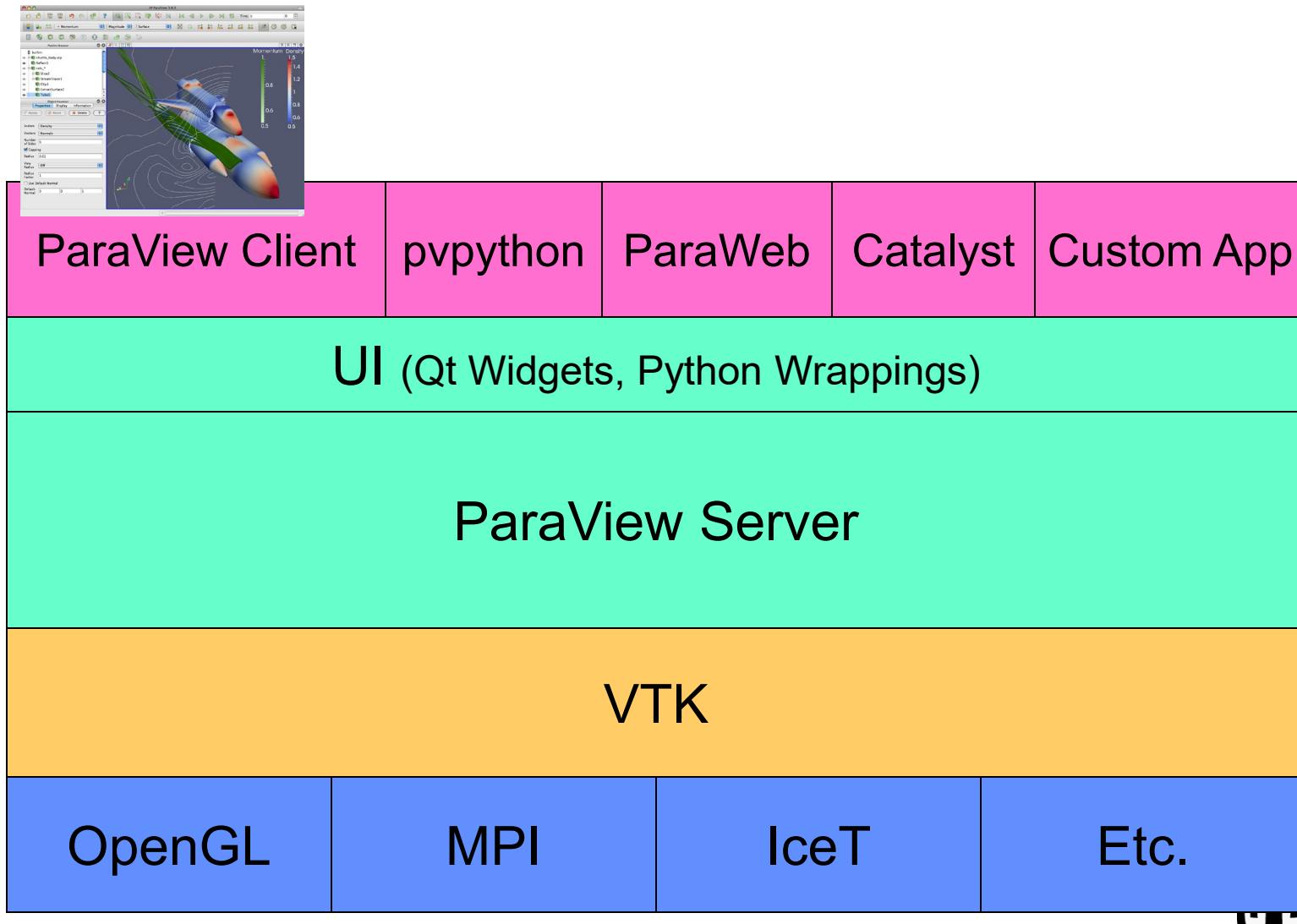
Downloads and data files

- Data files are available here:
 - <https://github.com/jonaslindemann/paraview-workshop>
- ParaView can be downloaded here
 - <https://www.paraview.org/download/>
- Download the stable version, currently 5.10

What is ParaView?

- Tool for visualising data
- Tool for analysing and data mining
- A scripting based visualisation tool
- Parallel visualisation application for large scale visualisation
- In-situ visualisation tool
- "Swiss army knife of visualisation"

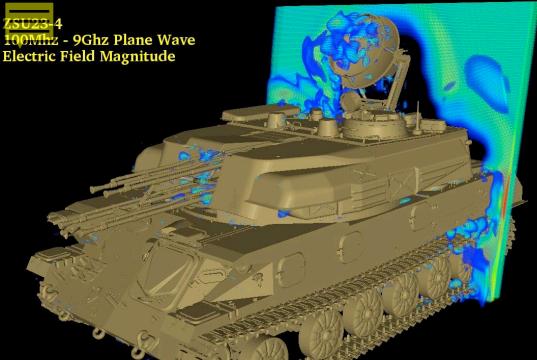
Paraview Application Architecture



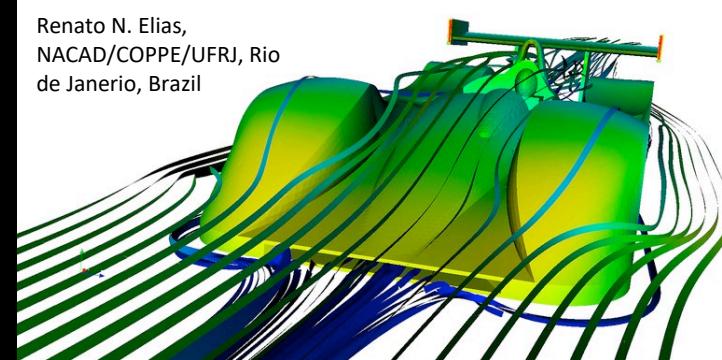
ParaView Development

- Started in 2000 as collaborative effort between Los Alamos National Laboratories and Kitware Inc. Sandia has been a major contributor since 2005.
 - ParaView 0.6 released October 2002.
- Paraview 3.0 release in May 2007.
 - GUI rewritten to be more user friendly and powerful.
- ParaView 4.0 released in June 2013.
 - Properties panel redesign for smoother interaction.
- ParaView 5.0 released in January 2016.
 - Updated to OpenGL 3.2 features. Huge performance improvements.

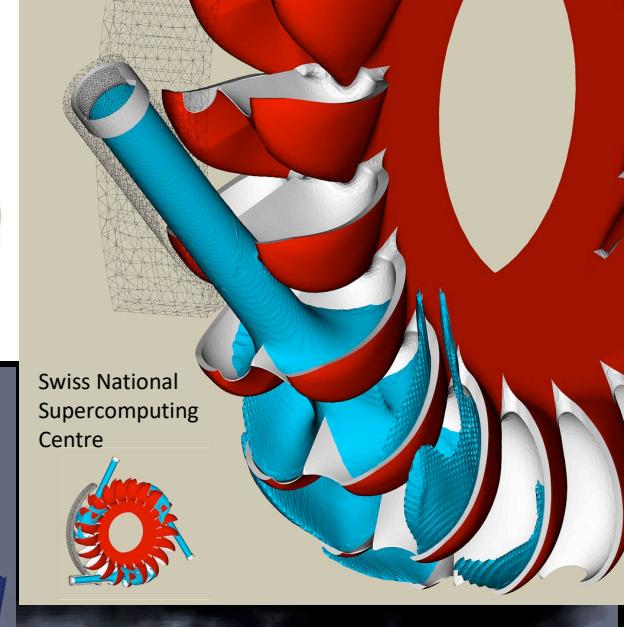
ZS123-4
100MHz - 9Ghz Plane Wave
Electric Field Magnitude



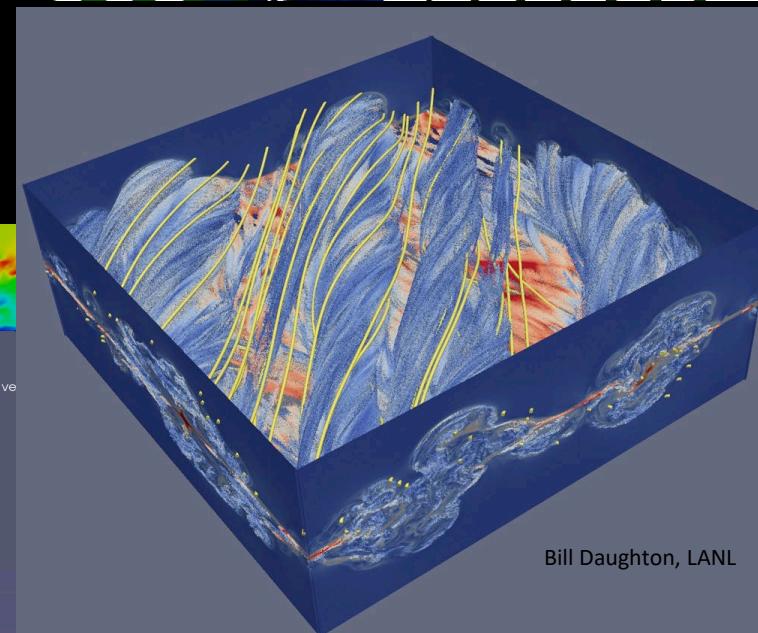
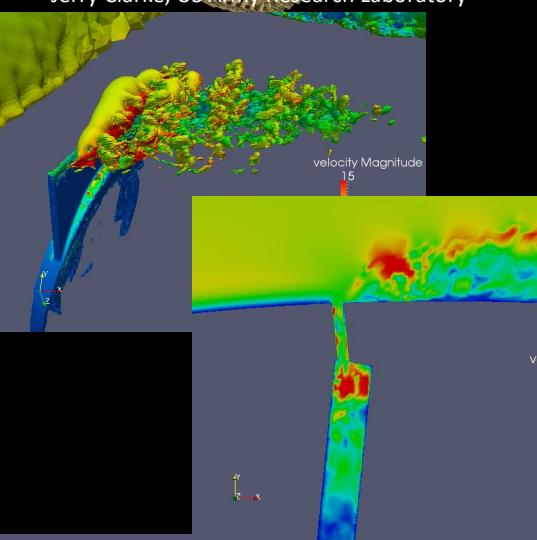
Jerry Clarke, US Army Research Laboratory



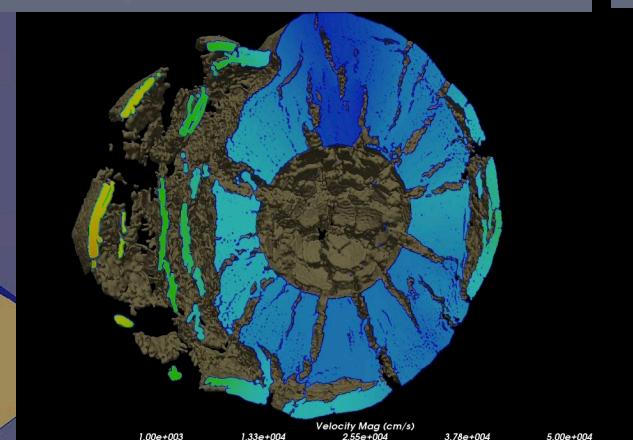
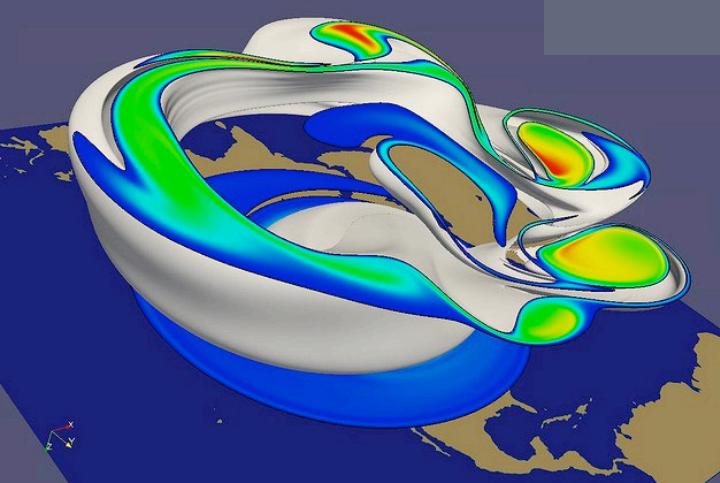
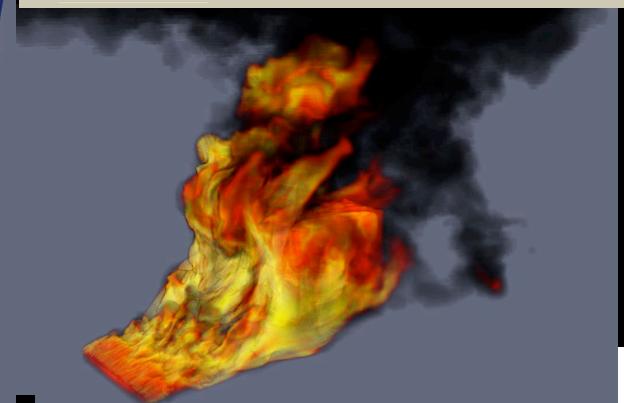
Renato N. Elias,
NACAD/COPPE/UFRJ, Rio
de Janeiro, Brazil



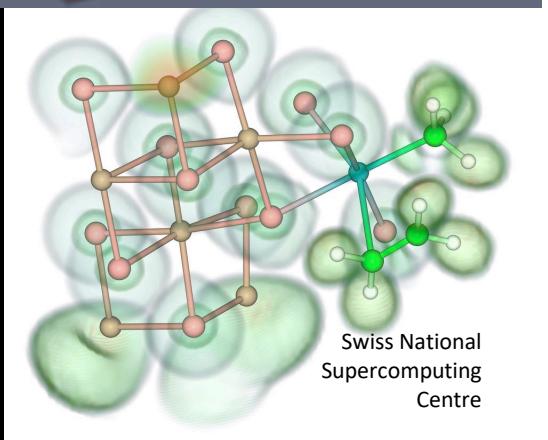
Swiss National
Supercomputing
Centre



Bill Daughton, LANL



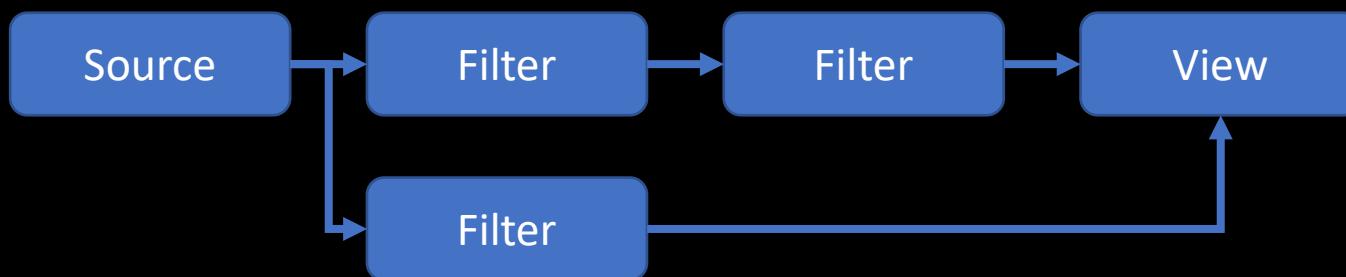
1.00e+003 1.33e+004 2.55e+004 3.78e+004 5.00e+004



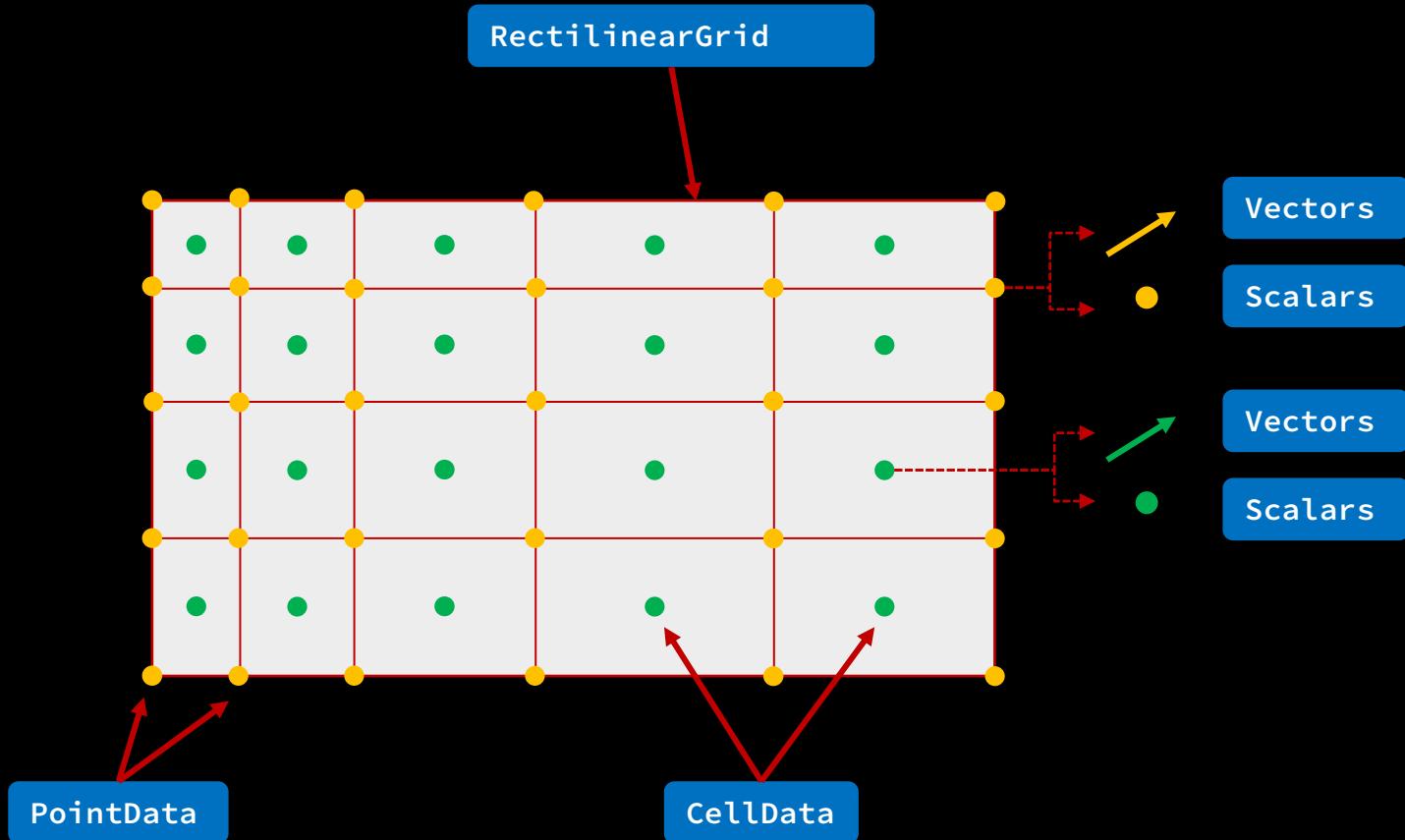
Swiss National
Supercomputing
Centre

Basic concept

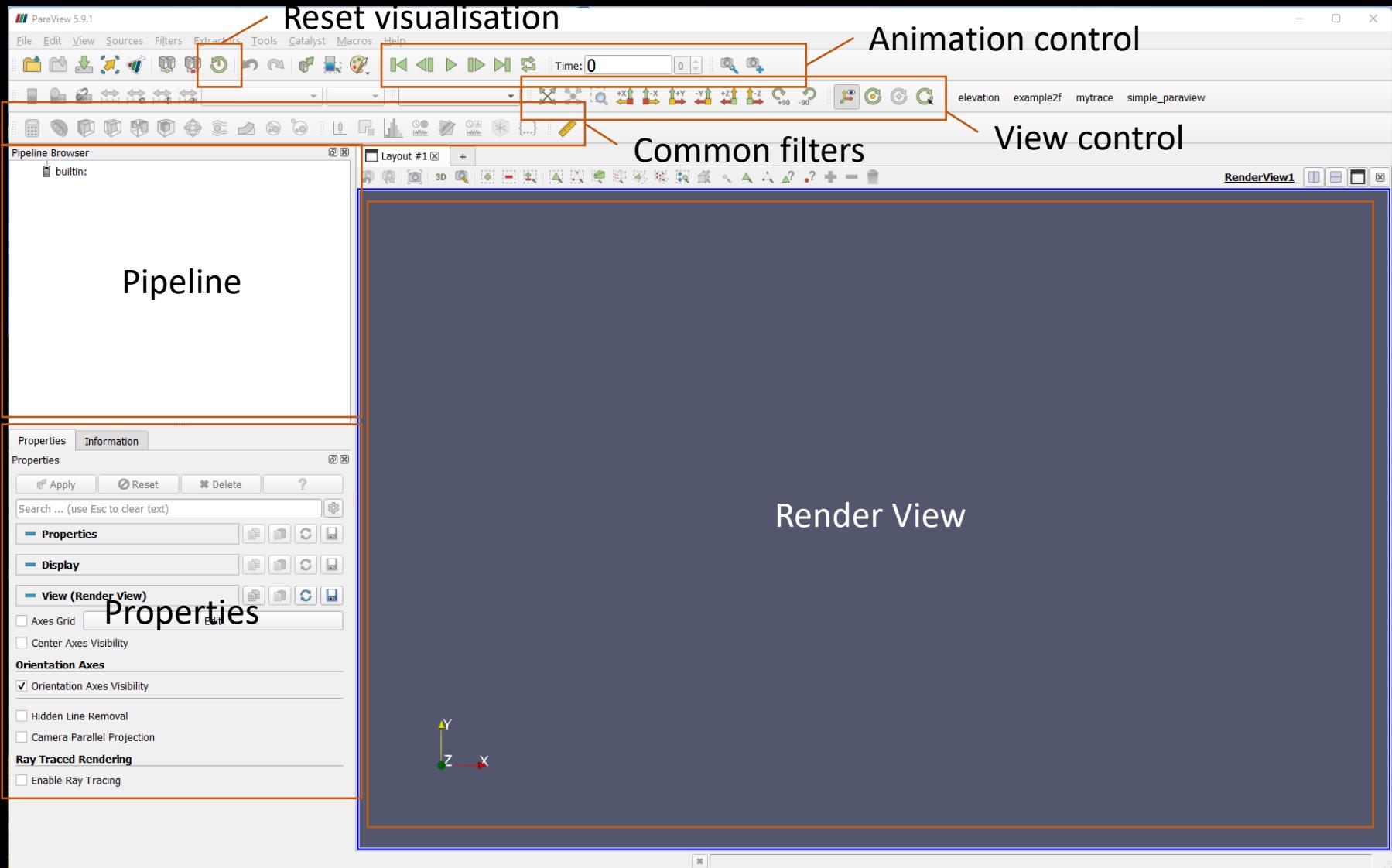
- Visualisation application developed by KitWare
- Uses the Visualisation ToolKit (VTK) as foundation
- Uses a flow concept for generating visualisations



