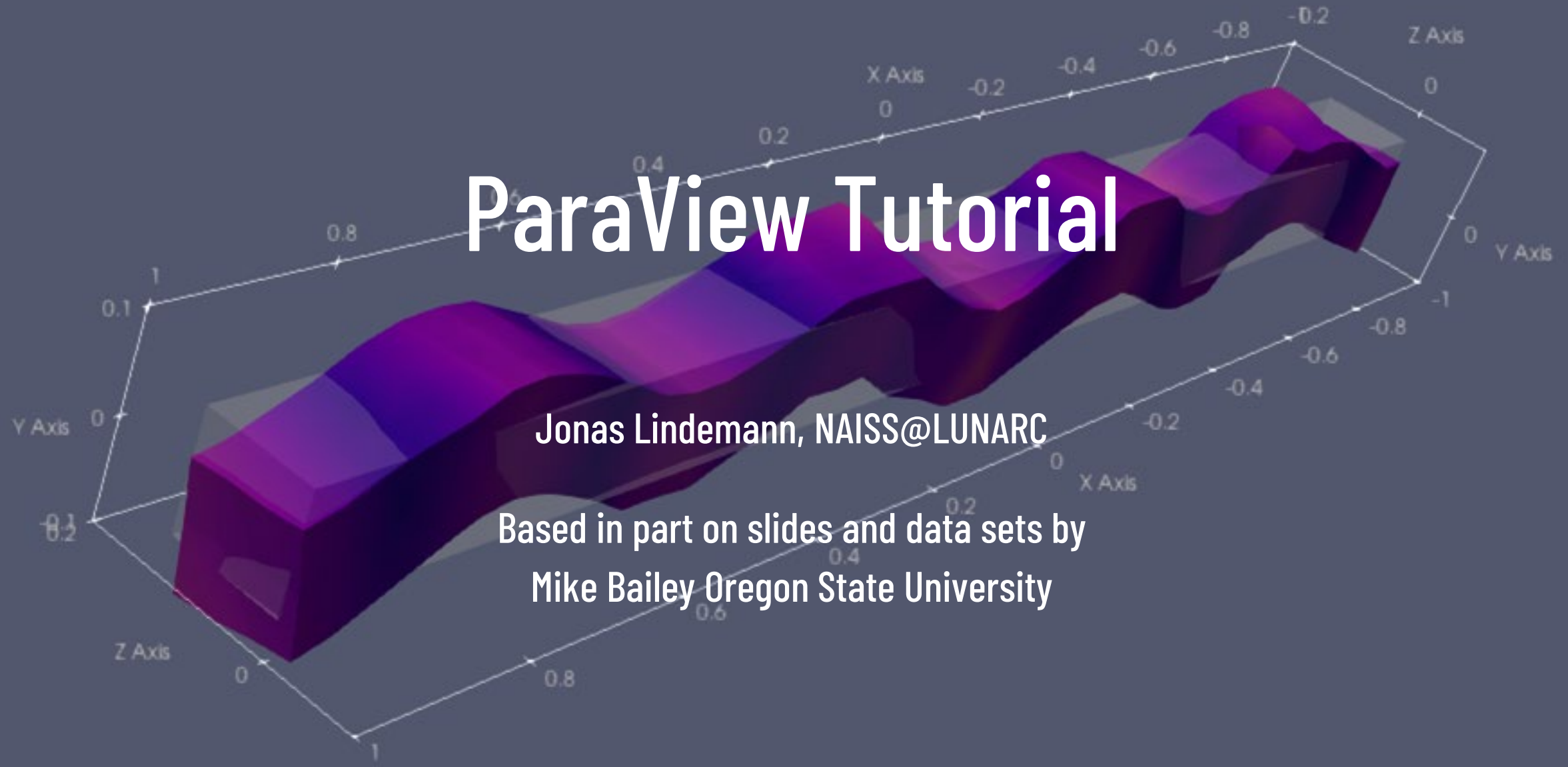
ParaView Tutorial

Jonas Lindemann, NAISS@LUNARC

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



[Mike Bailey's ParaView Page (oregonstate.edu)]