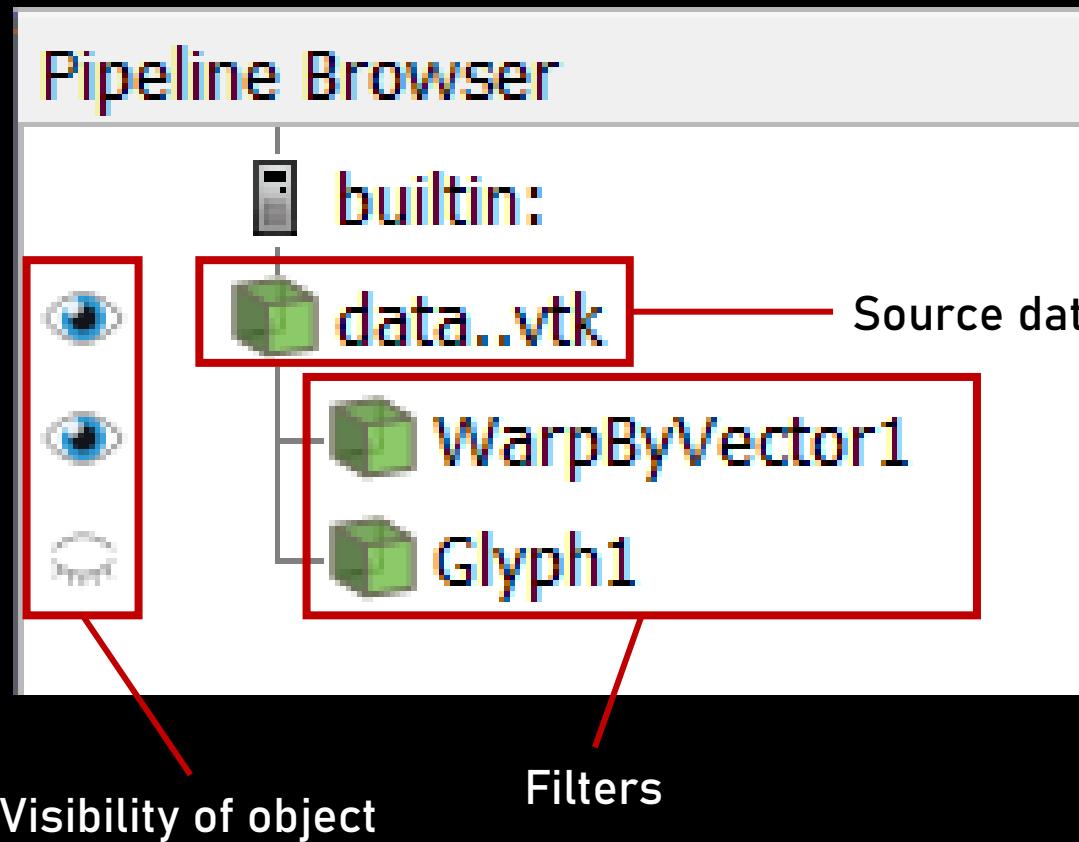
Data structures and data - High level



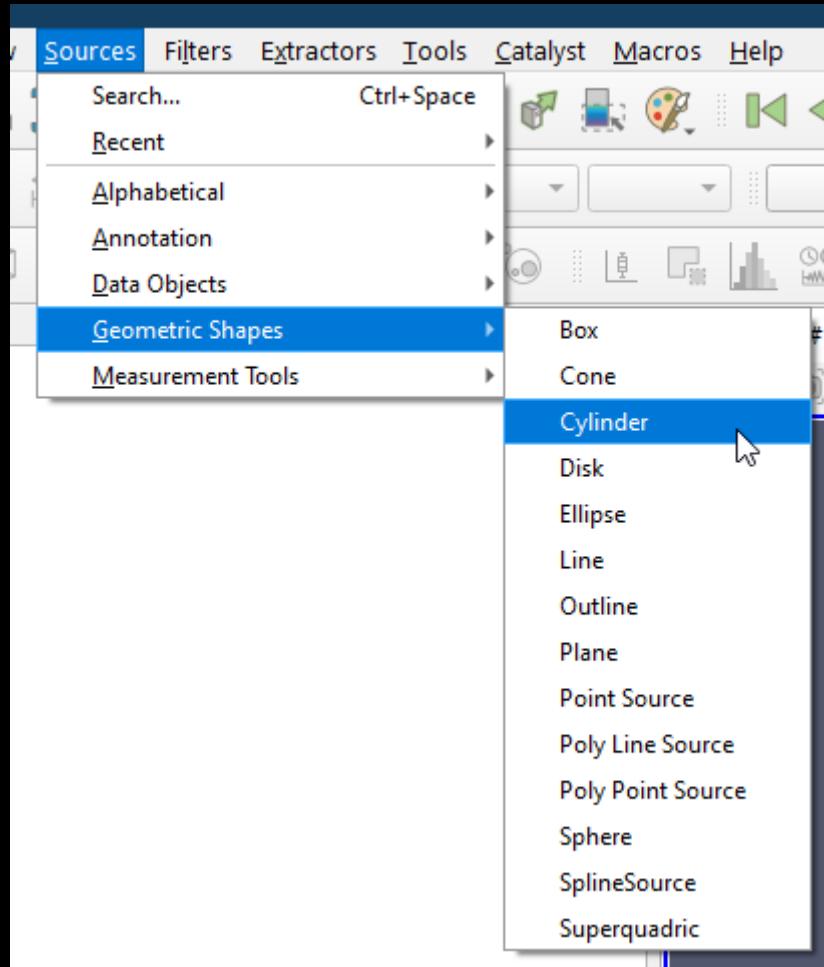
ParaView user interface



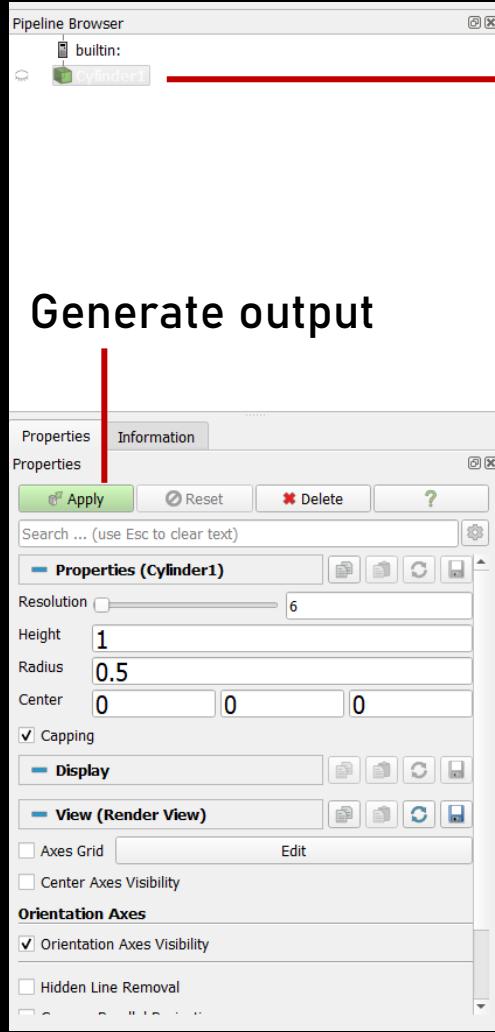
Flow in ParaView



Basic usage - Creating a source

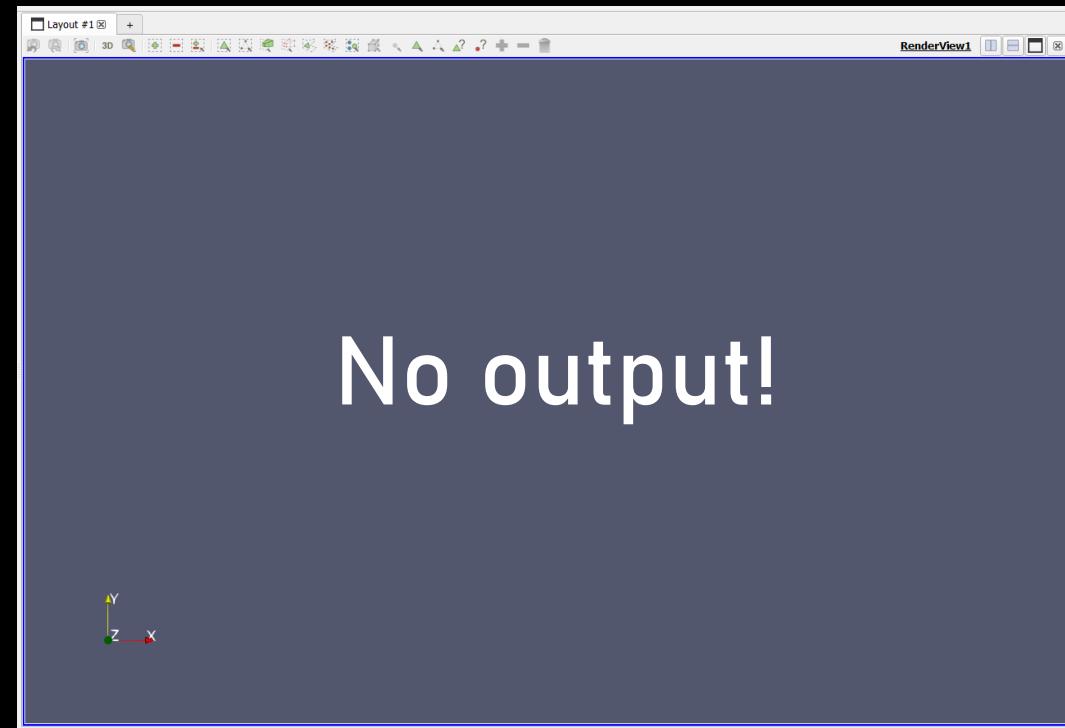


Creating a Cylinder



Generate output

Cylinder source

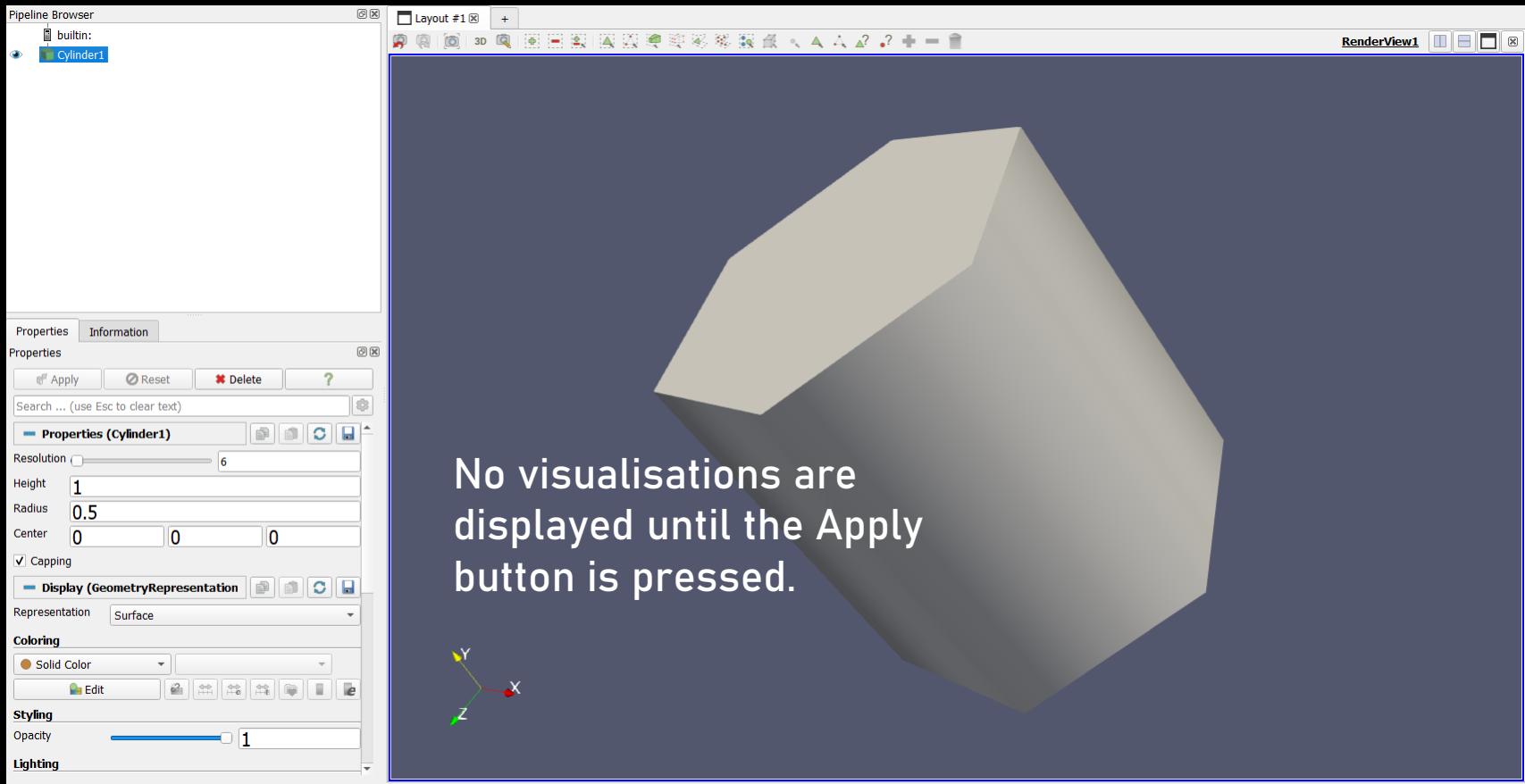


No output!

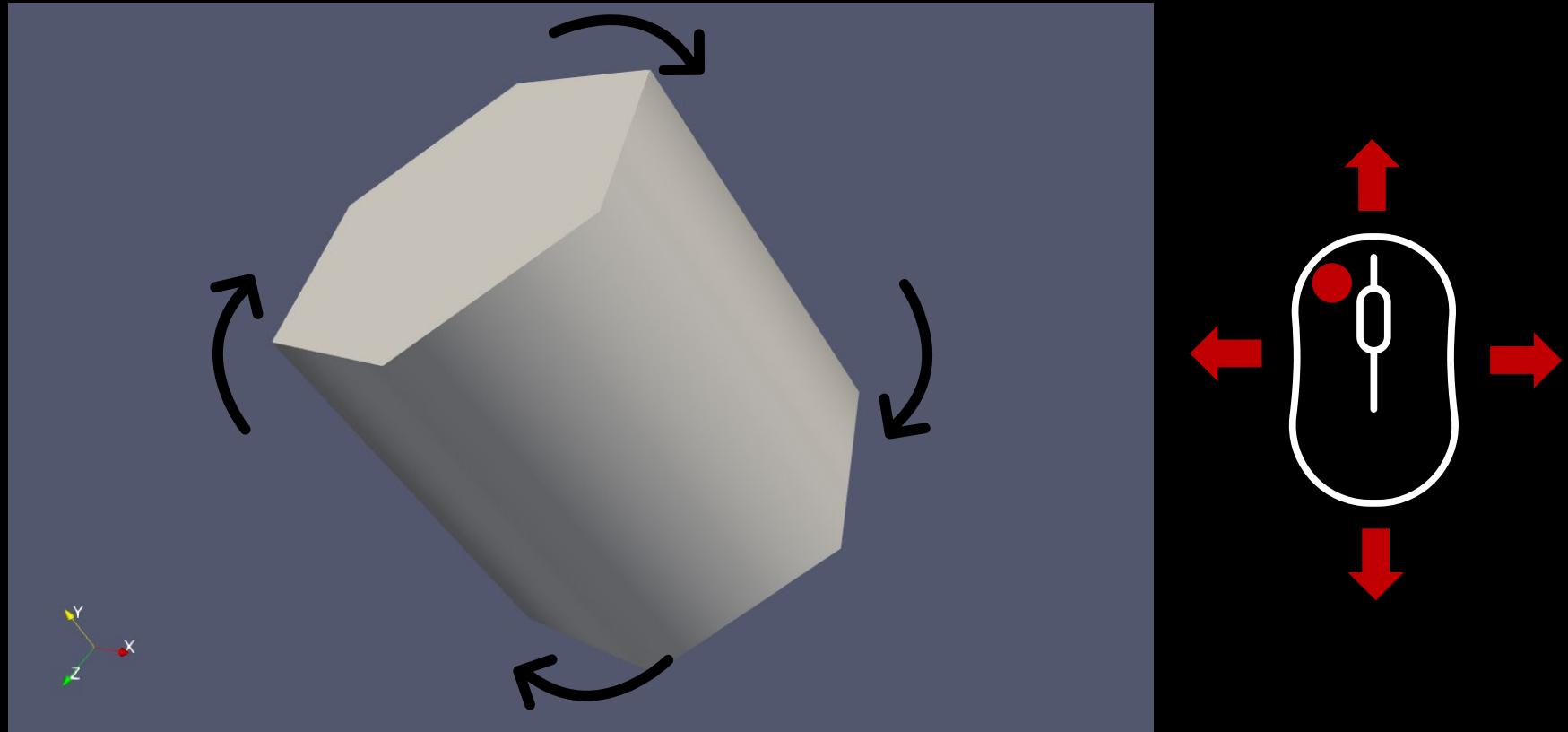
Creating a Cylinder



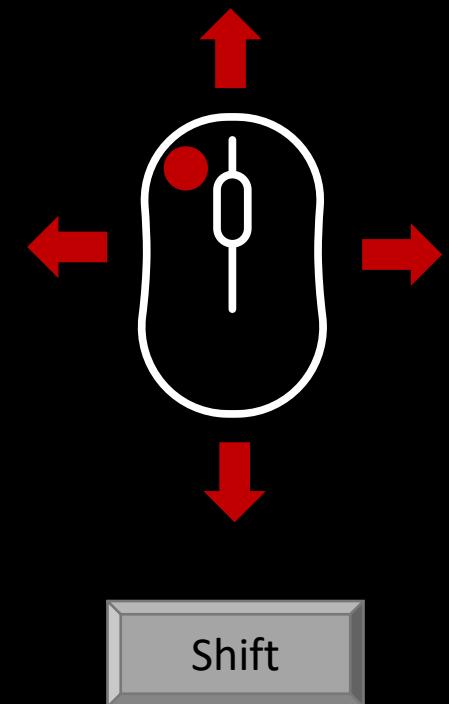
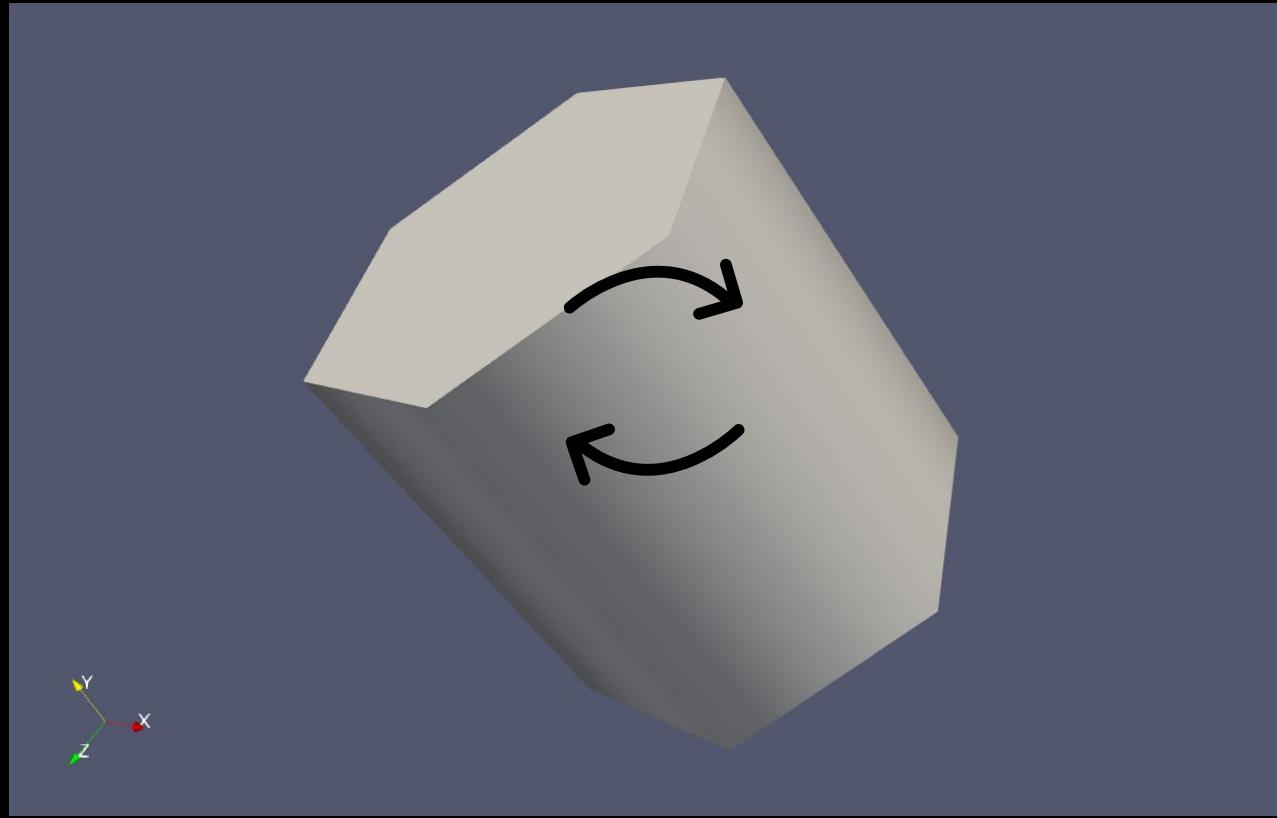
Enables auto-apply. Can be costly in some operations.



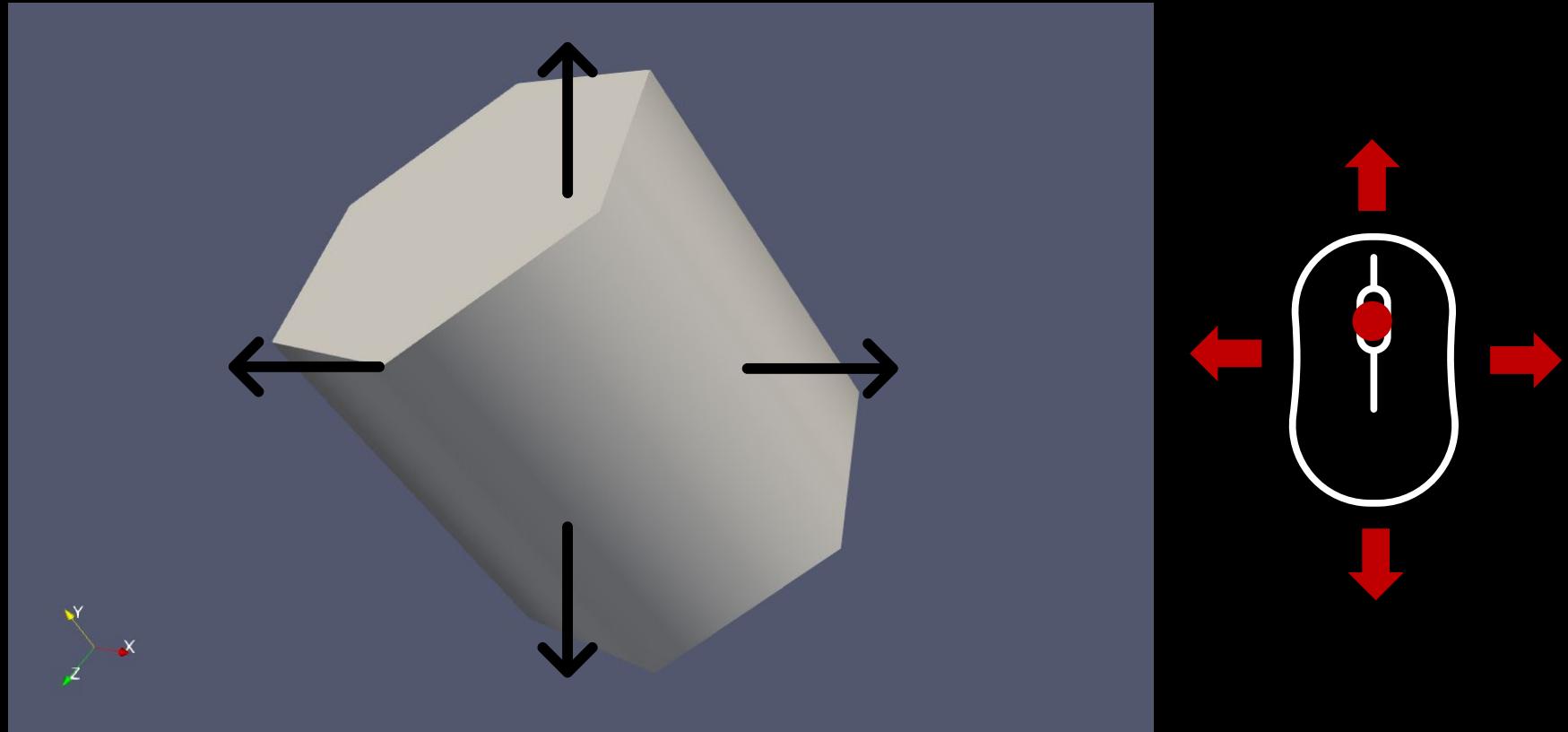
3D view manipulation



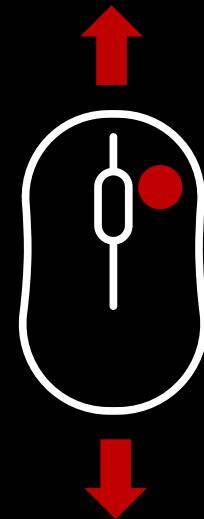
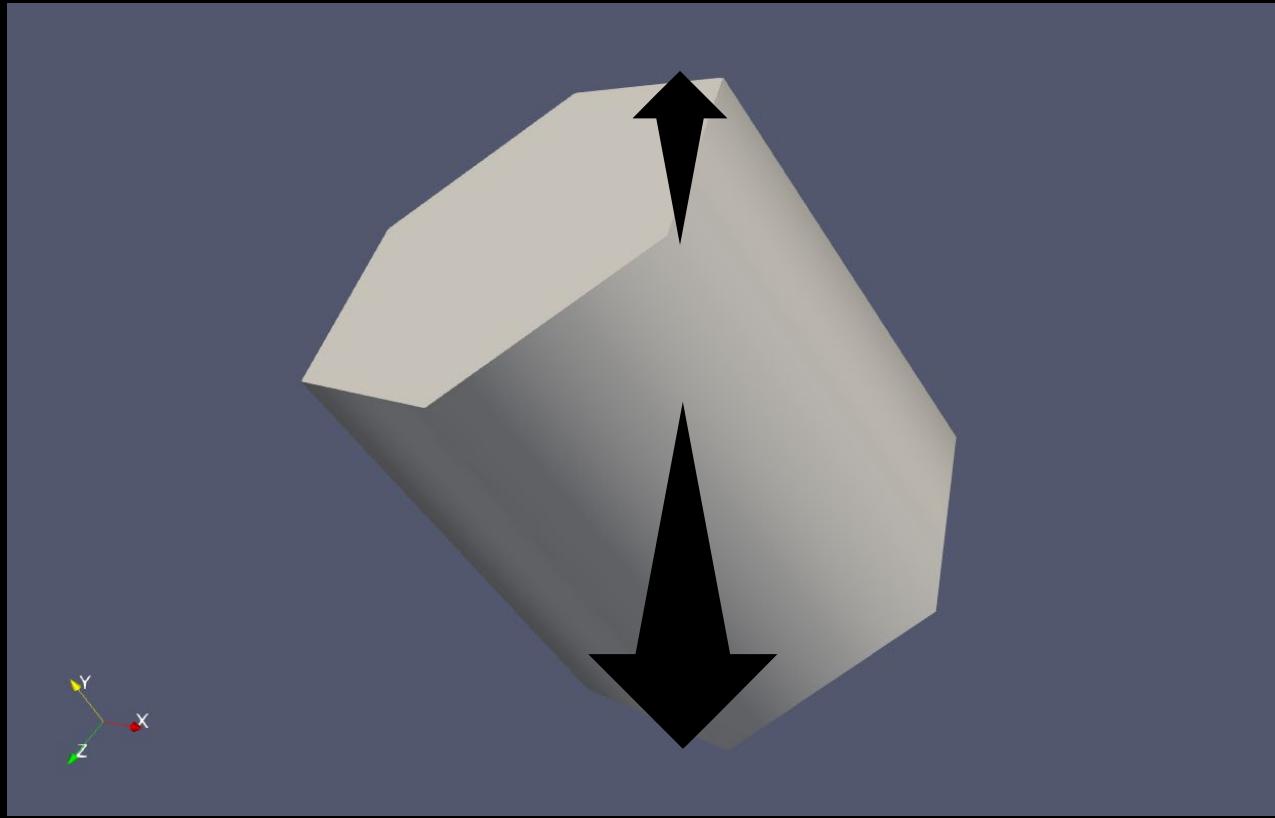
3D view manipulation



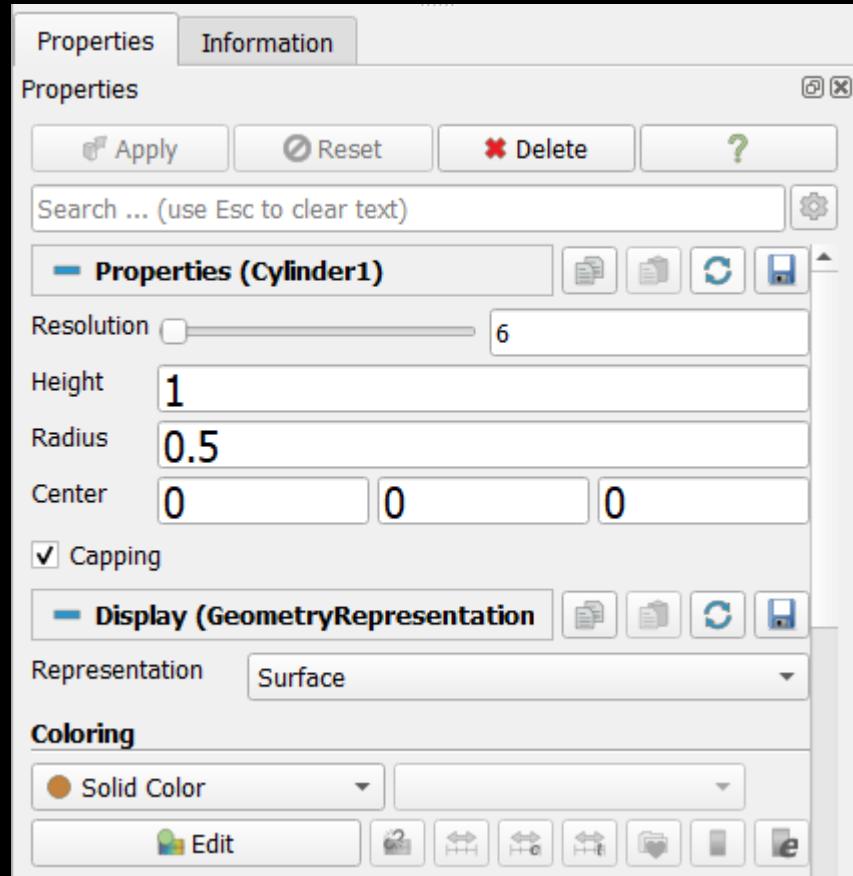
3D view manipulation



3D view manipulation



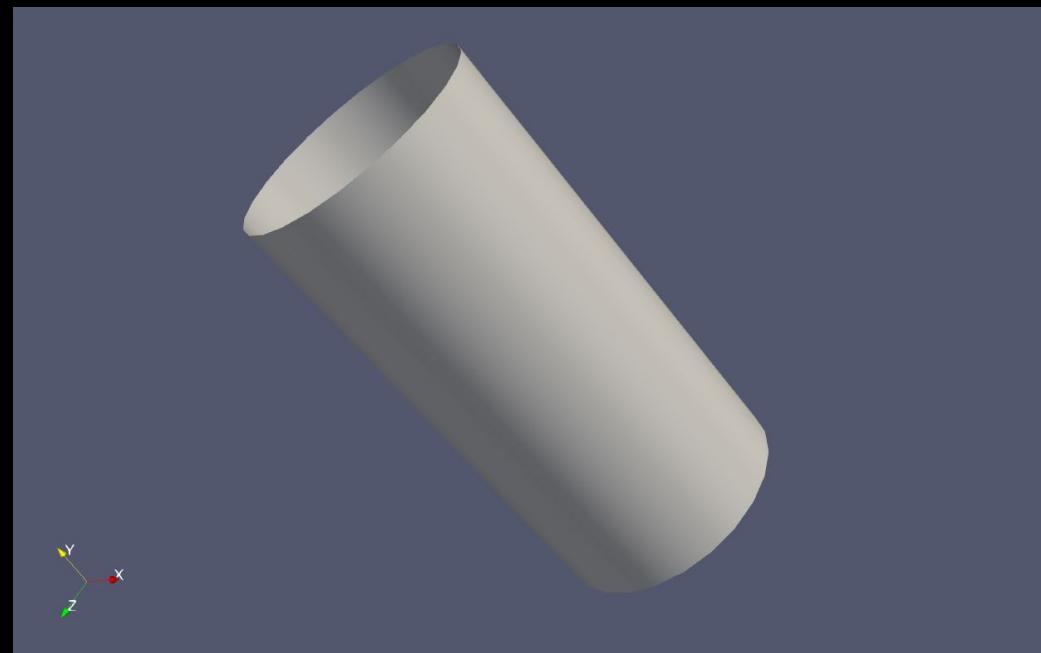
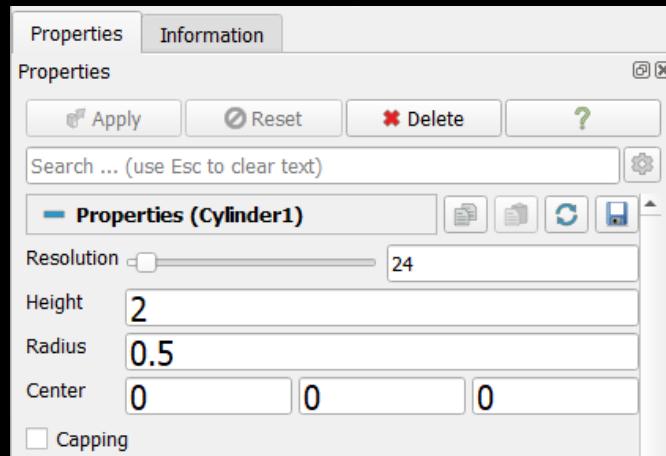
Specific object properties



All objects in ParaView have properties that can be changed in the property inspector.

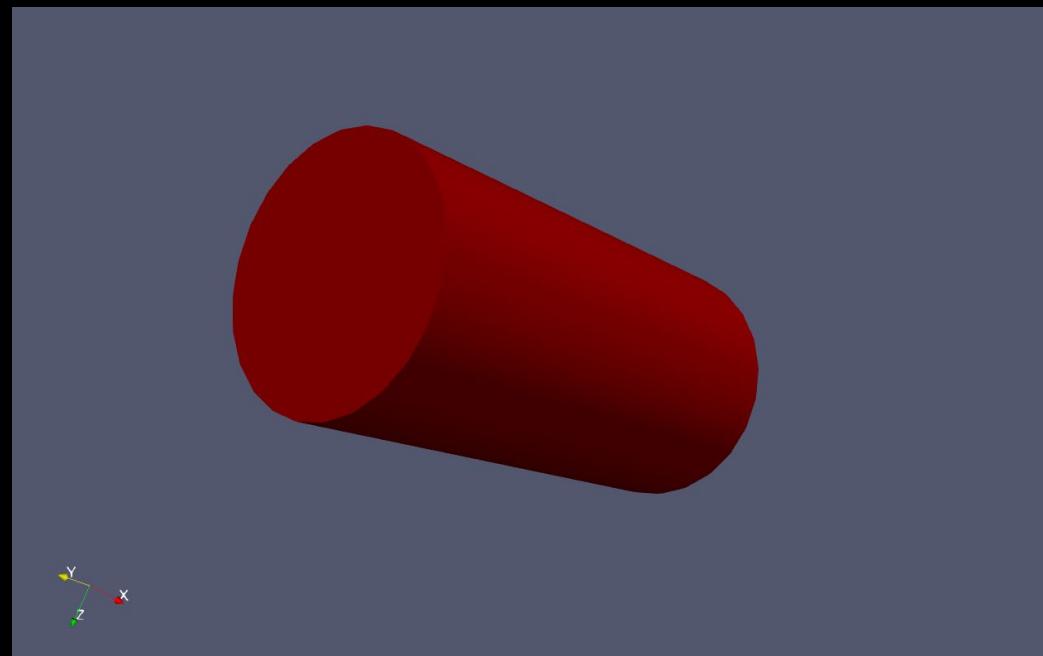
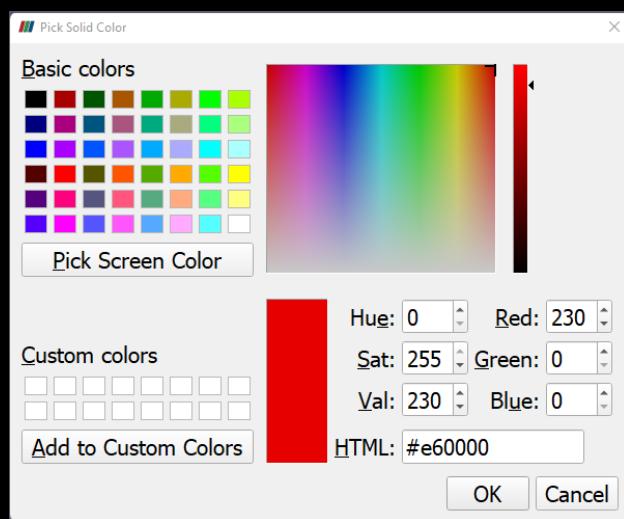
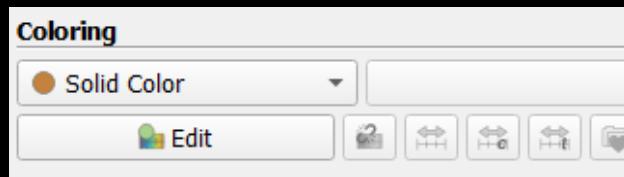
For the Cylinder this is resolution, height, radius, capping.

Specific object properties



Don't forget to press apply
after a change or use the
auto-apply button

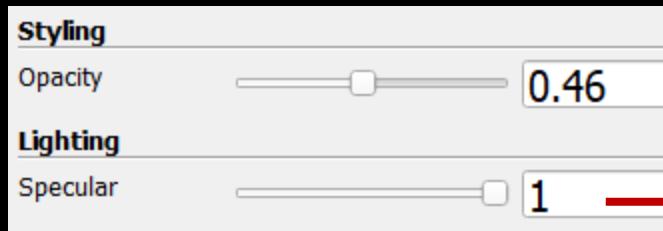
Color properties



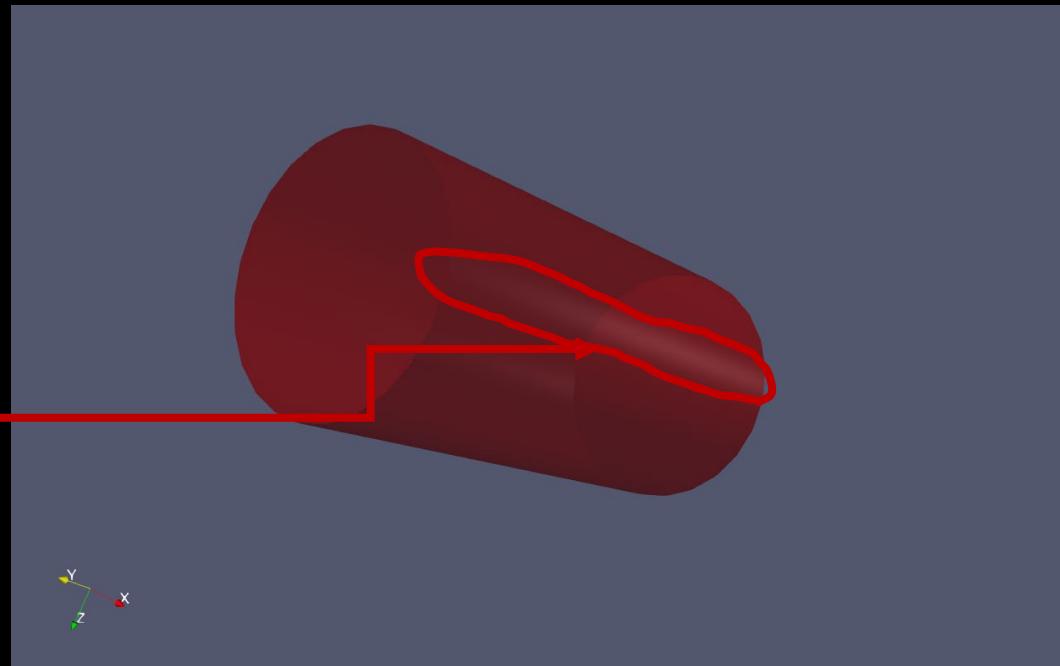
All objects can be given
colors either solid or by
values / vectors

Material properties

Objects can also be given opacity and specular properties.



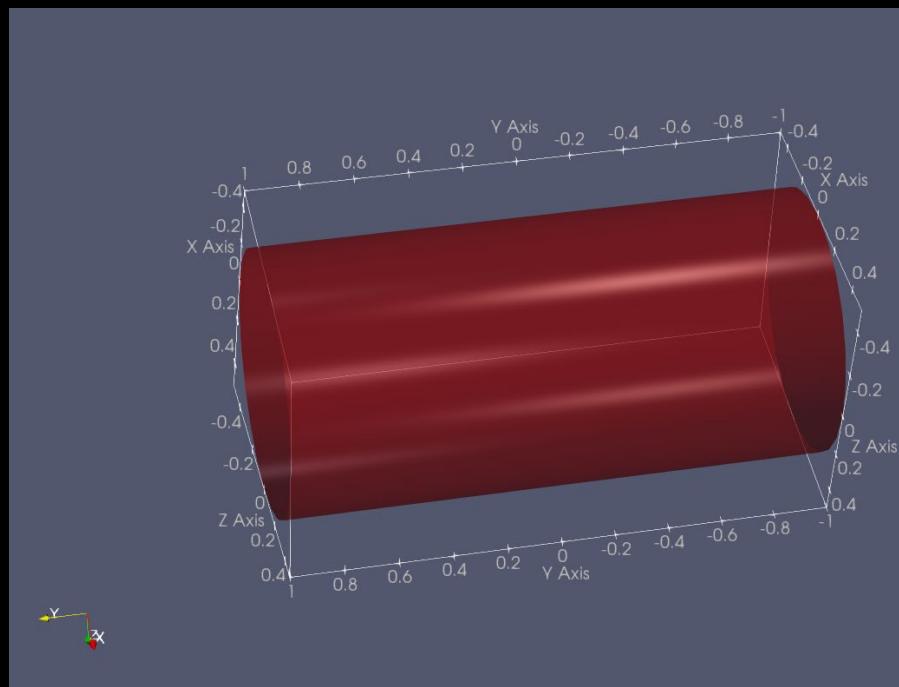
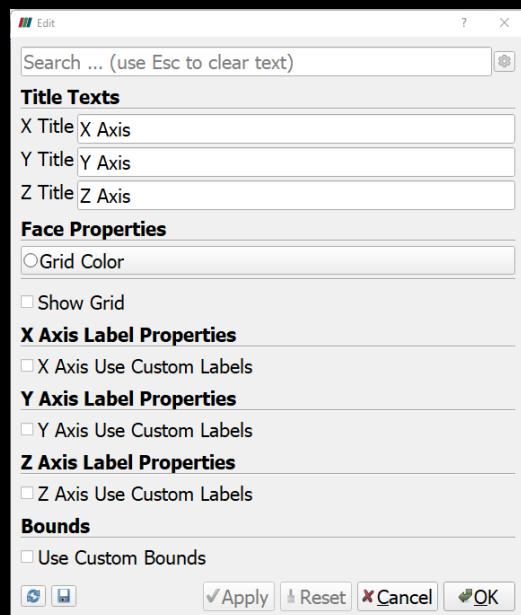
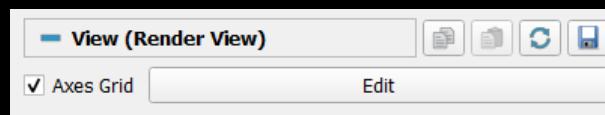
Specular controls how shiny an object is.



Opacity controls the transparency of an object. Can be useful to show outlines of objects.

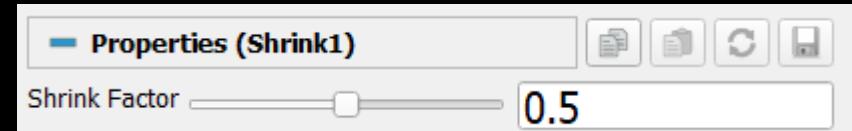
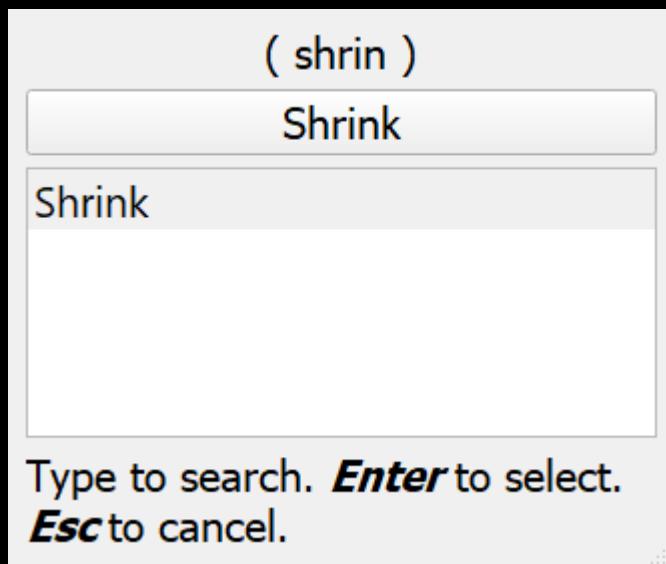
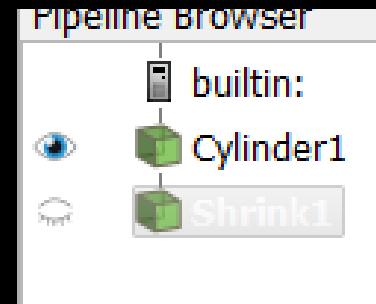
Axes

Axes can be used to give objects a sense of scale

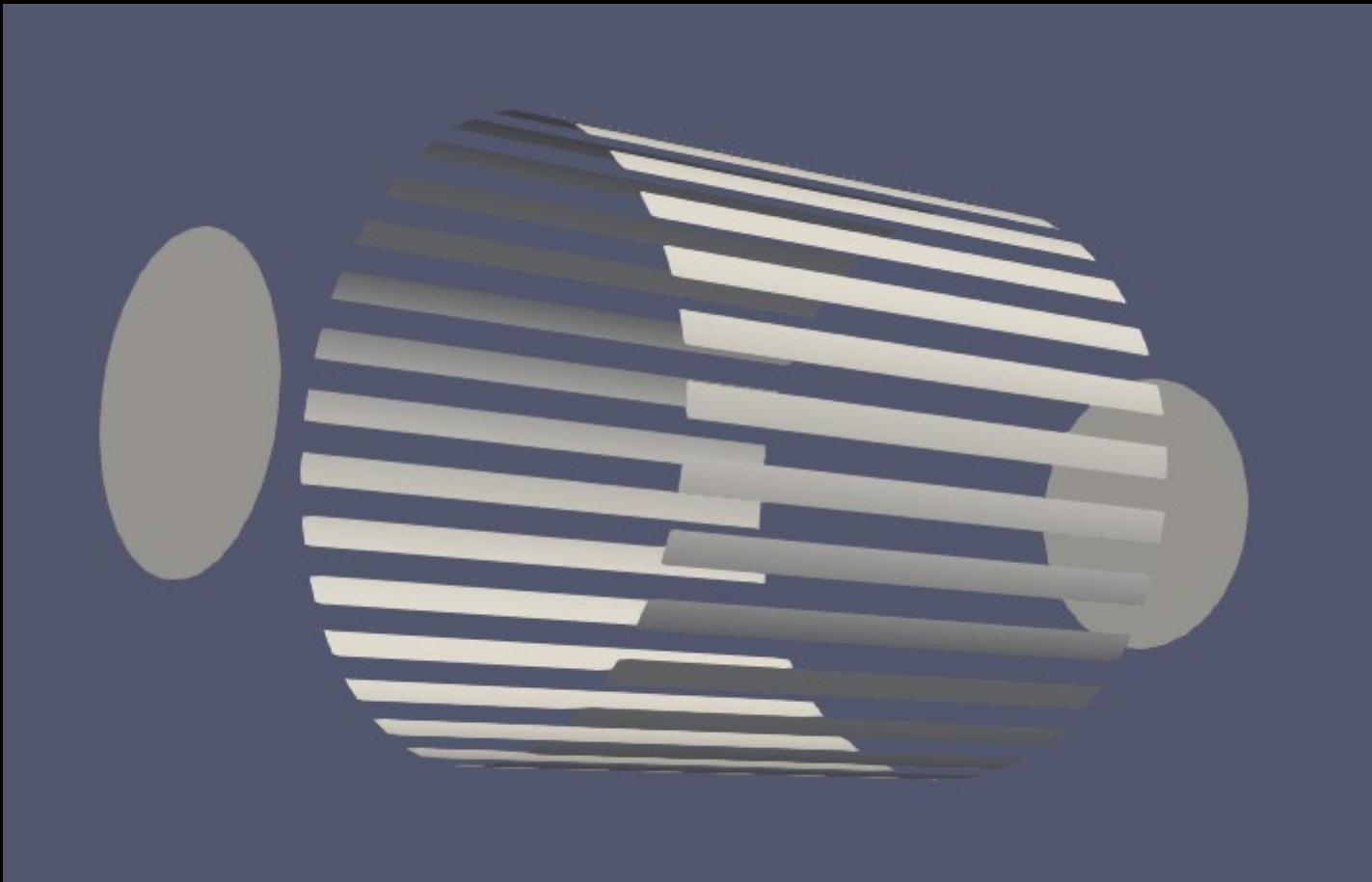


Applying a shrink filter

- Click Filters / Search...
- Type "Shrink" in the search box
- Press Enter or the button Shrink



Applying a shrink filter



Colors, opacity and specular properties can
be given for each separate object / filter

A close-up photograph of a car's tachometer. The needle is positioned exactly halfway between the number 5 and the number 6 on the scale. The numbers 5 and 6 are in red, while the other numbers are in a lighter gray. The word "RPM" is visible at the bottom of the scale, and "000" is partially visible to the left of the scale.

Demo time

Exercise

Create a yellow sphere with a radius of 3.0 and a theta and phi resolution of 32. Make the sphere really shiny!

Apply a shrink filter with a shrink factor of 0.8.

Color the shrunk sphere green. It should also be shiny.



Reset ParaView



A close-up, slightly blurred photograph of several open books stacked together. The pages are visible, showing various shades of brown and beige, suggesting old paper. The text on the pages is completely illegible due to the blur.

Reading scalar data

Open the scalar.csv file

- File/Open...
- Find the scalar.csv file.
- Click Apply to load the file

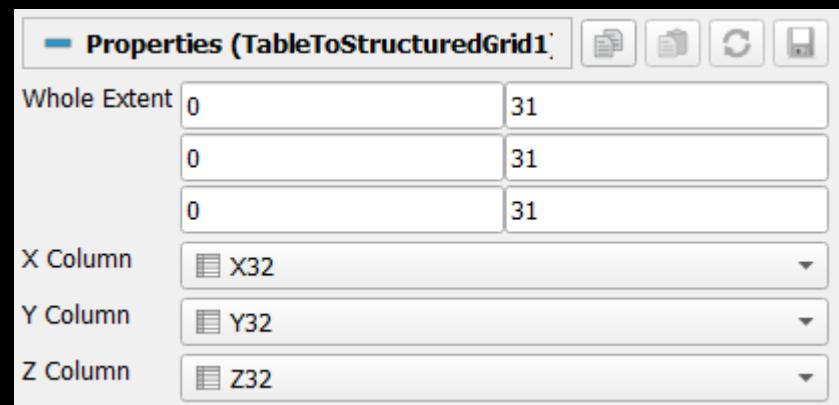
Reading CSV

Row ID	S	X32	Y32	Z32
0	0	-1	-1	-1
1	1	-0.94	-1	-1
2	2	-0.87	-1	-1
3	3	-0.81	-1	-1
4	4	-0.74	-1	-1
5	5	-0.68	-1	-1
6	6	-0.61	-1	-1
7	7	-0.55	-1	-1
8	8	-0.48	-1	-1

We now need to translate this data into something that can be visualised with ParaView.

To do this, we use the TableToStructureGrid filter.

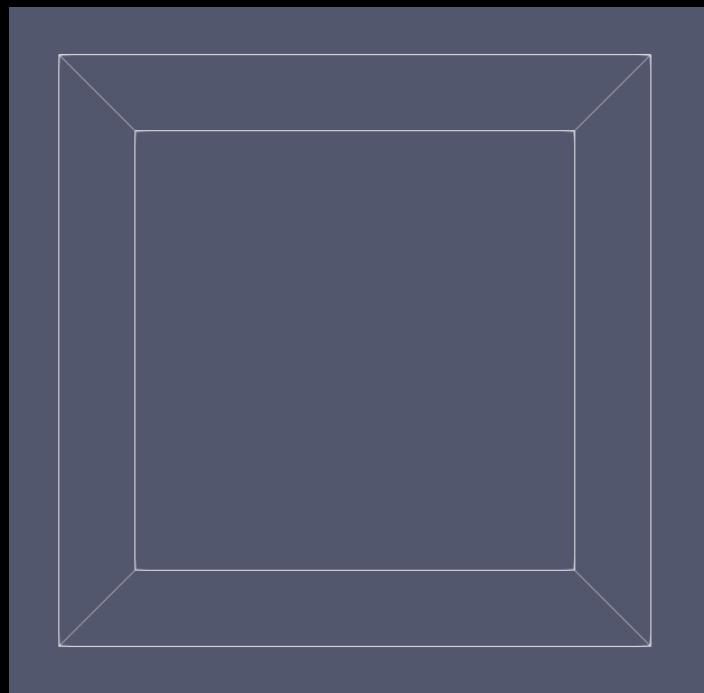
Before applying the filter we need to define the structure of the data and where it can find the X, Y, Z coordinates in the files.



Visualising Scalar data

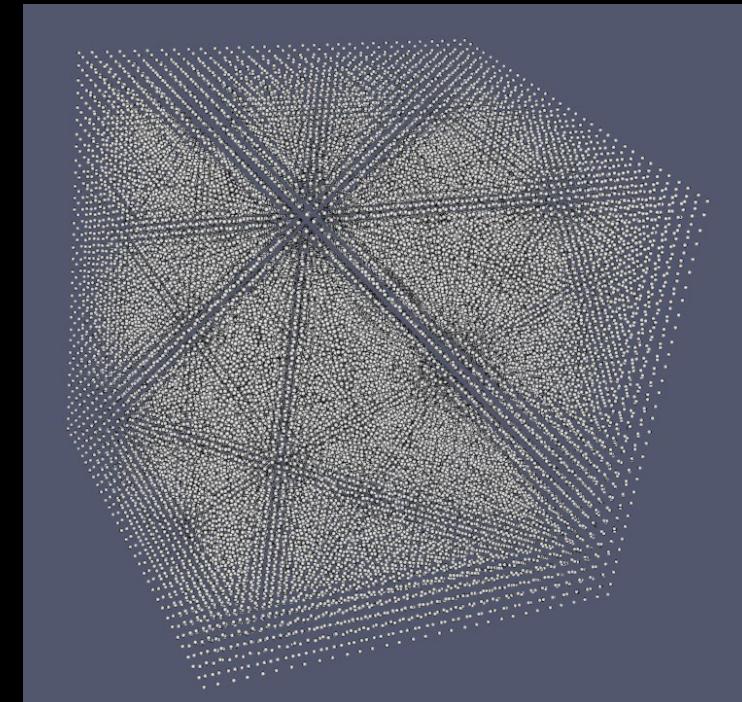


Click on the eye to display
the data in the view.



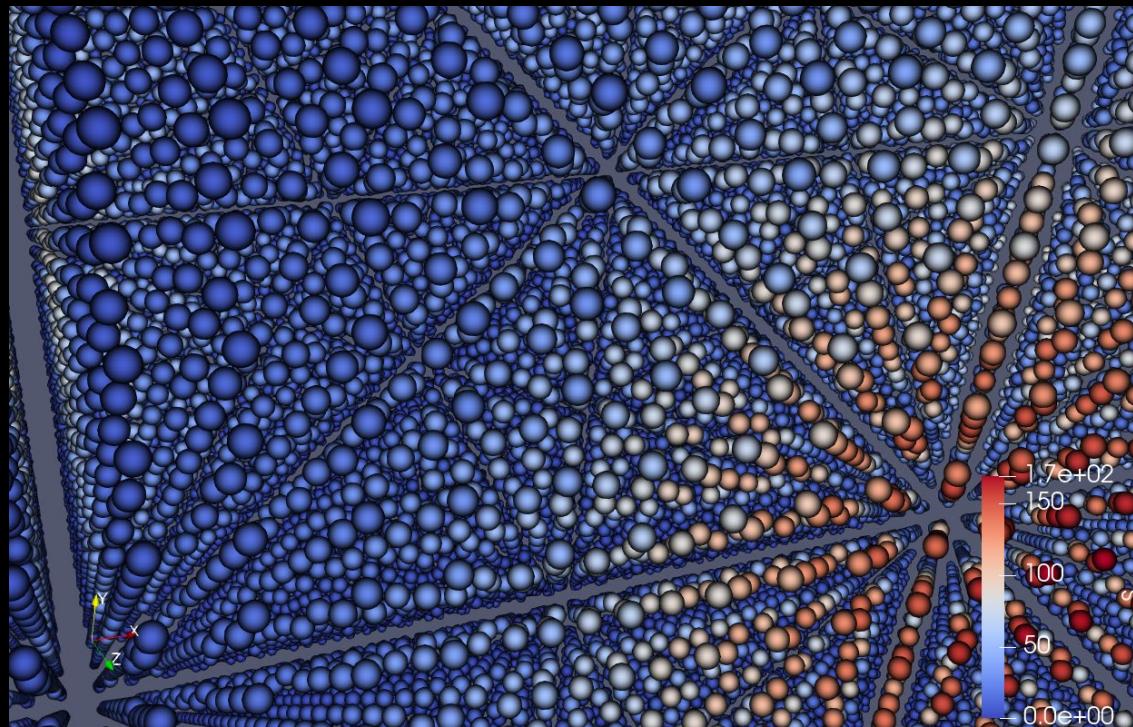
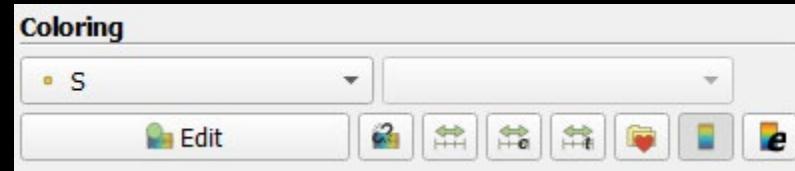
By default data is shown as
an outline.