InfraVis ParaView Tutorial

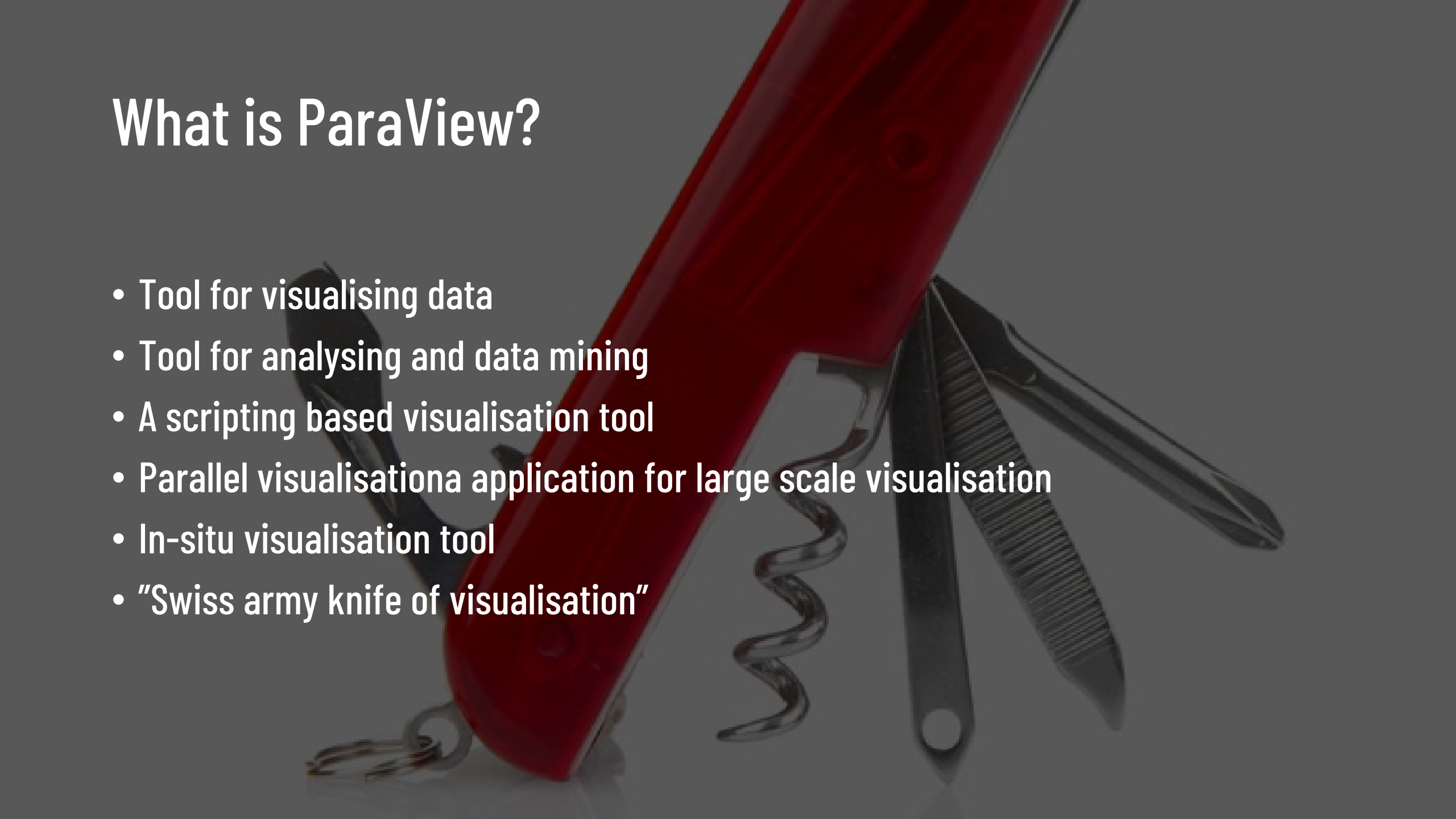
PLEASE NOTE

- 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

BREAK - Downloads and data files

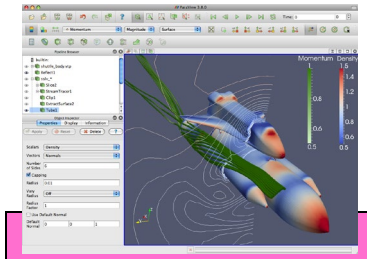
- Data files are available here:
  - https://github.com/jonaslindemann/paraview-workshop
- ParaView can be downloaded here
  - https://www.paraview.org/download/
  - Download the stable version, currently 5.13

What is ParaView?