Change the representation to
Point Gaussian to display the
values in the grid.



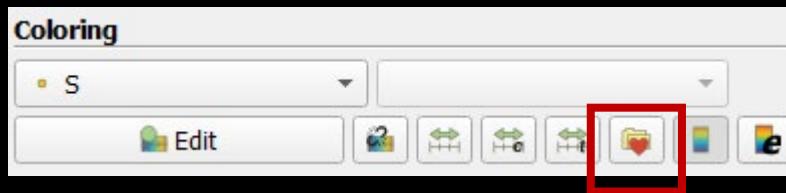
Displaying colors

The CSV file contained an S column of values. To display these using a color scale we need to select the data field in the Coloring section:

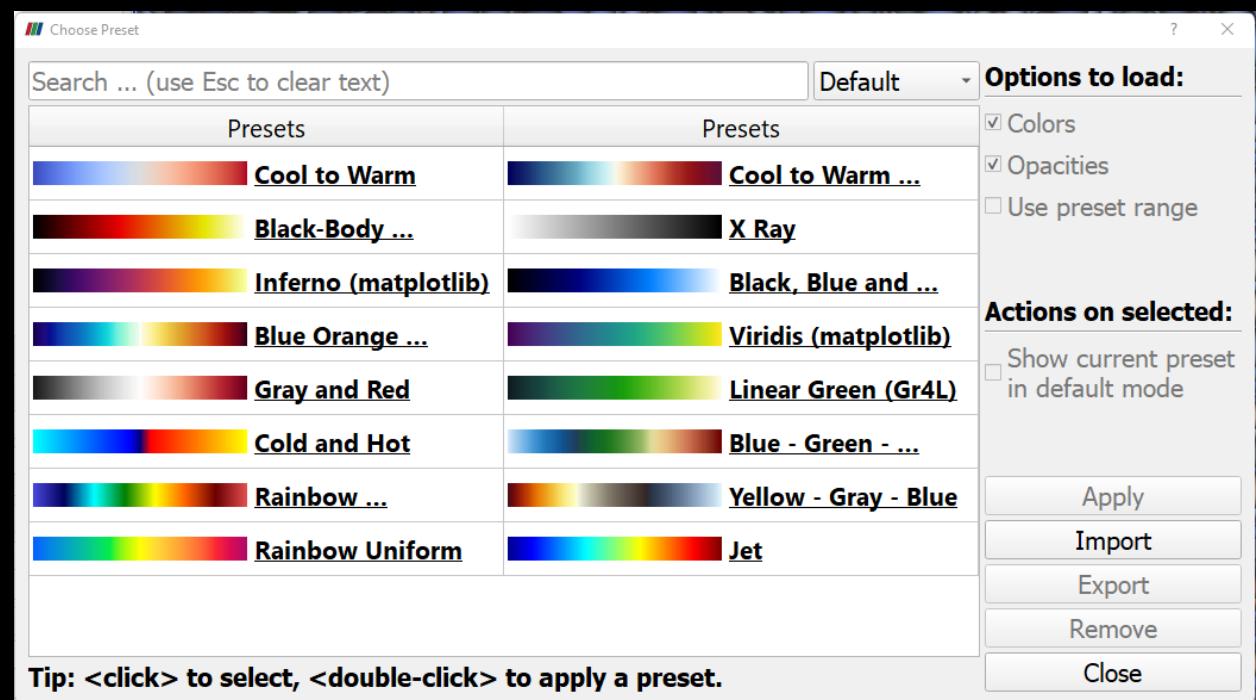


Changing color scale

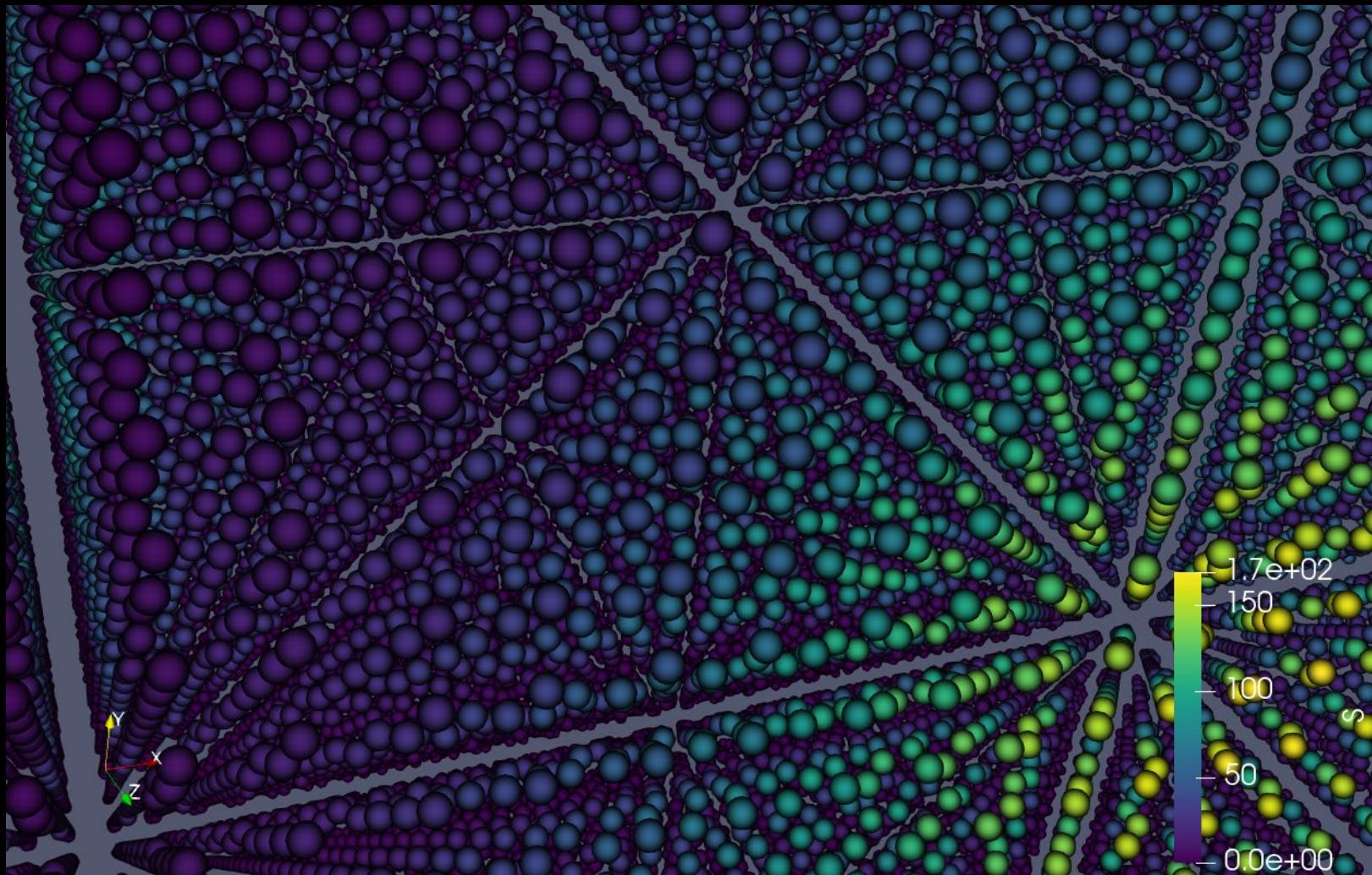
Changing color scale can be done by selecting the folder with a heart.



A selection of color scales are displayed. Select a color map and click Apply to change.

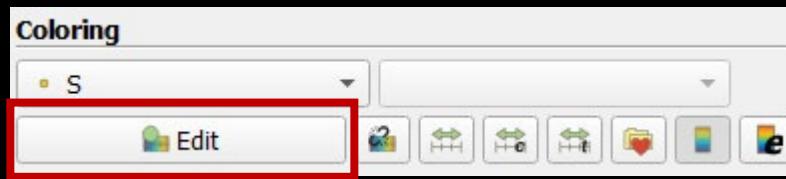


Changing color scale



Changing color scale

The selected color scale can be edited by clicking the Edit button



A selection of color scales are displayed. Select a color map and click Apply to change.

Color Map Editor

Search ... (use Esc to clear text)

Array Name: S

Automatic Rescale Range Mode

Interpret Values As Categories

Rescale On Visibility Change

Mapping Data

Transparency transfer function

Data: 0 Color transfer function

Enable freehand drawing of opacity transfer function

Use log scale when mapping data to colors

Enable opacity mapping for surfaces

Use log scale when mapping data to opacity

Data Histogram

Display data histogram

Automatically recompute data histogram

Number of bins:

Color Mapping Parameters

Color Space Diverging

Nan Color

Nan Opacity

Color Discretization

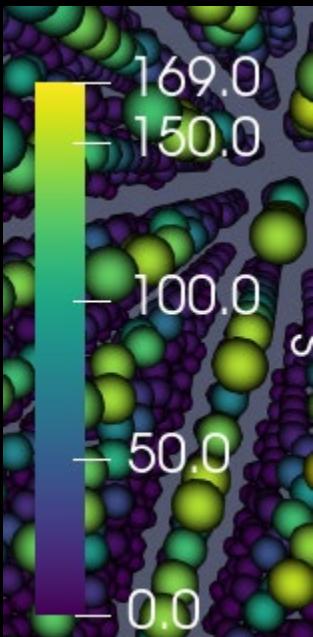
Discretize

Number Of Table Values

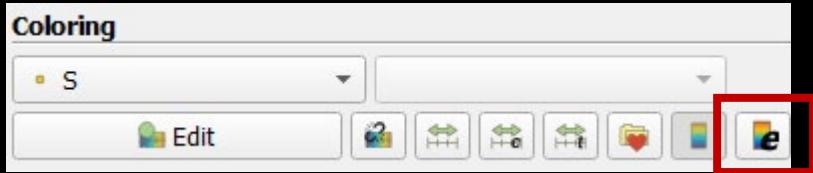
Color Map Editor interface showing various settings for color mapping, including a preview window, mapping data controls, and color mapping parameters.

Changing the legend

The color scale legend can be changed by selecting the button with an e.



Setting the label format using format strings 6 characters wide with one decimal in fixed format.



Search ... (use Esc to clear text)

Auto Orient

Orientation

Window Location

Position

Title Texts

Title

Component

Title Justification

Horizontal Title

Title Font Properties

Arial

Text/Annotation Font Properties

Arial

Labels

Automatic Label Format

Label Format

Draw Tick Marks

Draw Tick Labels

Use Custom Labels

Add Range Labels

Range Label Format

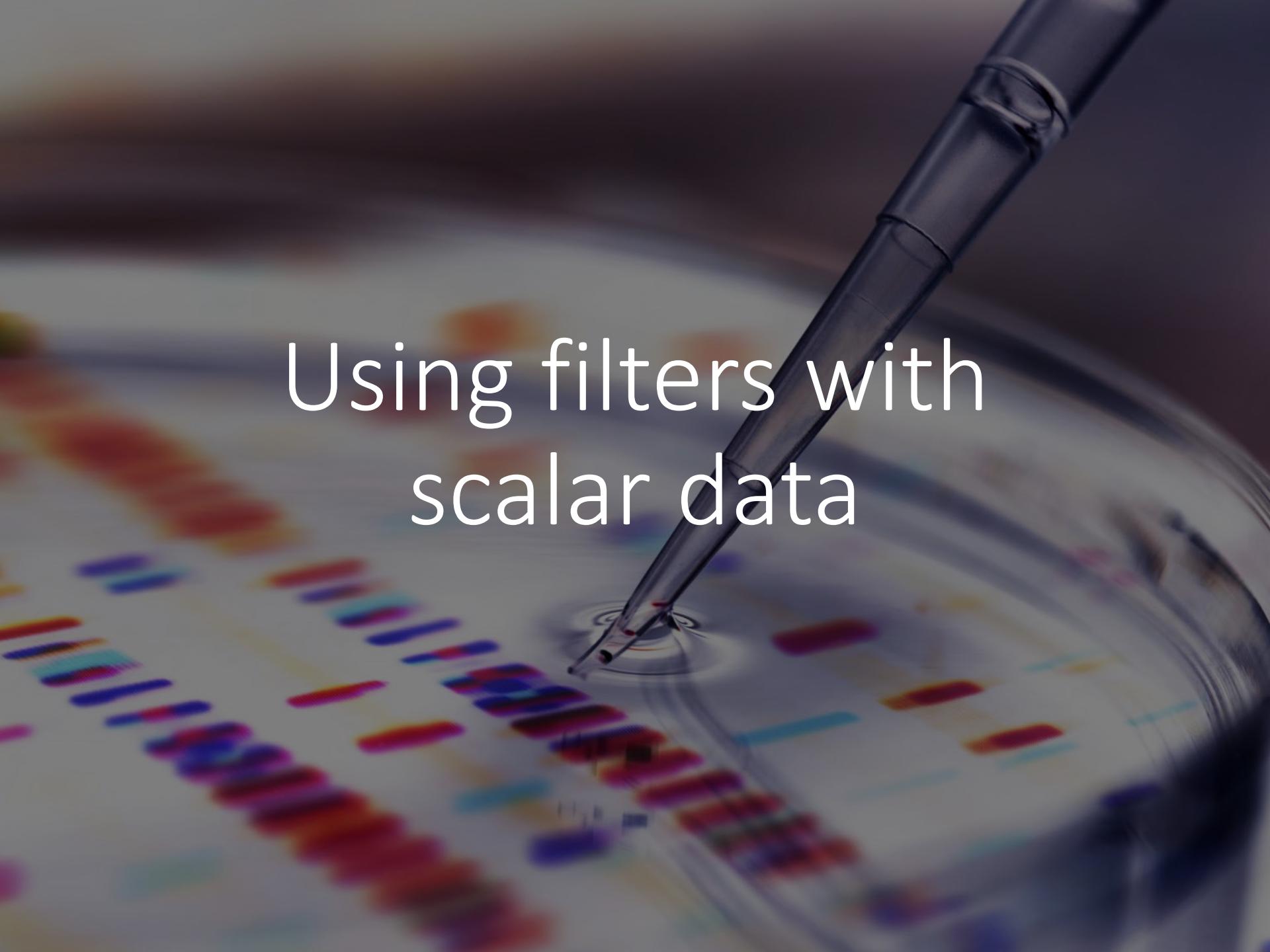
Annotations

Draw Annotations

Click the gear to show advanced options.

A close-up photograph of a car's tachometer. The needle is positioned exactly halfway between the number 5 and the number 6 on the scale. The numbers 5 and 6 are illuminated in red, while the numbers 4, 7, and 8 are in white. The text "000 rpm" is visible at the bottom of the scale.

Demo time

A close-up photograph of a laboratory pipette dispensing a small amount of clear liquid onto a circular pattern of colored bands on a light-colored surface. The pattern consists of several concentric rings of horizontal bands in various colors, including red, blue, green, and yellow. The pipette is positioned diagonally across the frame, with its tip pointing towards the bottom left.

Using filters with scalar data

Common filters

Contour. Creates contour lines or surfaces.



Calculator. Computes new fields based on existing fields.

Slice. Create a slice plane.



Clip. Clips the data at a certain plane



Threshold. Extracts data for a certain criteria



Subset. Extracts a sub grid of the data



Glyph. Generate glyphs at data points



Warp by vector. Warps data mesh with a vector field



Stream tracer. Generate glyphs at data points



Glyph cloud

Glyph Source

Glyph Type Sphere

Orientation

Orientation Array No orientation array

Scale

Scale Array No scale array

Scale Factor X ↻

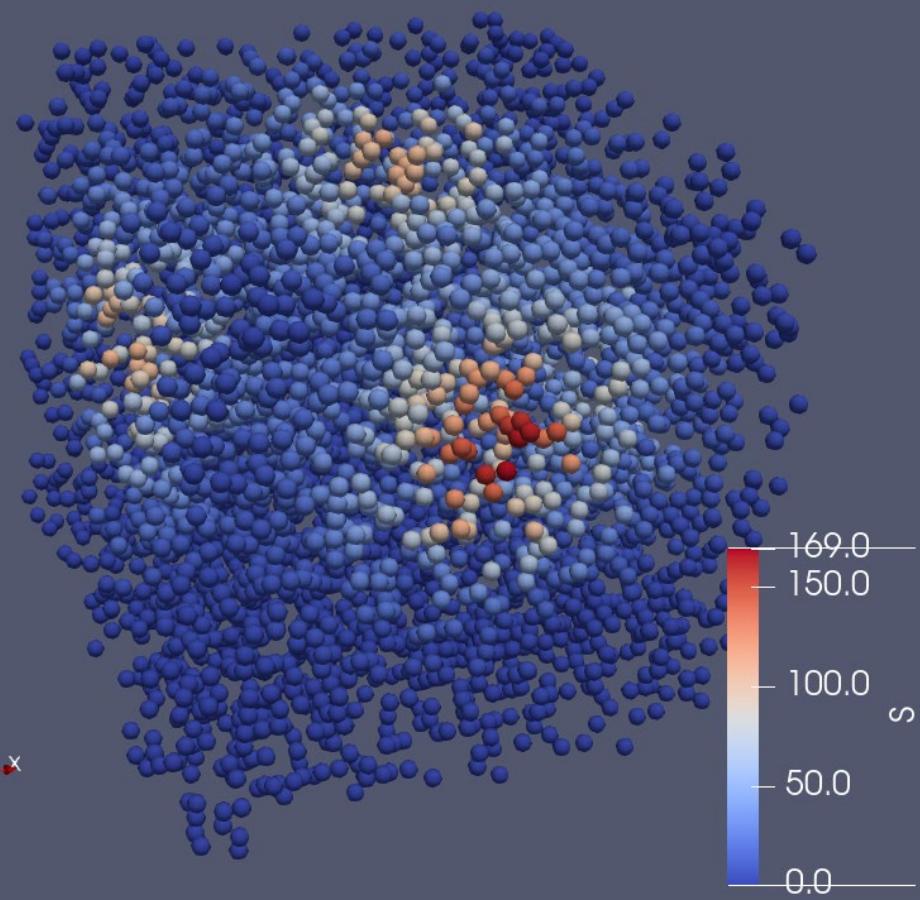
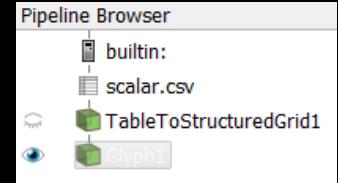
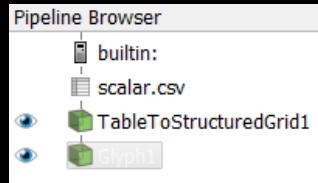
Masking

Glyph Mode Uniform Spatial Distribution (Bounds Based)

Maximum Number Of Sample Points 5000

Seed 10339

We need to turn off the TableToStructuredGrid



A close-up photograph of a car's tachometer. The needle is positioned exactly halfway between the number 5 and the number 6 on the scale. The numbers 5 and 6 are illuminated in red, while the numbers 4, 7, and 8 are in white. The text "000 rpm" is visible at the bottom of the scale.

Demo time

Exercise

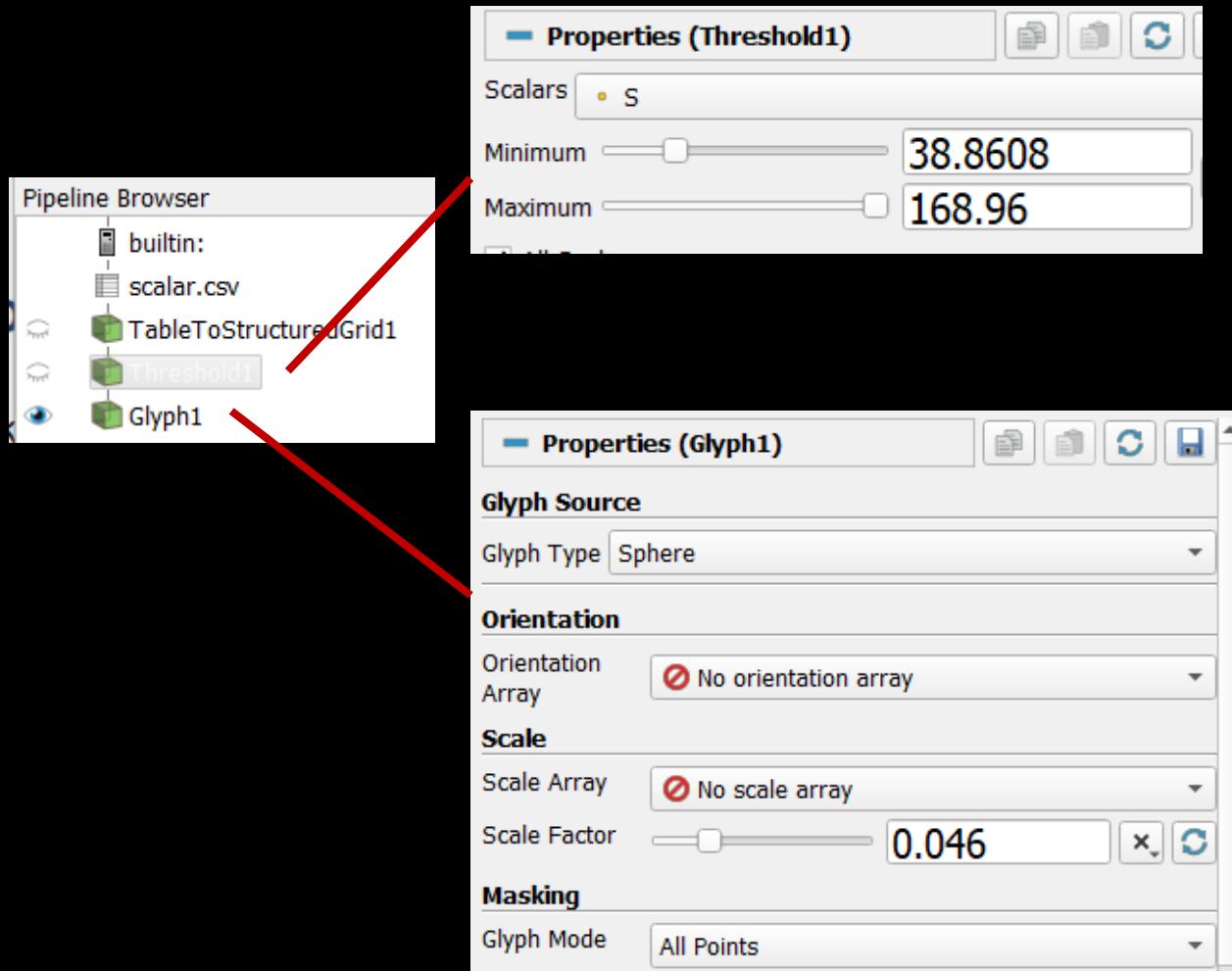
Play with the parameters to see how it affects the visualisation.

Thresholding data

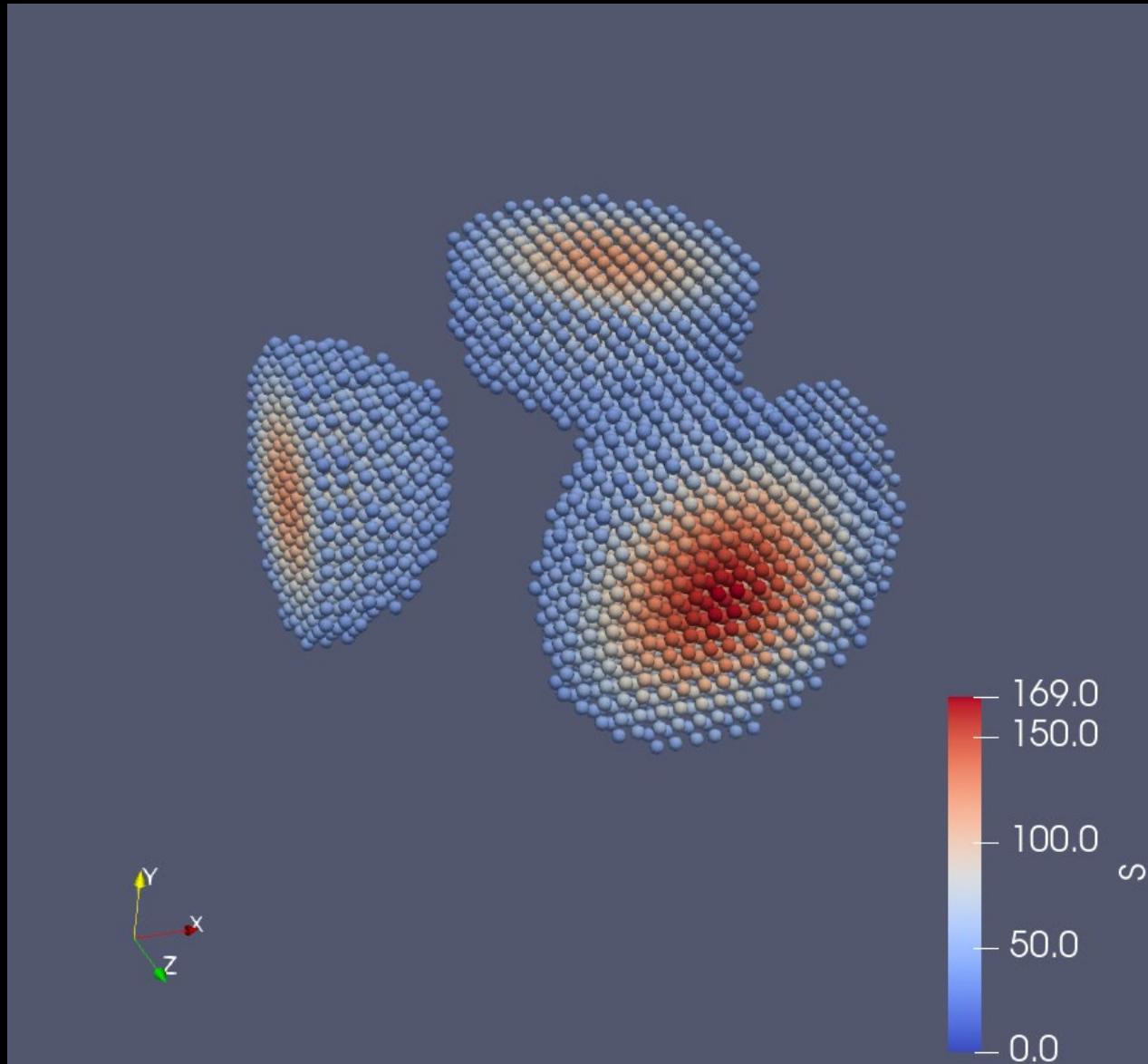
We want to limit data in the glyph visualisation.

We Create the following pipeline

We use Delete to remote the existing filters and apply



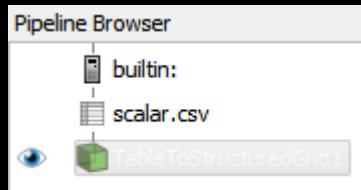
Thresholding data



Creating a cutting plane

Delete all filters except the
TablesToStructuredGrid

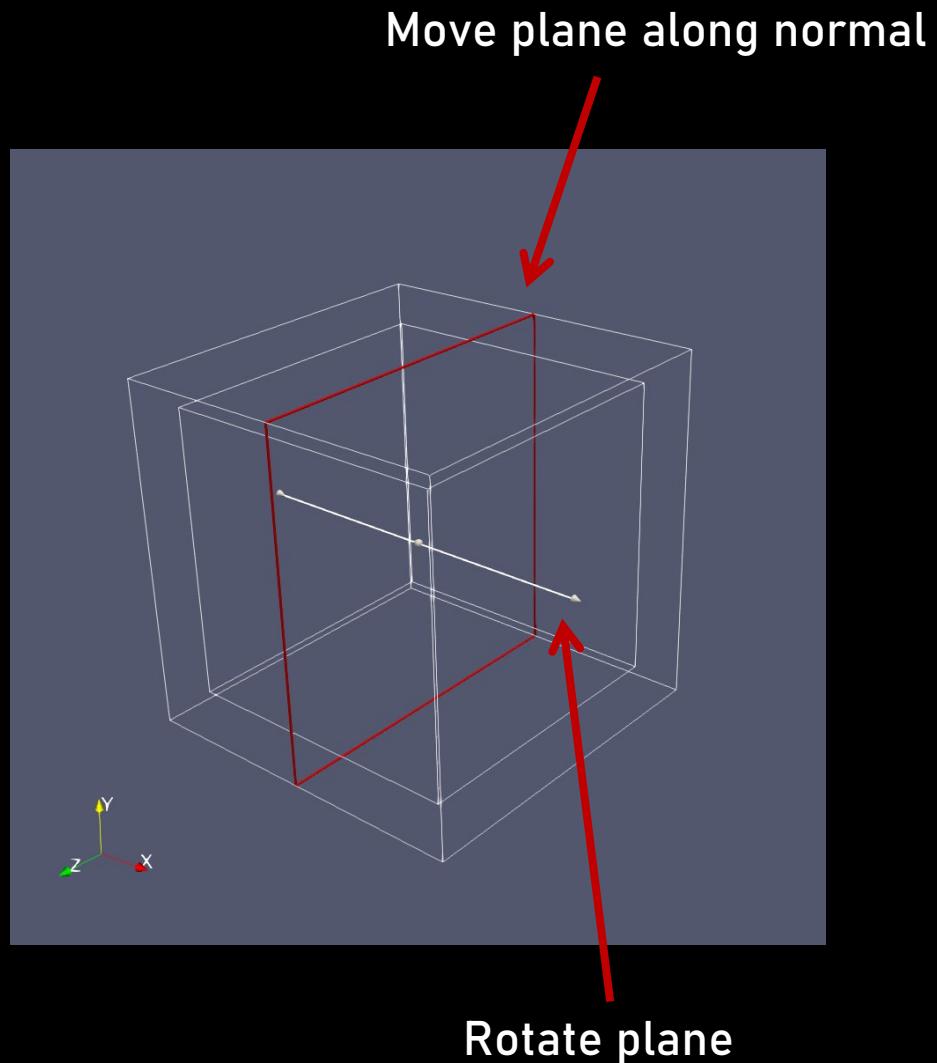
Set the filter representation
to Outline



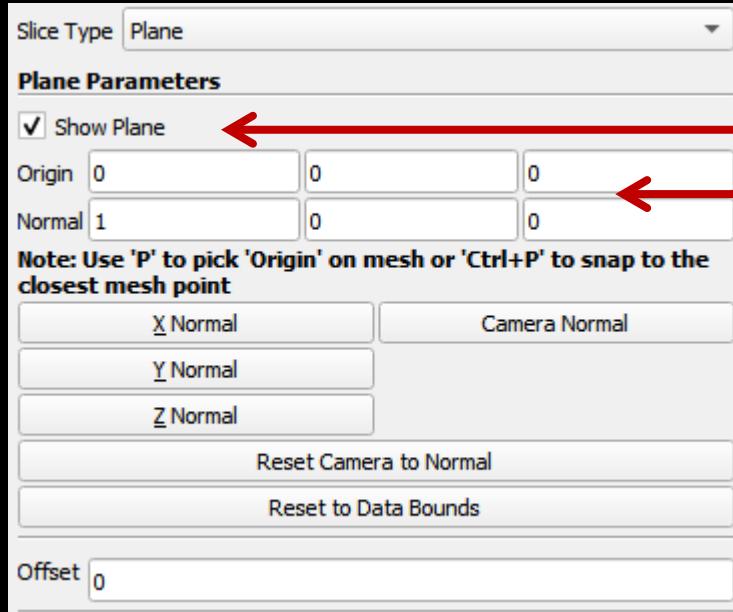
Add a Slice filter



The view shows a
representation of the slice
plane.



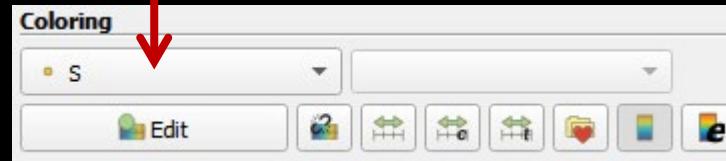
Creating a cutting plane



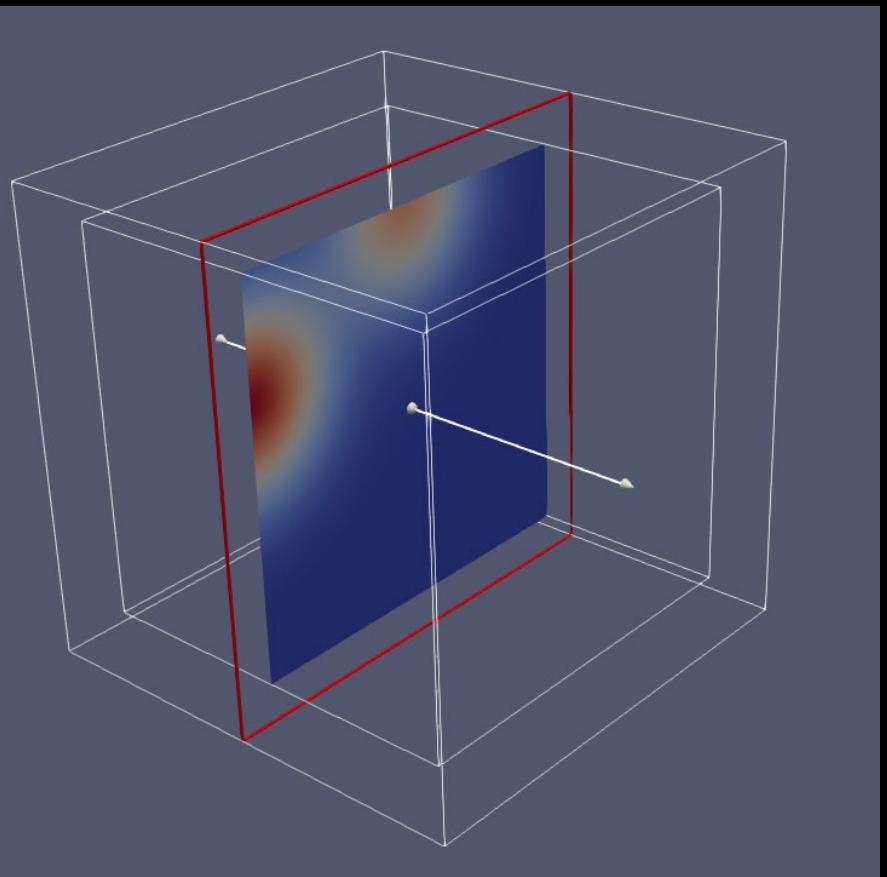
Show a visual representation of the plane

Manual plane parameters

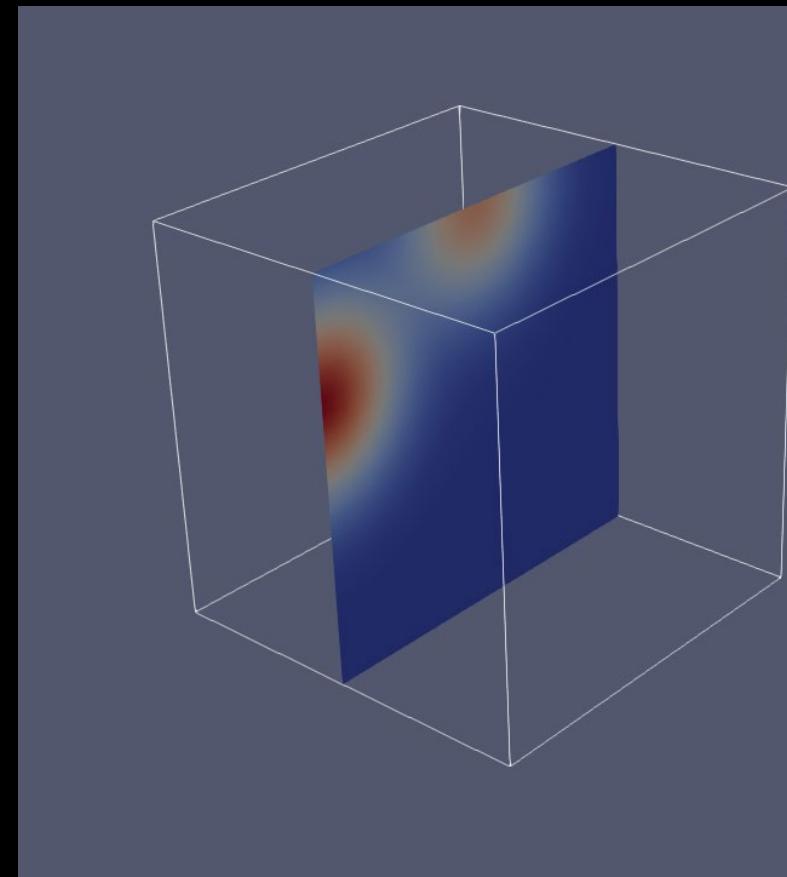
Make sure to color the plane according to the S field



Creating cutting plane



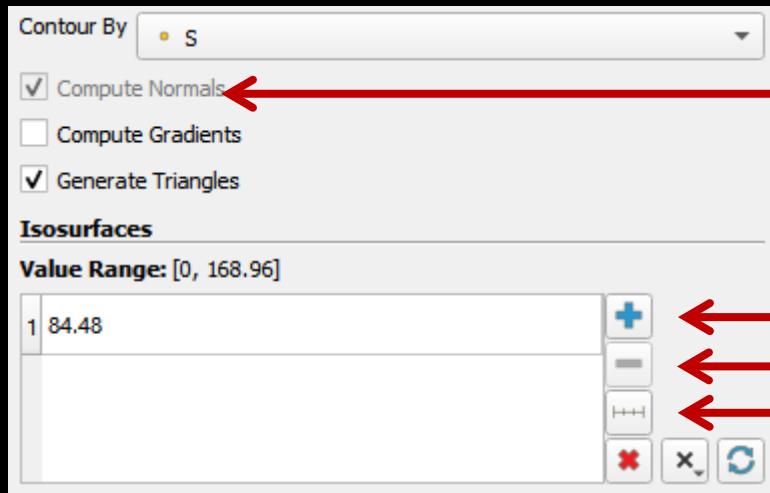
Show Plane



Show Plane

Displaying cutting planes with contours

Add a Contour filter



Show a visual representation
of the plane

Add contour line
Remove contour line
Create multiple contour lines

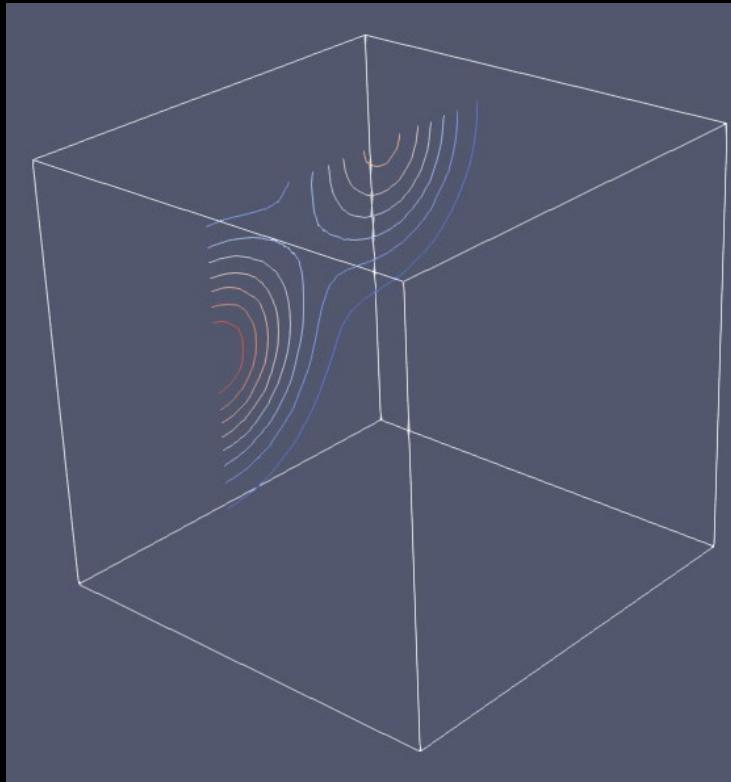
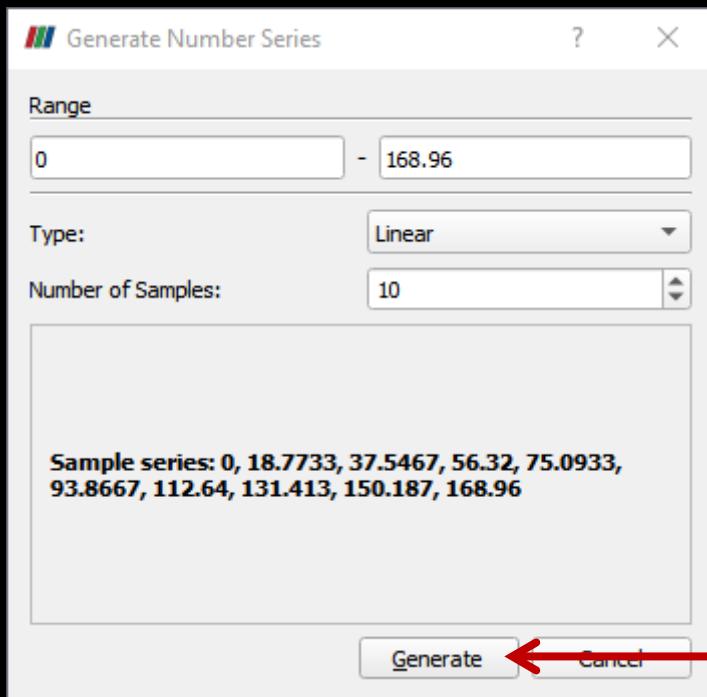
Clear all contour lines

Displaying cutting planes with contours

Now we erase all existing contour lines



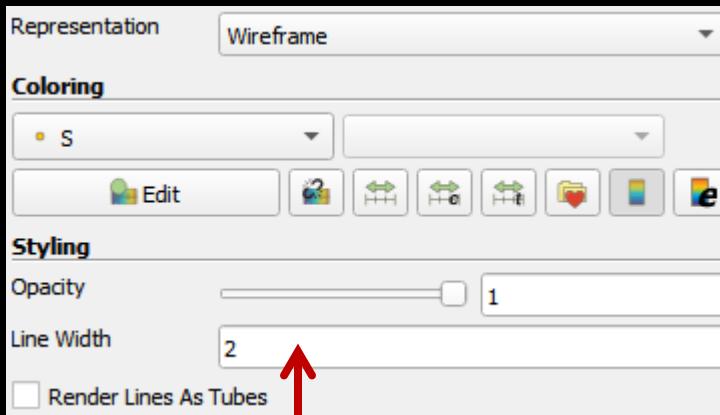
Create a range of contour lines



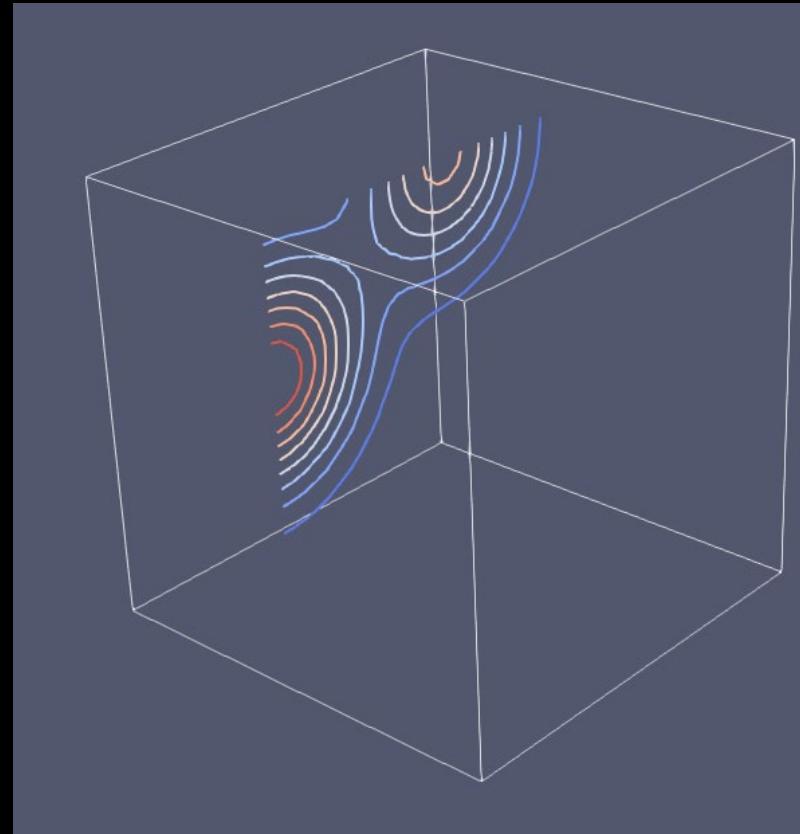
Click Generate and
then Apply

Displaying cutting planes with contours

We can control the contour better by switching to a "Wireframe" Representation



Line thickness

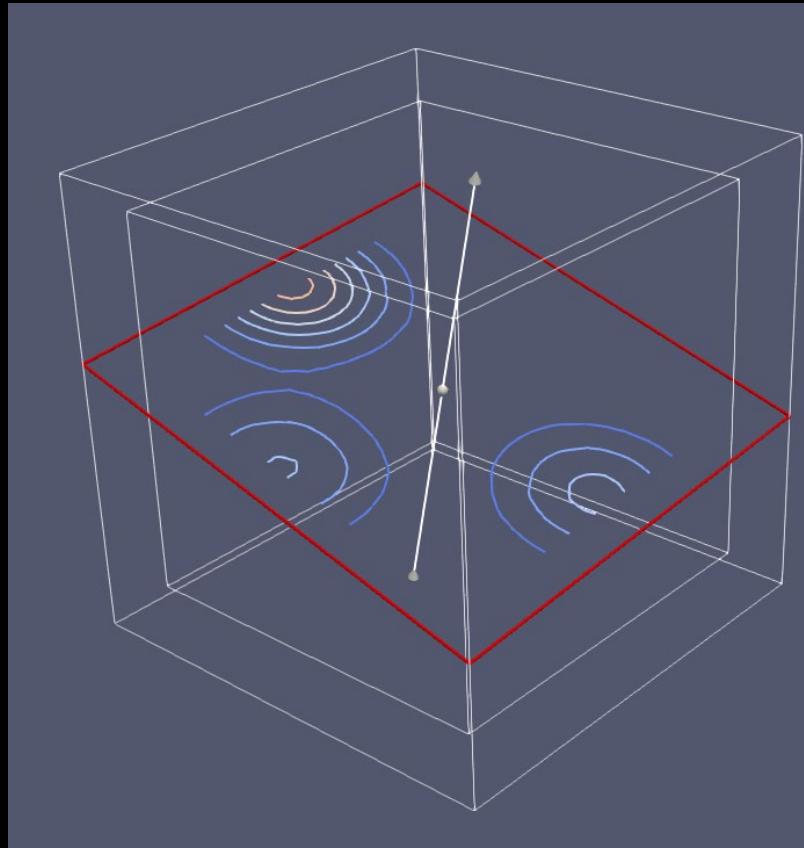


Displaying cutting planes with contours

Select the Slice filter and enable

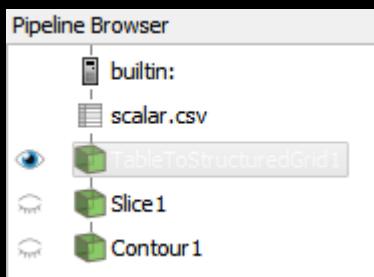


This enables you to move the cutting planes displaying contours on the plane.

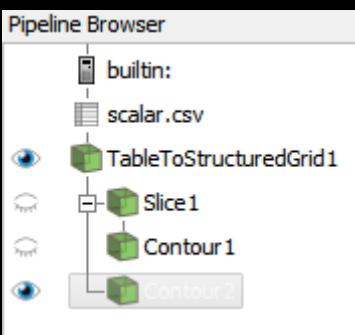
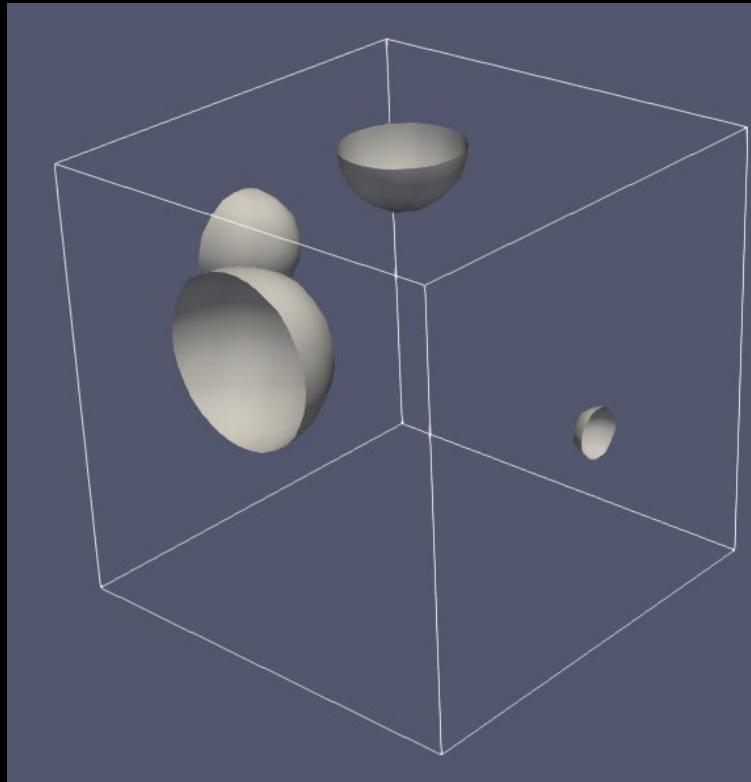


Displaying iso surfaces

Select TableToStructuredGrid1 and disable the other filters



Create a Contour filter from the grid

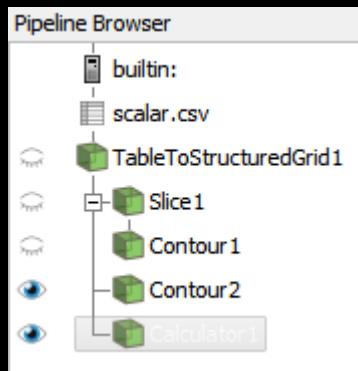


Notice that it is not possible to color the iso lines as S is not available anymore.

We need a way of duplicating the S field

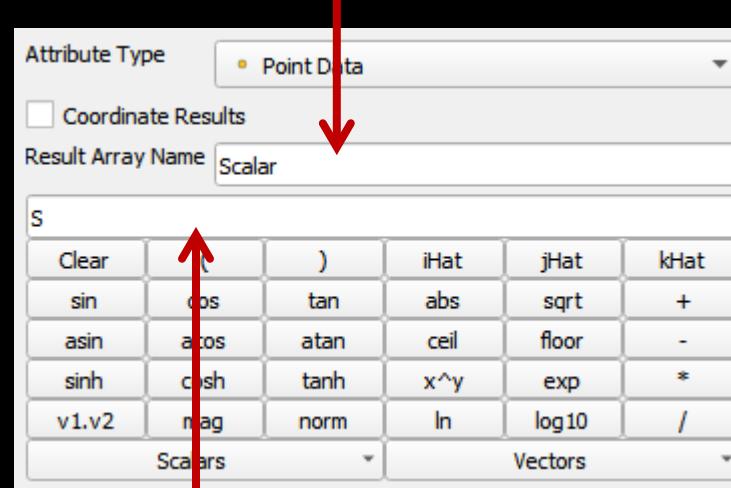
Displaying iso surfaces

Create a Calculator filter from the TableGrid



We also need to move the Contour2 object after the Calculator filter.

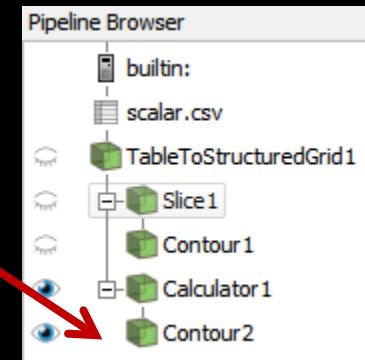
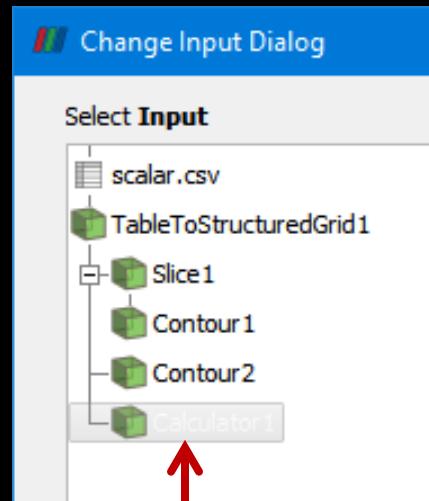
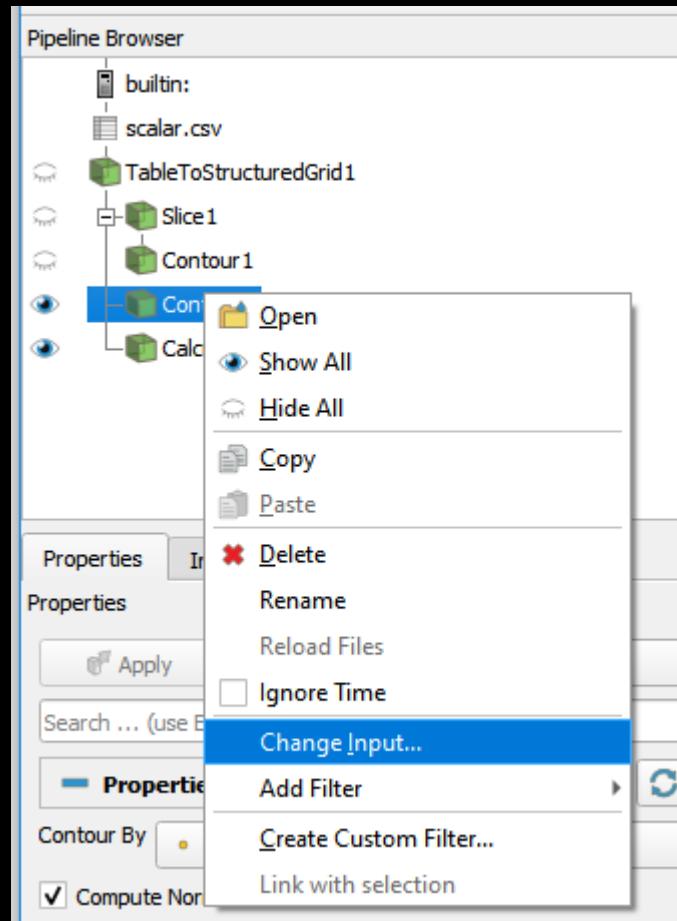
Name the result of the expression. We name it Scalar



Enter the expression here. In this case just return the scalar value from S

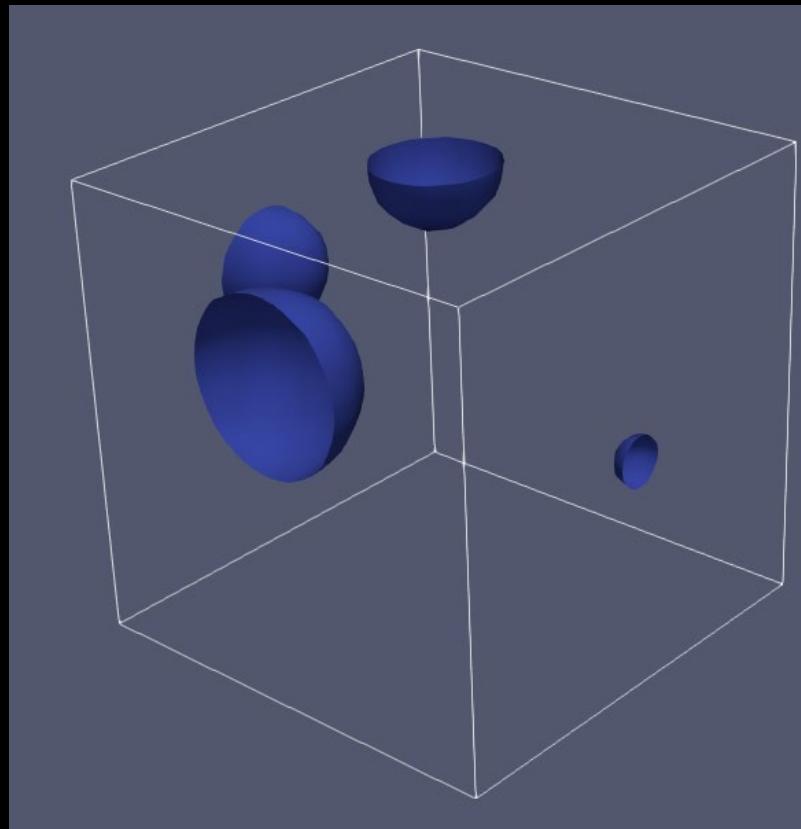
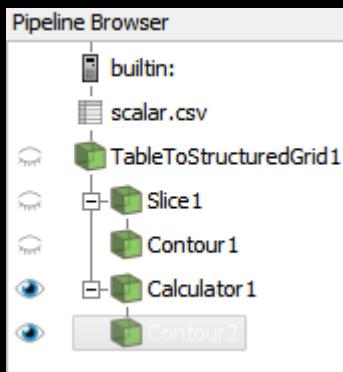
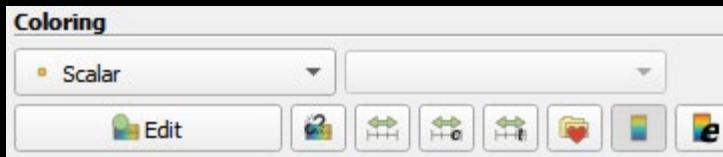
Displaying iso surfaces

Moving Contour2 after Calculator

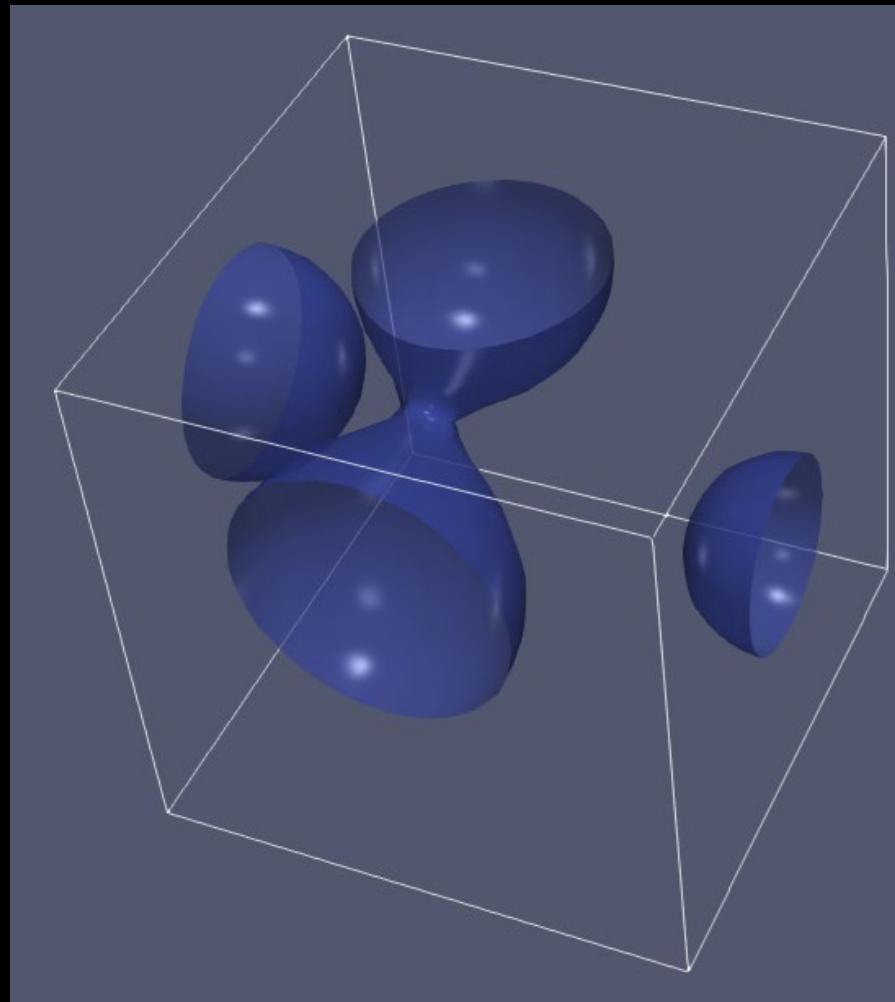
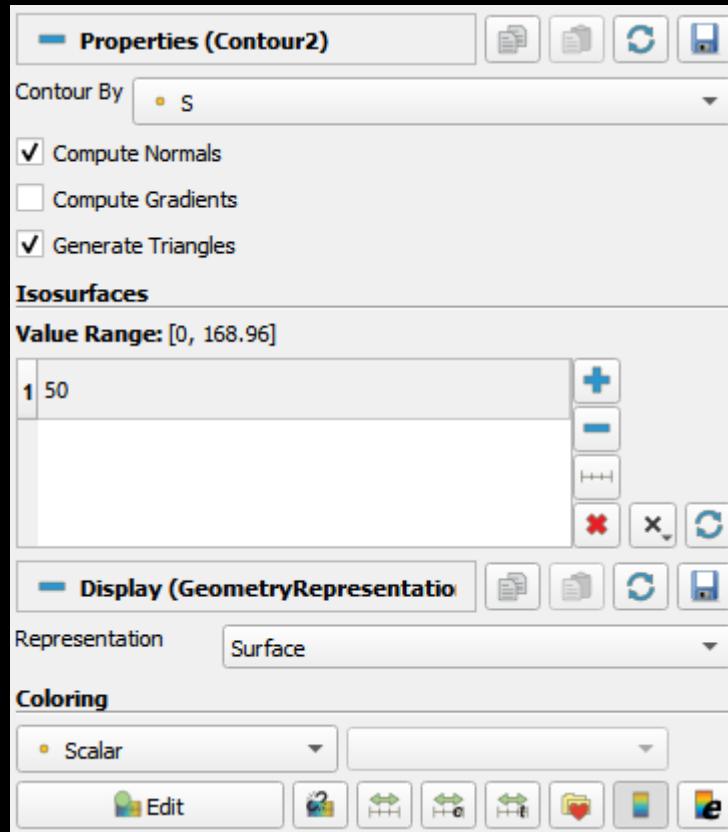


Displaying iso surfaces

We can now select the computed field in the Contour filter



Displaying iso surfaces



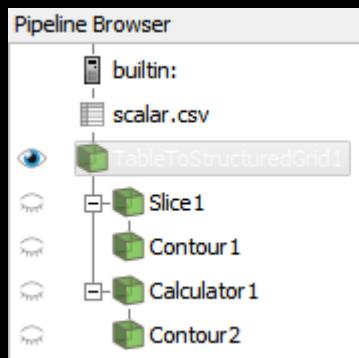
A close-up photograph of a car's tachometer. The needle is positioned exactly halfway between the number 5 and the number 6 on the scale. The numbers 5 and 6 are illuminated in red, while the numbers 4, 7, and 8 are in white. The text "000 rpm" is visible at the bottom of the scale.

Demo time

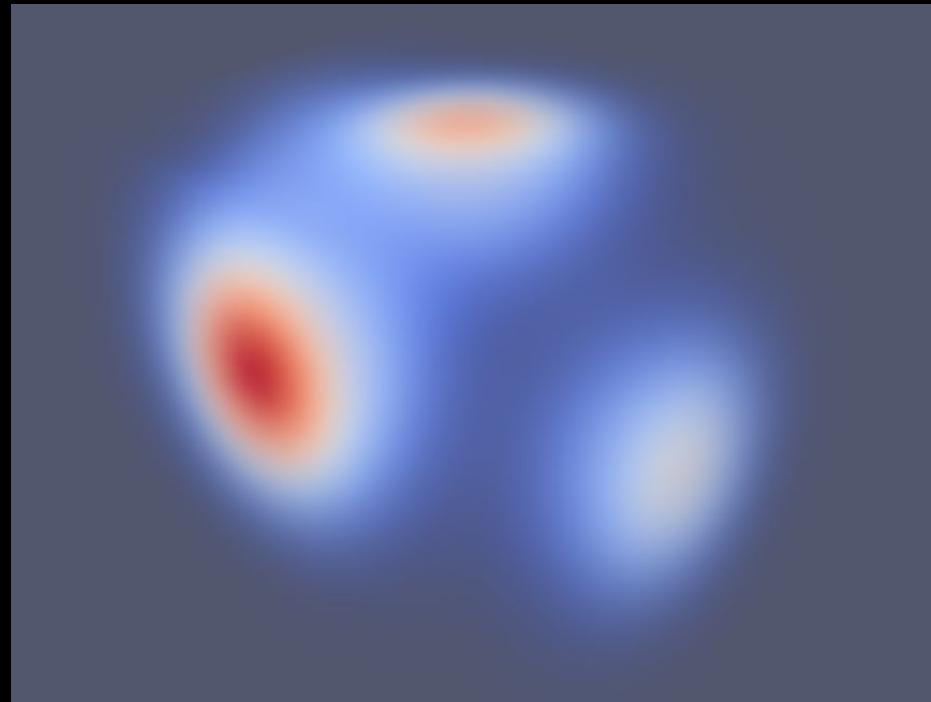
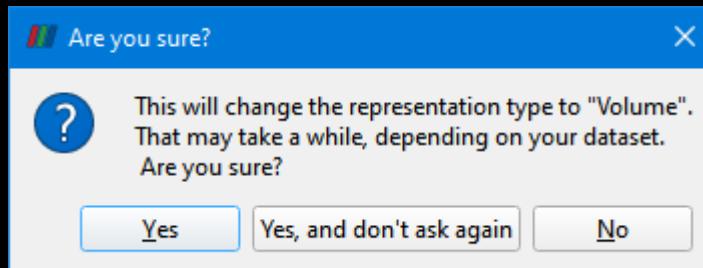


Displaying data as a volume

Select TableToStructuredGrid1 and disable the other filters



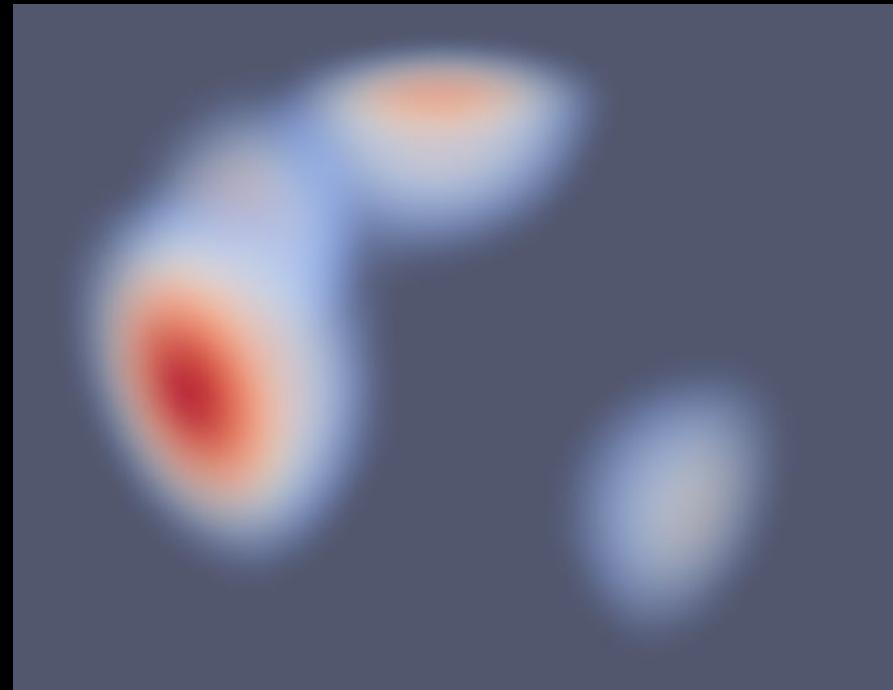
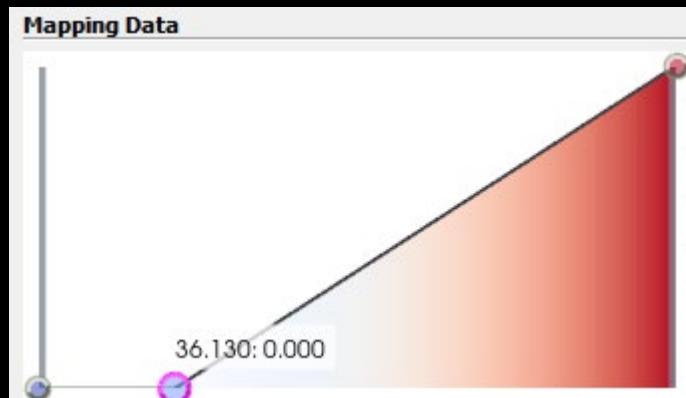
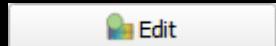
Set representation to "Volume"



Controlling the opacity properties

To better see structures in the volume
the transparency transfer function can
be modified.

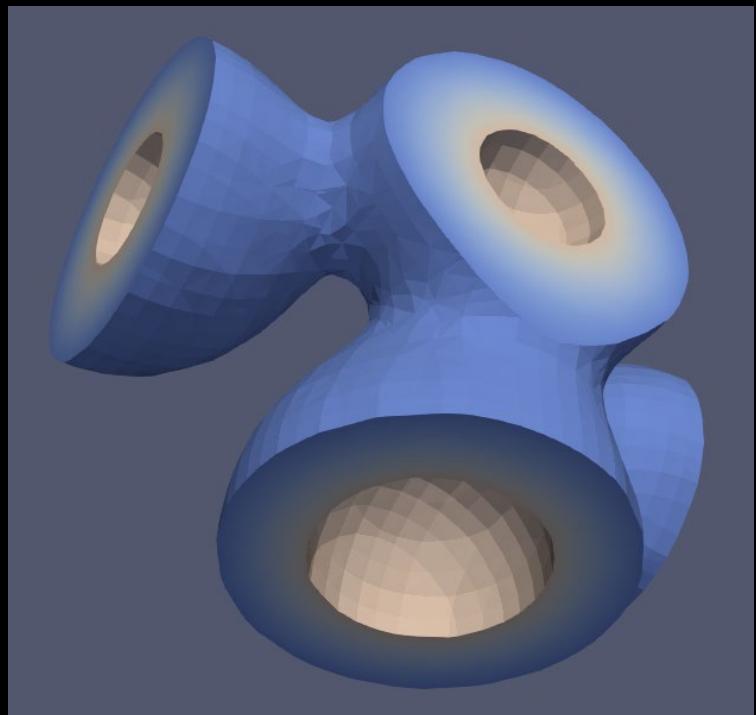
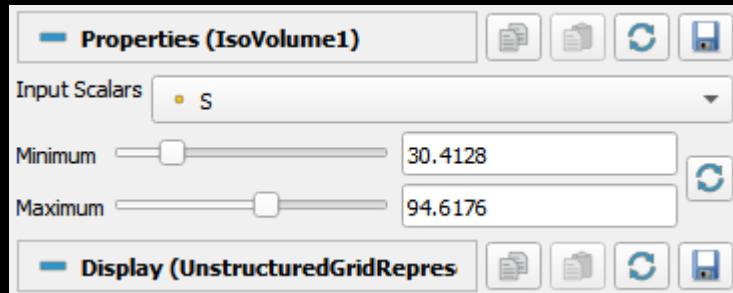
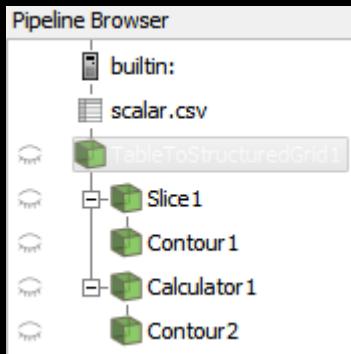
Click



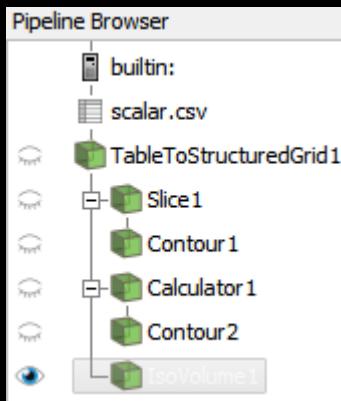
Add a point by clicking on the curve.
Drag the point down to make values
left of the point transparent

Creating iso volumes

Select TableToStructuredGrid1
and disable the other filters

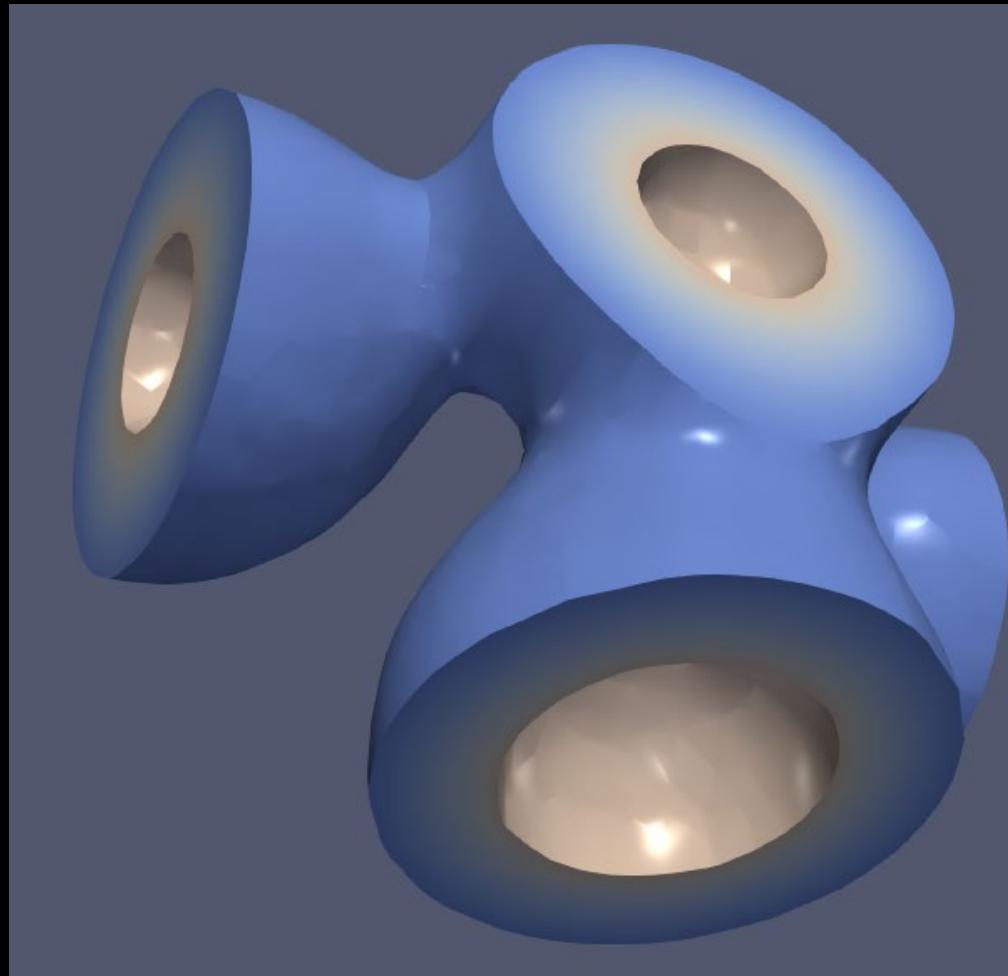
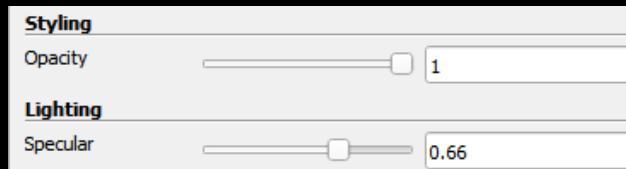
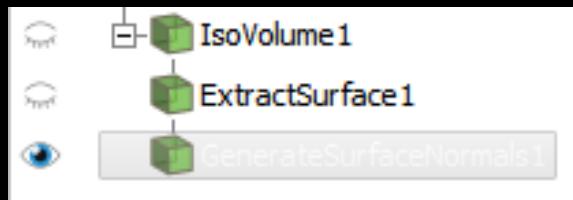


Add a IsoVolume filter from the
menu or searching for it



Creating smooth iso volumes

Add ExtractSurface and a GenerateNormals filter to the IsoVolume filter



A close-up photograph of a car's tachometer. The needle is positioned exactly halfway between the number 5 and the number 6 on the scale. The numbers 5 and 6 are illuminated in red, while the numbers 4, 7, and 8 are in white. The text "000 rpm" is visible at the bottom of the scale.

Demo time

Reset ParaView

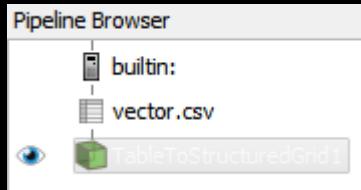


Reading vector data

Open the vector.csv file

- File/Open...
- Find the vector.csv file.
- Click Apply to load the file

Create a Grid from the data



The Properties panel for the 'TableToStructuredGrid1' node. It includes tabs for 'Properties' and 'Connections'. Below are configuration settings:

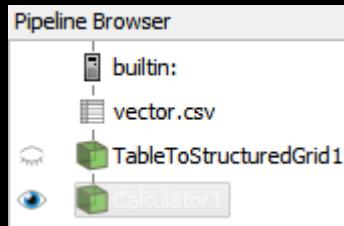
Whole Extent	0	31
	0	31
	0	31
X Column	X32	
Y Column	Y32	
Z Column	Z32	

Vector data from the CSV file is provided as 3 scalar fields.

We need to convert these to vectors

Calculator to the rescue

Combine scalar fields to vectors



A screenshot of the 'Properties (Calculator1)' dialog. The 'Attribute Type' is set to 'Point Data'. The 'Result Array Name' field contains the text 'v'. A red arrow points from this field down to the expression 'Vx*iHat+Vy*jHat+Vz*kHat' in the 'Expression' field below. The 'Expression' field also has a red arrow pointing to it from the text 'Vx*iHat+Vy*jHat+Vz*kHat' in the text block above. The dialog also includes tabs for 'Scalars' and 'Vectors'.

iHat, jHat and kHat
are unit vectors
that can be used to
create a vector
field from scalar
fields

We now have a
vector field we can
visualise. We start
with glyphs

A close-up photograph of a car's tachometer. The needle is positioned exactly halfway between the number 5 and the number 6 on the scale. The numbers 5 and 6 are illuminated in red, while the numbers 4, 7, and 8 are in white. The text "000 rpm" is visible at the bottom of the scale.

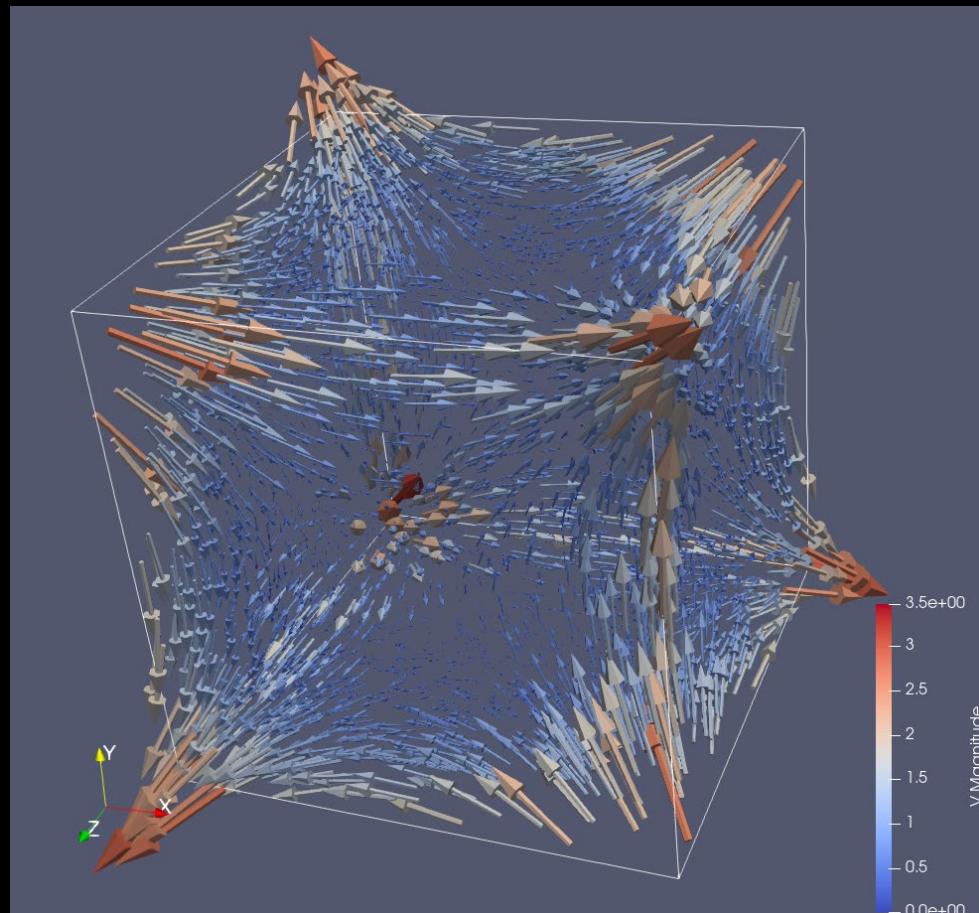
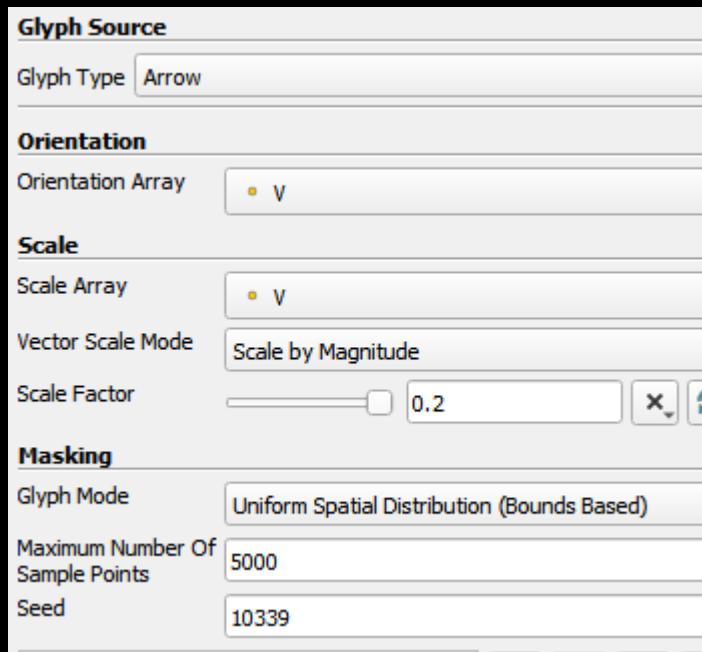
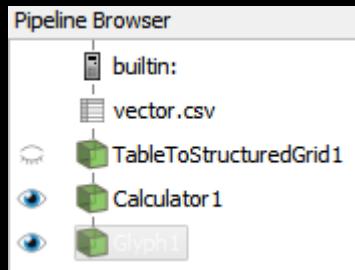
Demo time

The background of the slide features a dynamic, swirling pattern of blue and white smoke or vapor against a solid black background. The smoke is more concentrated on the right side, creating a sense of motion and depth.

Using filters with vector data

Visualise vectors using oriented glyphs

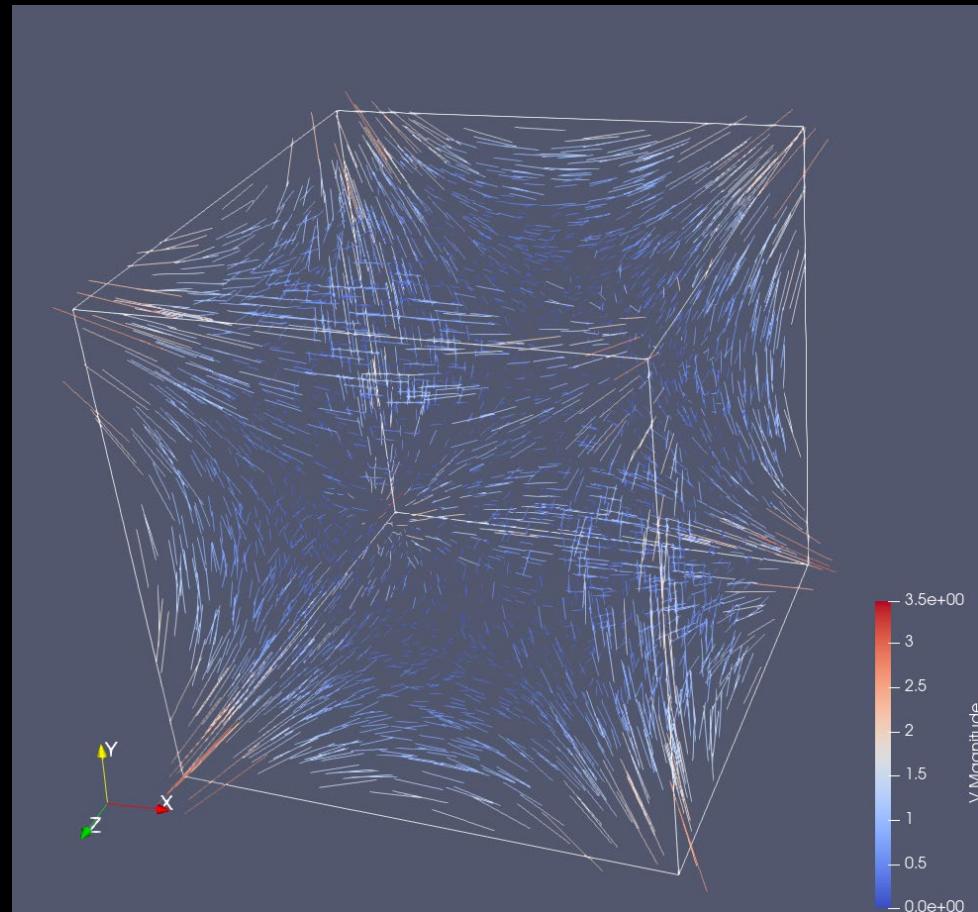
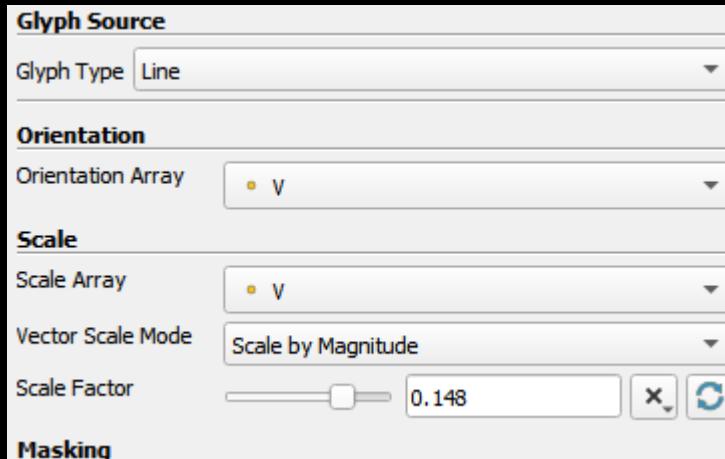
Add a Glyph to the Calculator



Visualise vectors using oriented glyphs

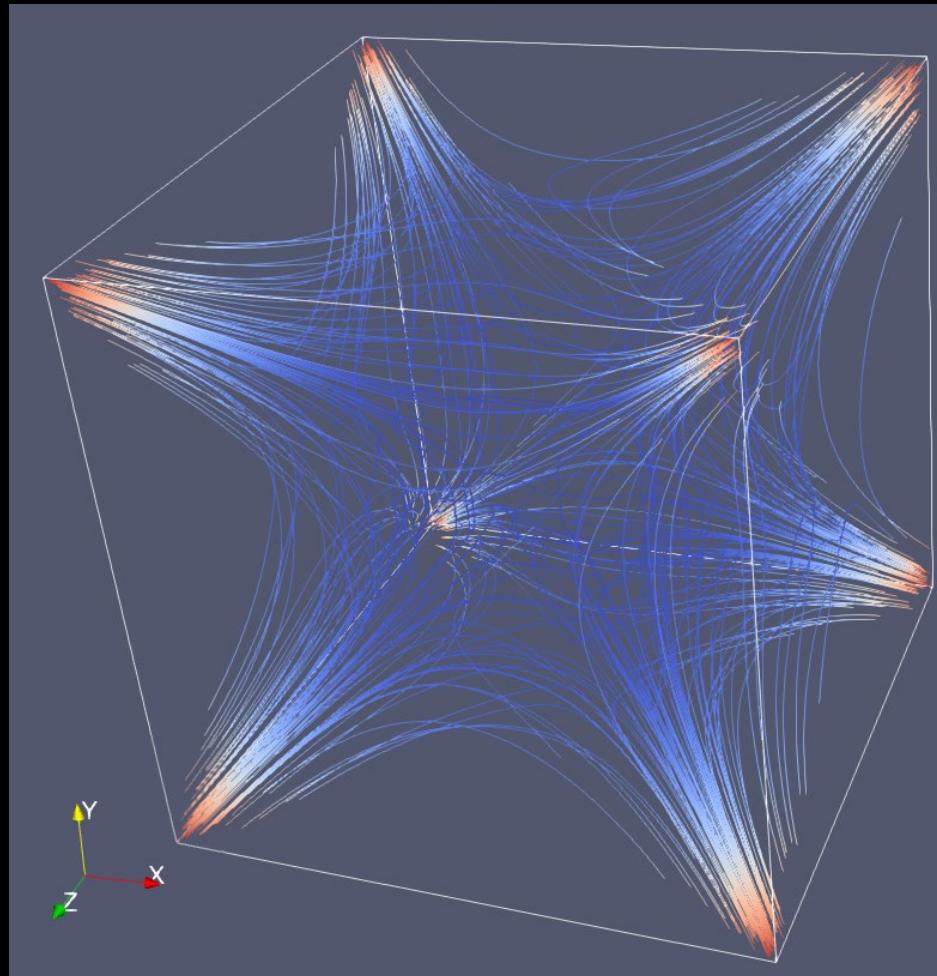
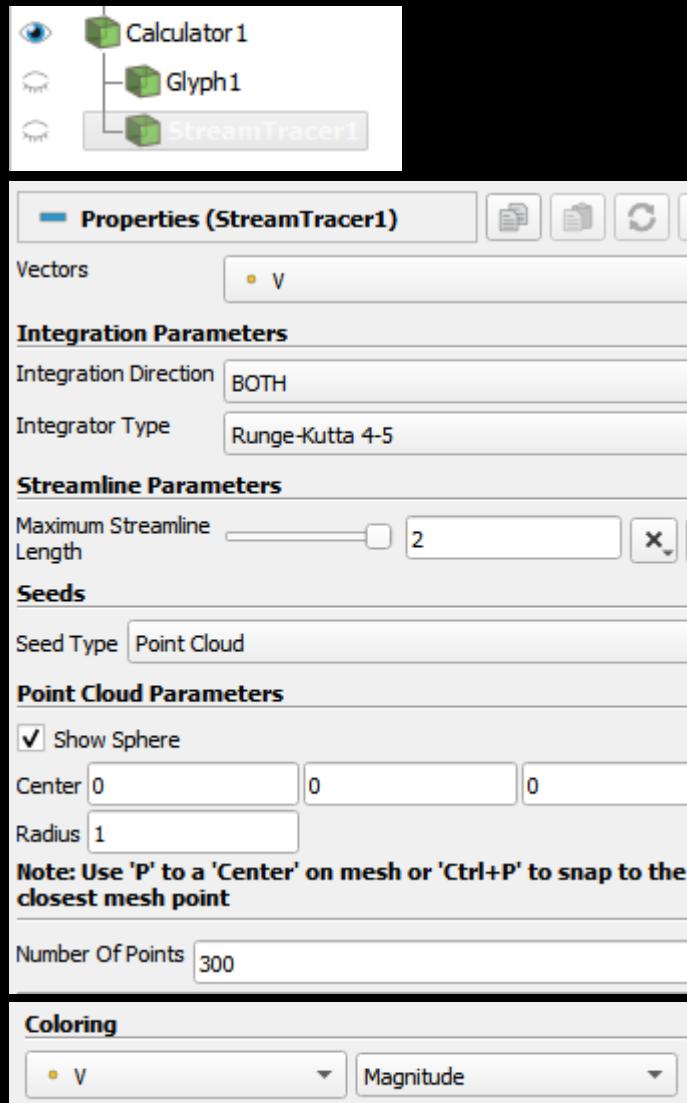
Sometimes it can be beneficial to visualise glyphs as thin lines

Change representation to "Line"



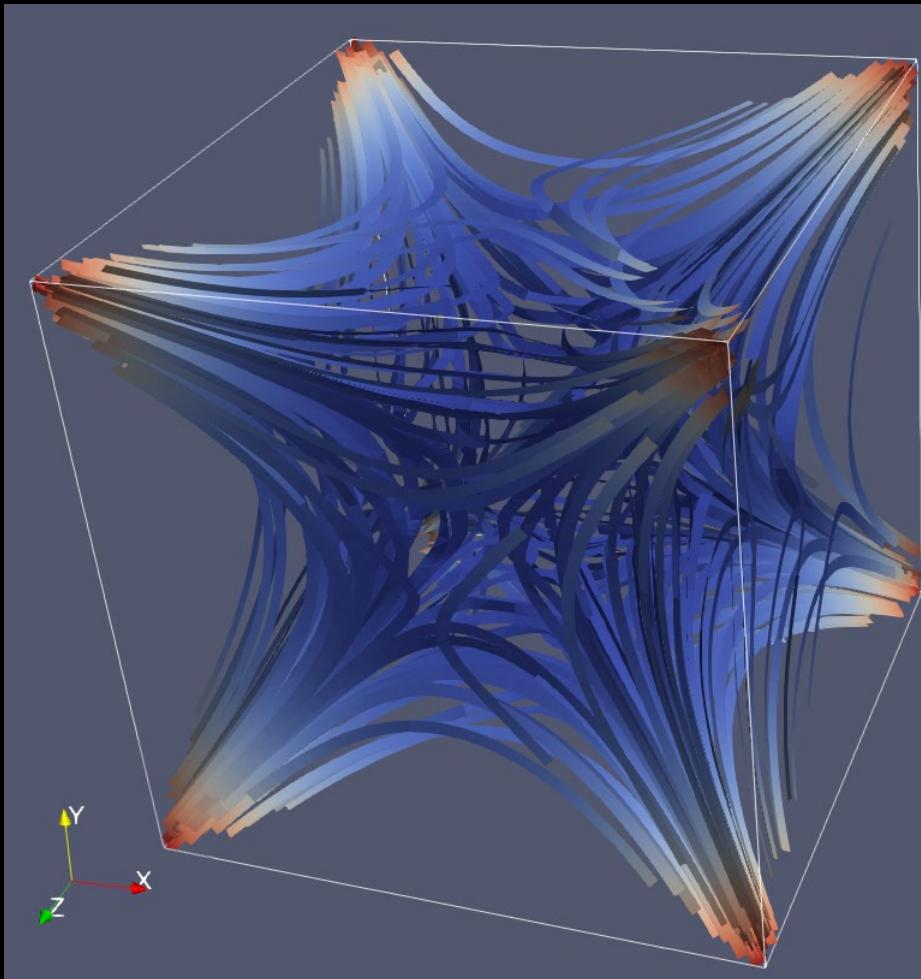
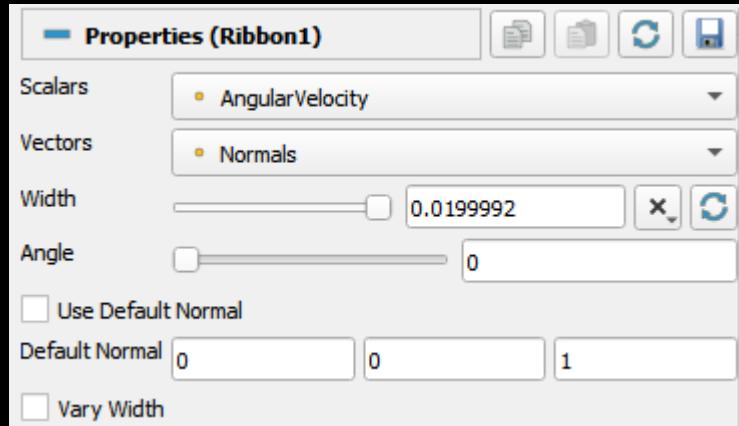
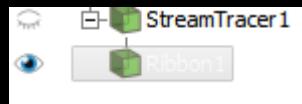
Visualise vectors stream tracer

Add a StreamTracer to the Calculator filter



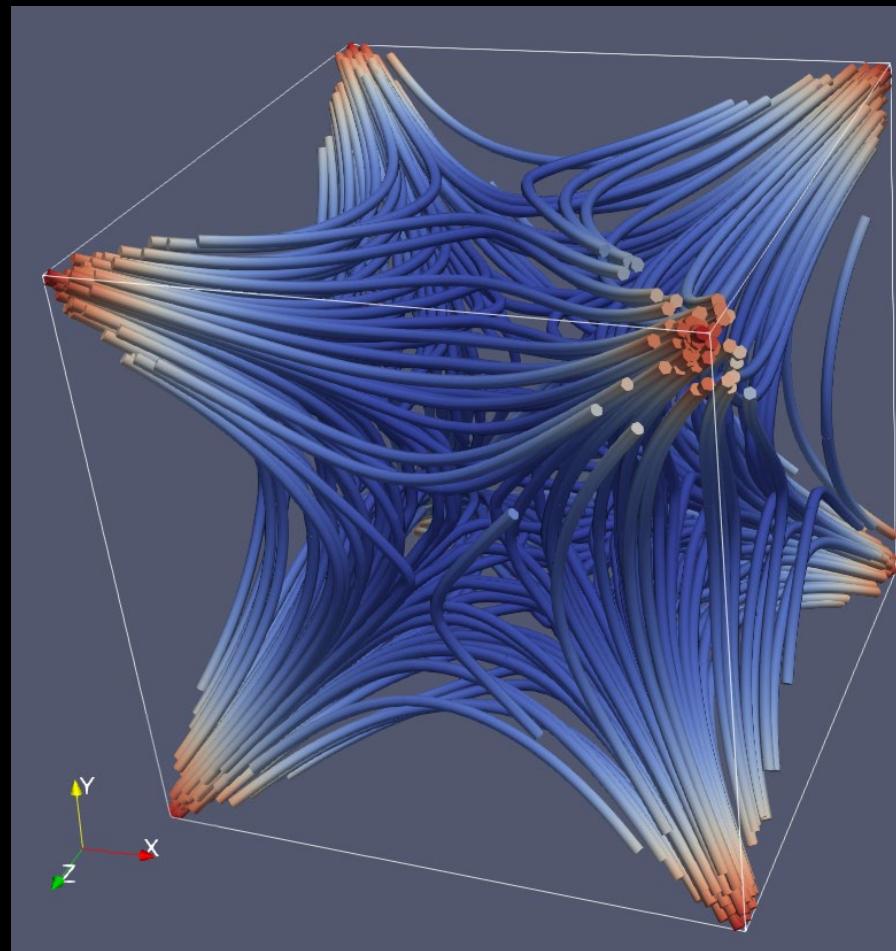
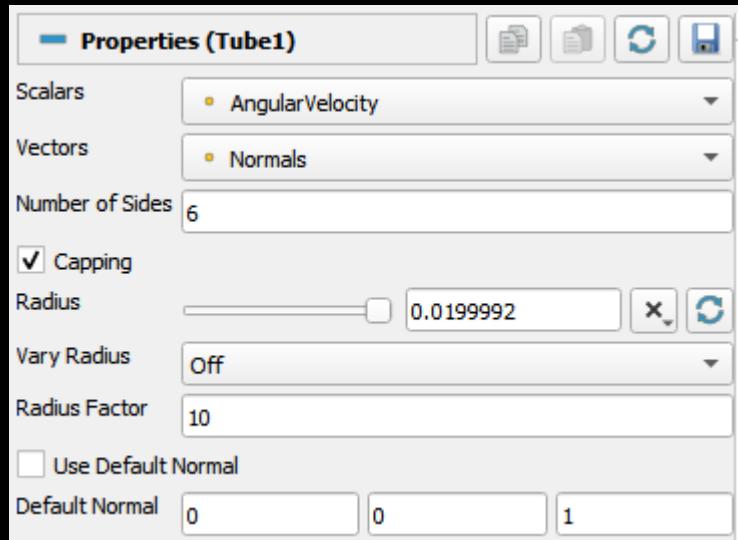
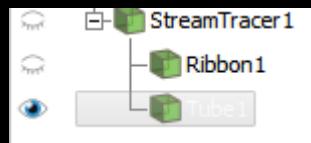
Visualise vectors as ribbons

Add a Ribbon filter to the StreamTracer filter



Visualise vectors as tubes

Add a Tube filter to the StreamTracer filter



A close-up photograph of a car's tachometer. The needle is positioned exactly halfway between the number 5 and the number 6 on the scale. The numbers 5 and 6 are illuminated in red, while the numbers 4, 7, and 8 are in white. The text "000 rpm" is visible at the bottom of the scale.

Demo time

Reset ParaView





Reading terrain data

Open the terrain.csv file

- File/Open...
- Find the terrain.csv file.
- Click Apply to load the file

The background image shows a serene mountain landscape. A deep blue lake occupies the foreground and middle ground, its surface perfectly still and reflecting the clear blue sky above. On either side of the lake are rugged, rocky mountains. The mountains on the left are steep and have distinct horizontal sedimentary rock layers, some showing signs of erosion. The mountains on the right are more craggy and sparsely vegetated. The lighting suggests it's either early morning or late afternoon, with the sun low on the horizon, casting long shadows and painting the sky in warm, golden hues.

Visualising terrain data

Create a 2D grid from terrain data

Pipeline Browser

- builtin:
- terrain.csv
- TableToStructuredGrid1**

Properties (TableToStructuredGrid1)

Whole Extent	0	511
	0	360
	0	0

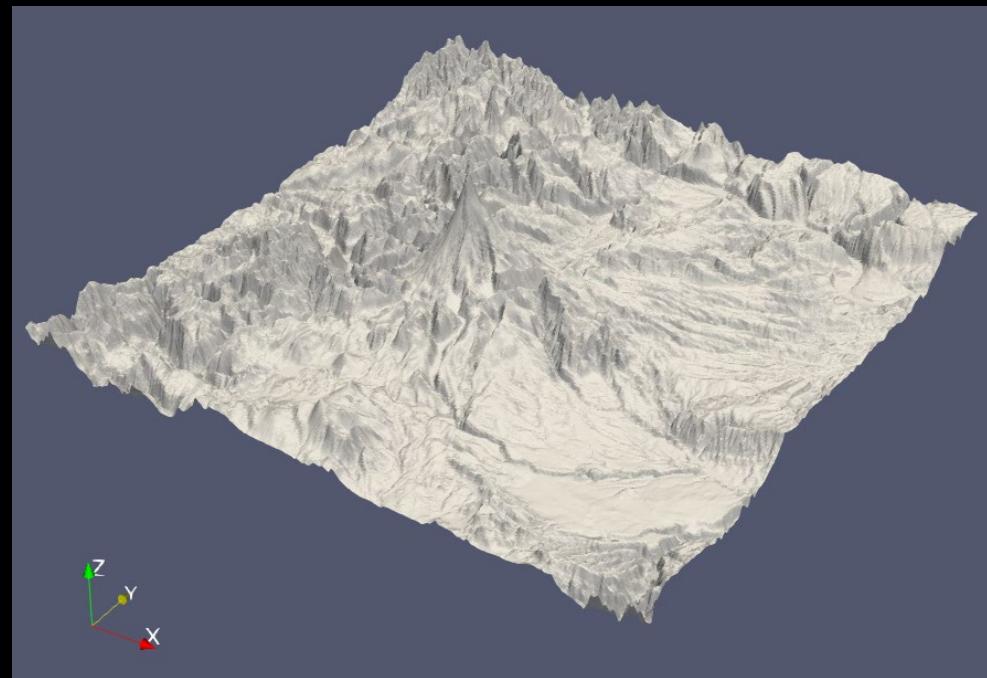
X Column: UTMx512

Y Column: UTMy361

Z Column: Z

Display (StructuredGridRepresentation)

Representation: Surface



Adjusting scale factor

Data is not square, so we need to adjust the scaling.

Add a Transform filter to the grid

Pipeline Browser

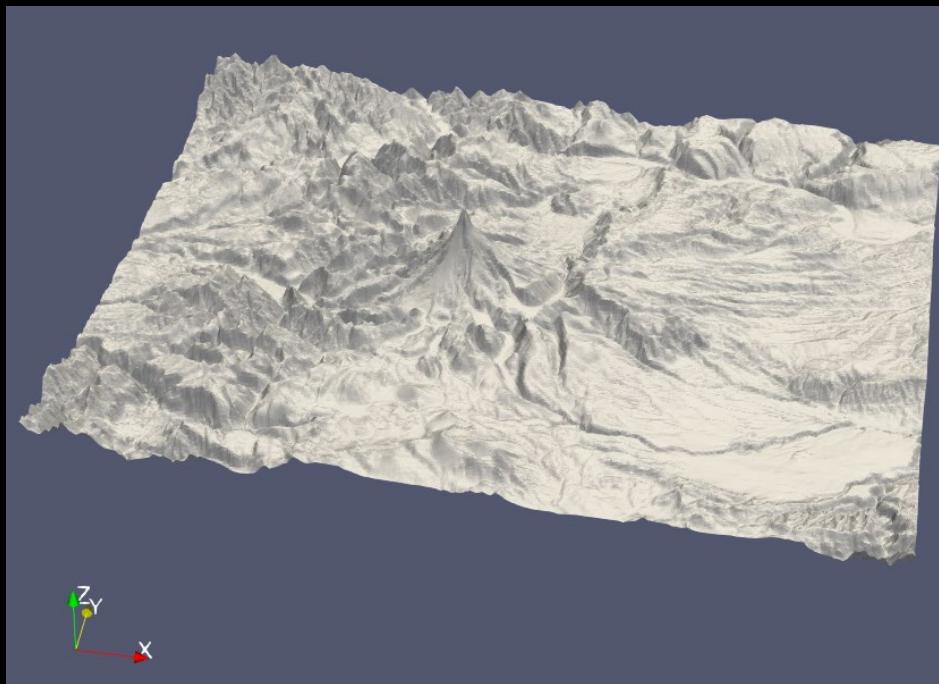
- builtin:
- terrain.csv
- TableToStructuredGrid1
- Transform1

Properties (Transform1)

Transform

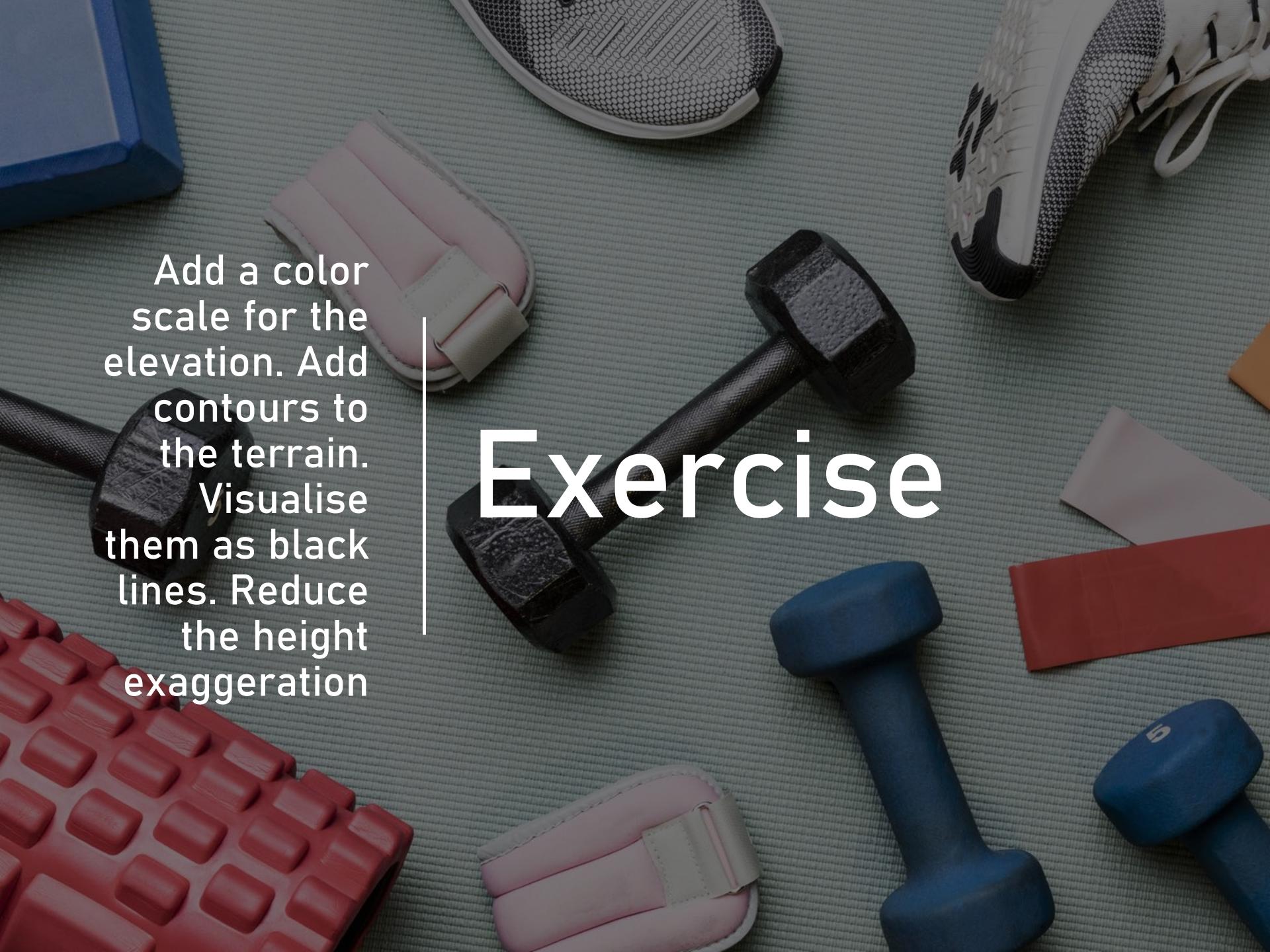
Show Box

Translate	0	0	0
Rotate	0	0	0
Scale	1.4194	1	1



A close-up photograph of a car's tachometer. The needle is positioned exactly halfway between the number 5 and the number 6 on the scale. The numbers 5 and 6 are illuminated in red, while the numbers 4, 7, and 8 are in white. The text "000 rpm" is visible at the bottom of the scale.

Demo time



Add a color scale for the elevation. Add contours to the terrain. Visualise them as black lines. Reduce the height exaggeration

Exercise

Getting data into ParaView

Using PyVTK to read data into ParaView



Getting data into ParaView

- ParaView supports a large amount of data formats.
- In some cases you need to create your own export functions
- The default fileformat format for ParaView is .vtk
- .vtk-files are text files that can represent all data structures supported by VTK.
- Complicated to create these manually
- The PyVTK library can be used to simplify the creation of .vtk-files.

PyVTK Notebook

CO Visualisation with PyVTK ★

Arkiv Redigera Vy Infoga Körning Verktyg Hjälp

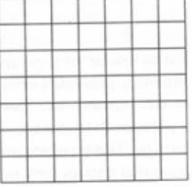
+ Kod + Text Anslut ▾ | Redigera ▾

PyVTK

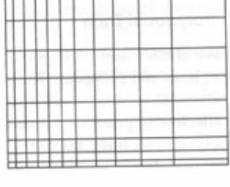
{x} PyVTK is a Python library for creating files for use with ParaView or VTK toolkit.

VTK supports the following data types:

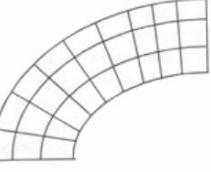
(a) Image Data



(b) Rectilinear Grid



(c) Structured Grid



(d) Unstructured Points

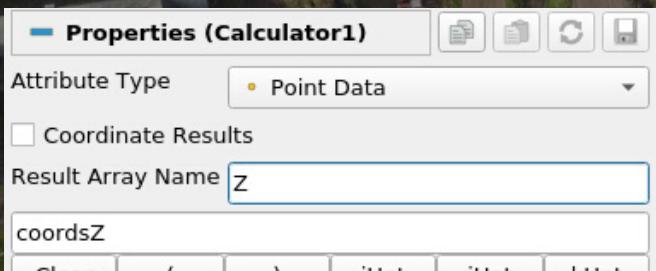


The background of the slide features a dynamic, radial pattern of light rays emanating from the bottom right corner. The rays are composed of numerous thin, horizontal bands of light in shades of blue, purple, and red, creating a sense of motion and depth. The overall effect is reminiscent of a starburst or a high-energy particle collision.

Exercises with real data

Photogrammetry

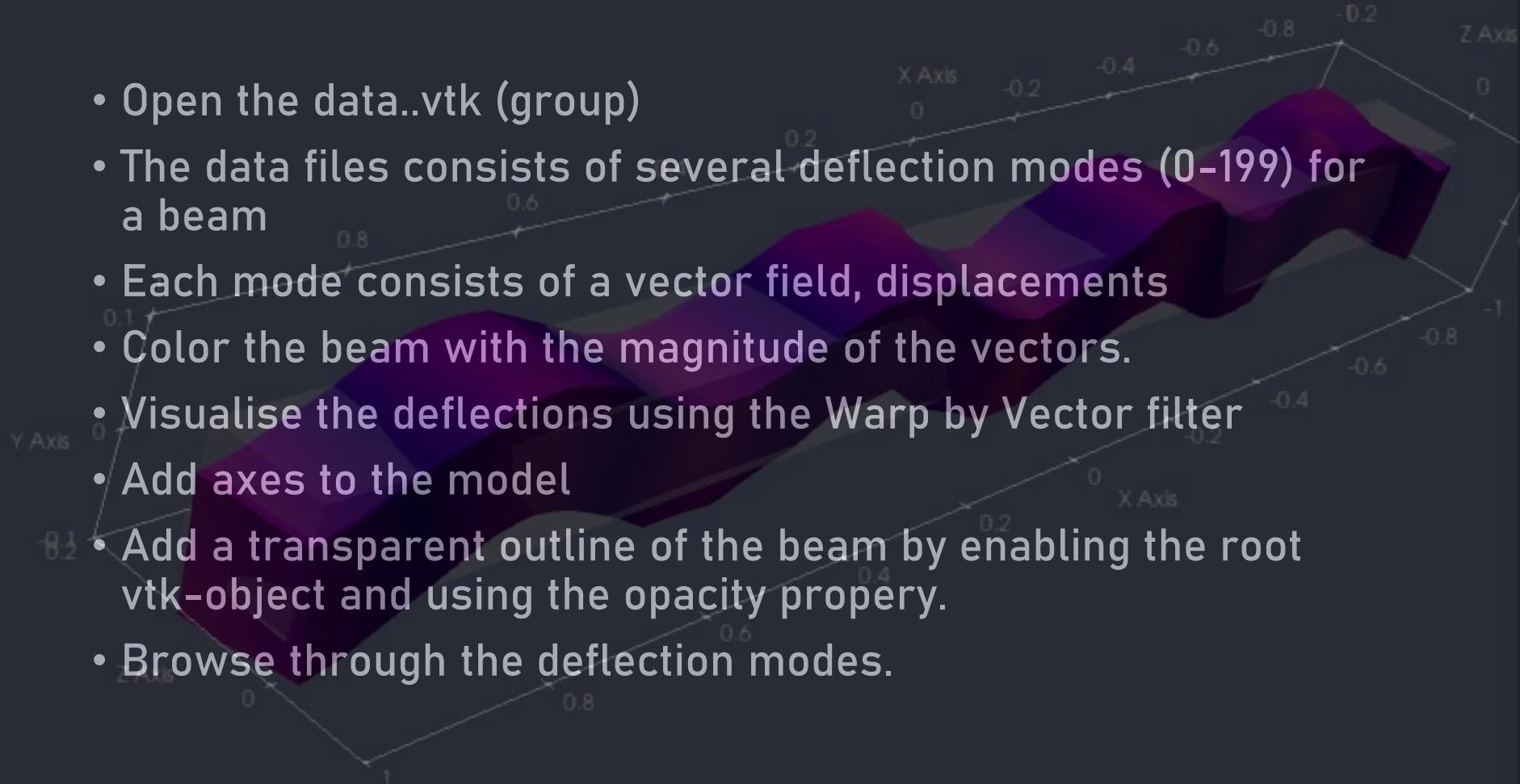
- Open the `skivar.ply` file.
- Use the Calculator filter to create a scalar field, Z, with elevation data.



- Visualise elevation data using a suitable color scale
- Add black contour lines with a interval of 1 m

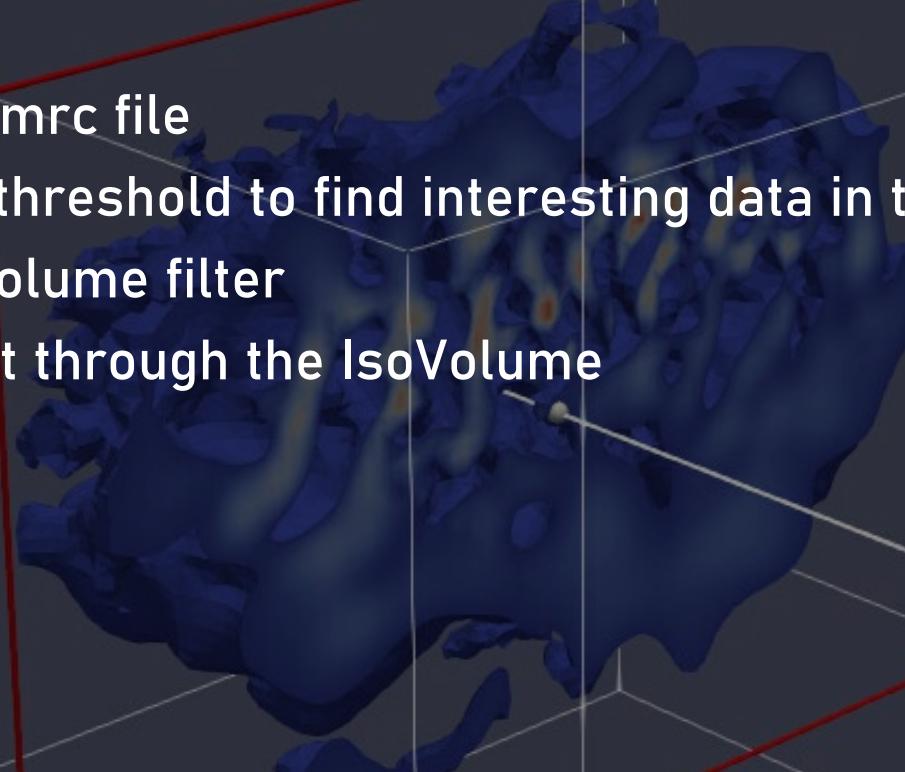
Structural Mechanics

- Open the data..vtk (group)
- The data files consists of several deflection modes (0-199) for a beam
- Each mode consists of a vector field, displacements
- Color the beam with the magnitude of the vectors.
- Visualise the deflections using the Warp by Vector filter
- Add axes to the model
- Add a transparent outline of the beam by enabling the root vtk-object and using the opacity property.
- Browse through the deflection modes.



Cryo-EM

- Open the .mrc file
- Try using threshold to find interesting data in the data set.
- Try a IsoVolume filter
- Make a cut through the IsoVolume



CFD

- Open the pipe.vtk file. Note this is a large data set and can take some time to load.
- Visualise the flow field U
 - Using Glyphs
 - Stream Tracer
 - Stream Trace with Tubes.



Starting
ParaView
on Aurora



LUNARC Applications
Lunarc Applications On-Demand
LUNARC Support
LUNARC Tools
LUNARC Windows Desktop On-Demand
Disconnect Lunarc remote visualization session
Logout Lunarc Remote Visualization Session

Old Firefox Data



Trash



Lunarc Documentation



Windows Desktop Session



Abaqus V6R2017x



ParaView 5.8.0

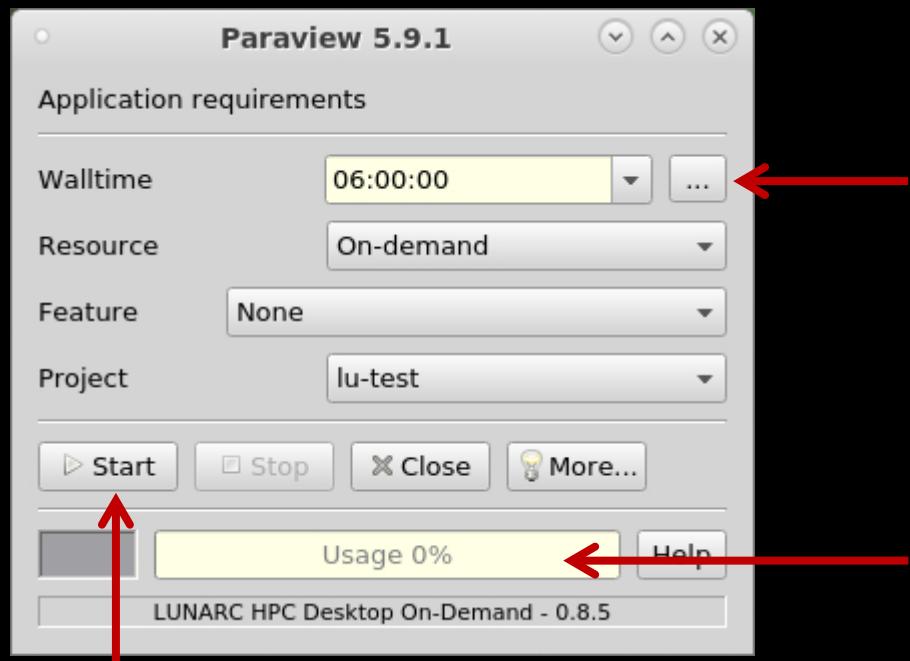


Terminal

- 3D Modeling
 - CAE
 - Chemistry
 - Comsol
 - Data analysis
 - Development
 - Jupyter
 - Mathematica
 - Matlab
 - Medical Imaging
- Post Processing
 - LS PrePost 4.3
 - Ovito 3.4.1
 - ParaView 5.1.0 - OpenFOAM
 - ParaView 5.1.0 - OpenFOAM[32;101;31M]
 - ParaView 5.1.1
 - ParaView 5.4.1
 - ParaView 5.5.0 (swr)
 - ParaView 5.8.0
 - ParaView 5.9.1
- Volume Rendering
 - Scionp

VMD 1.9.2

ParaView from LUNARC On-Demand

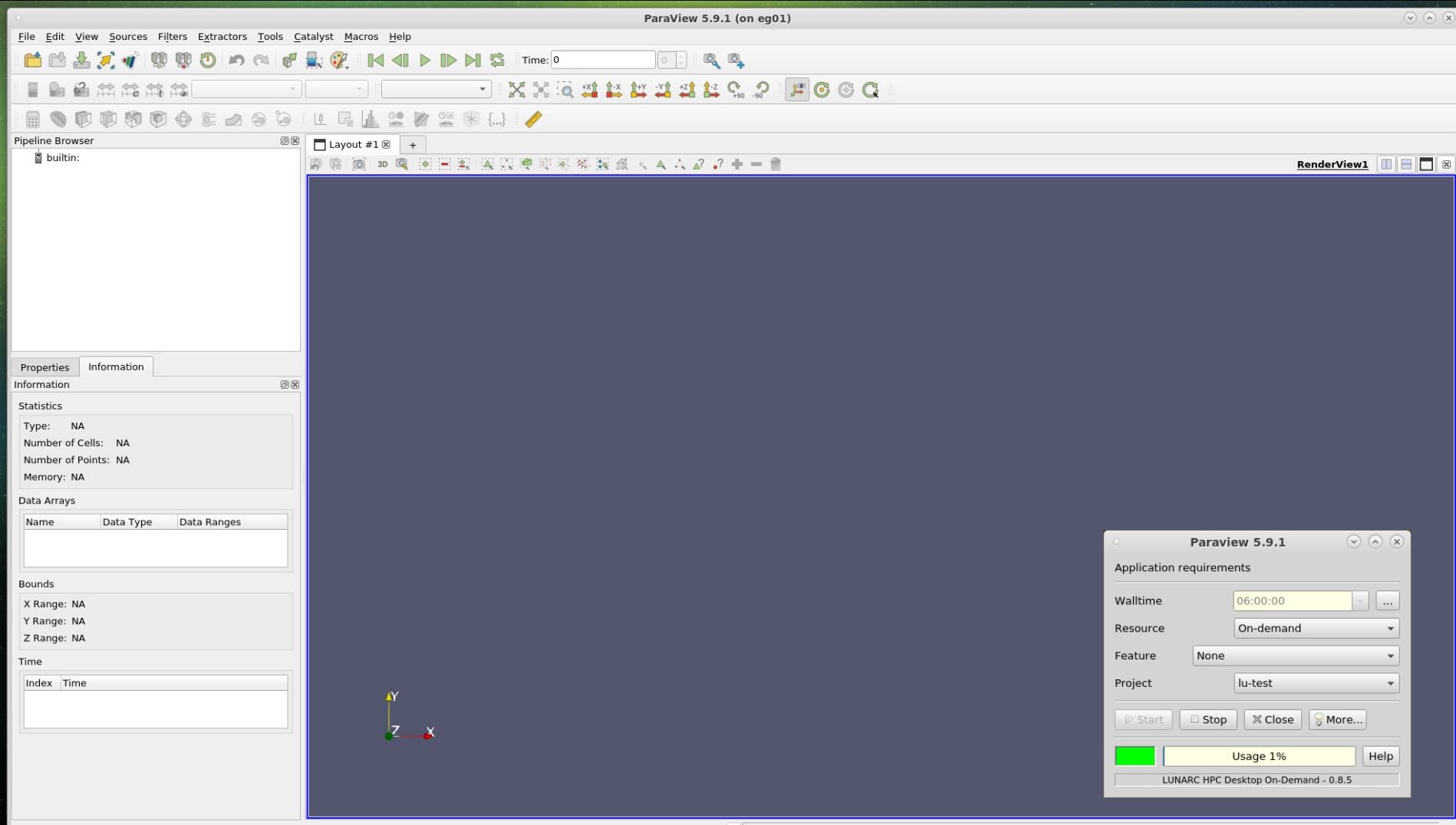


Start ParaView here

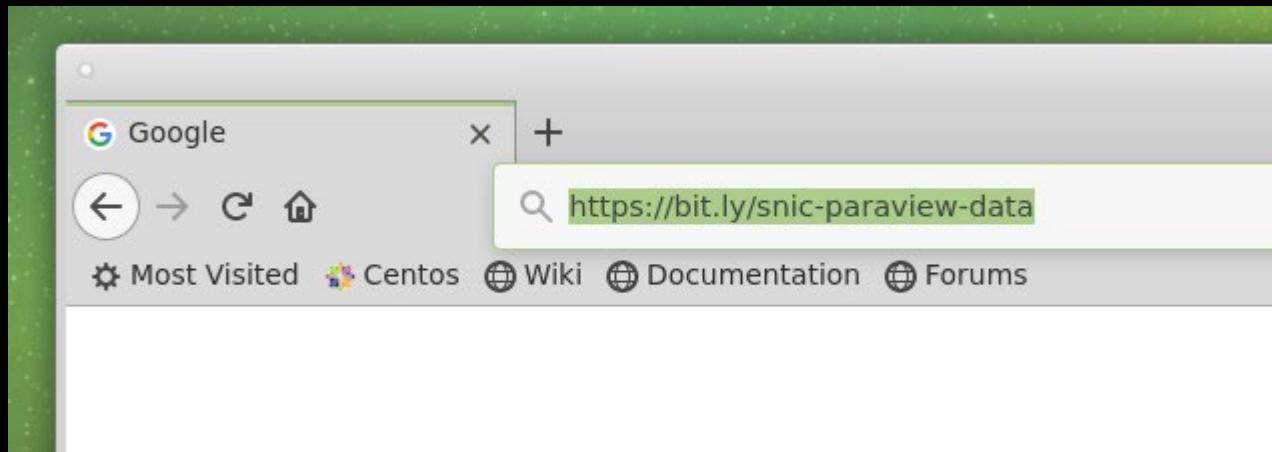
Give an estimate of how long you will need ParaView

Shows how much of the allocation have been used.

ParaView running on Aurora



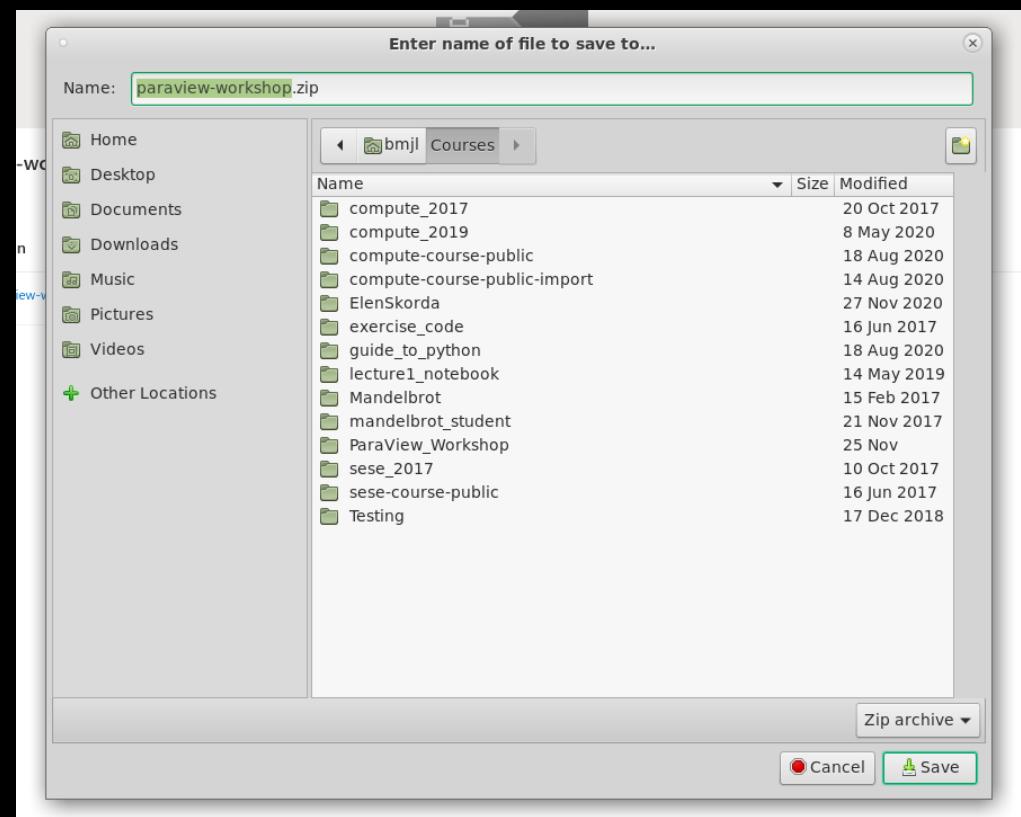
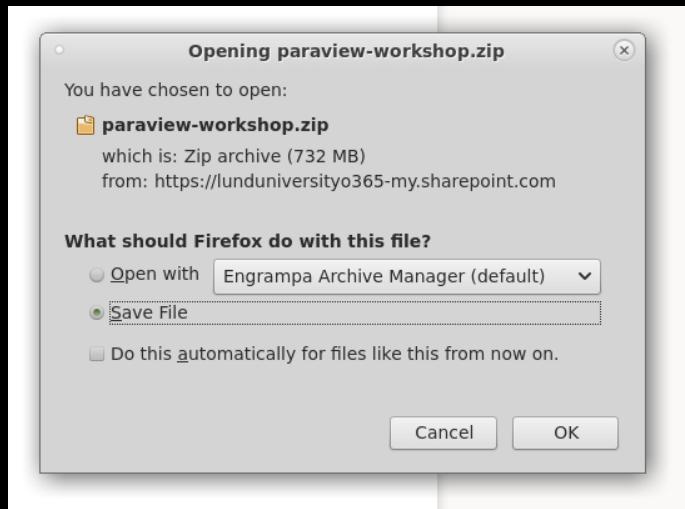
Downloading data to Aurora



A screenshot of a web browser window titled "Shared - OneDrive - Mozilla Firefox". The address bar shows the URL https://lunduniversityo365-my.sharepoint.com/personal/bygg-jli_lu_se/_layouts/. Below the address bar, there is a red arrow pointing to a blue download icon followed by the text "Ladda ned". The main content area displays a file named "paraview-workshop.zip". A table below the file name lists its details:

Namn	Ändringsdatum	Filstorlek
paraview-workshop	2021-12-14	

Downloading data to Aurora



Extracting data set