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

ParaView Application Architecture



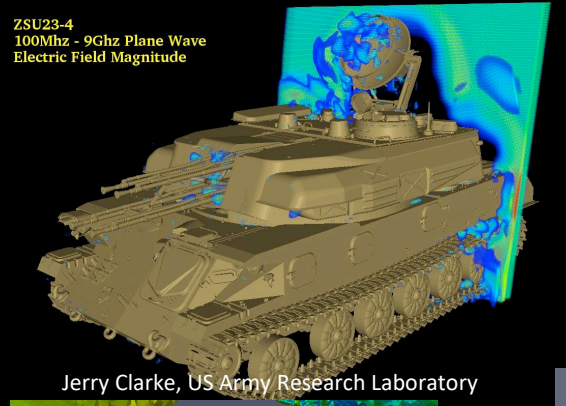
ParaView Client	pvpython	ParaWeb	Catalyst	Custom App
UI (Qt Widgets, Python Wrappings)				
ParaView Server				
VTK				
OpenGL	MPI	IceT	Etc.	

Kitware

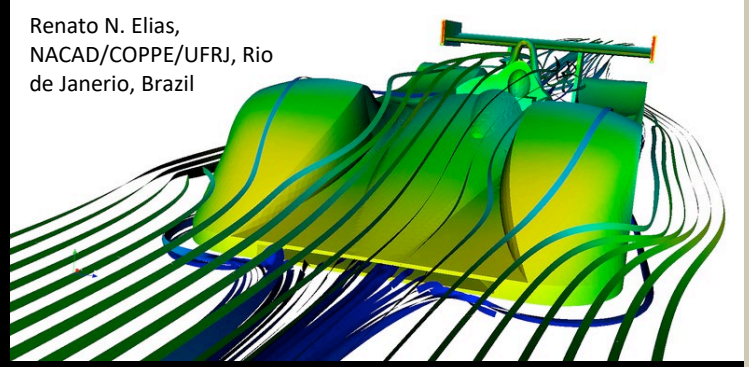
Sandia National Laboratories

ParaView Development

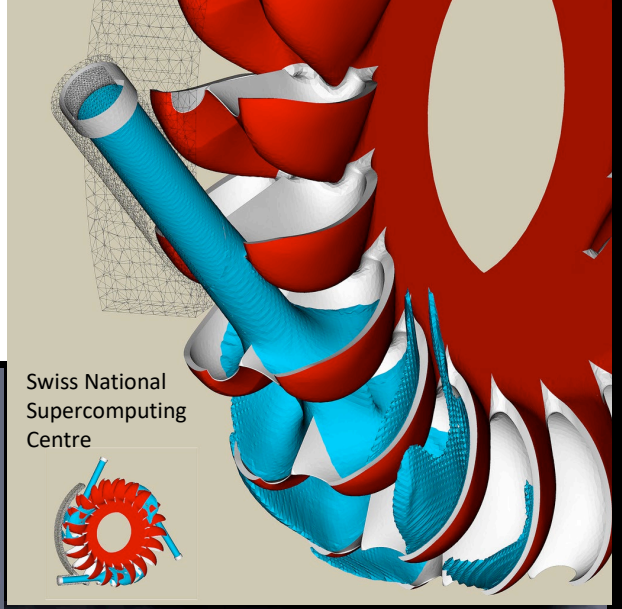
- Started in 2000 as collaborative effort between Los Alamos National Laboratories and Kitware Inc. Sandia has been a major contributor since 2005.
  - ParaView 0.6 released October 2002.
- Paraview 3.0 release in May 2007.
  - GUI rewritten to be more user friendly and powerful.
- ParaView 4.0 released 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.



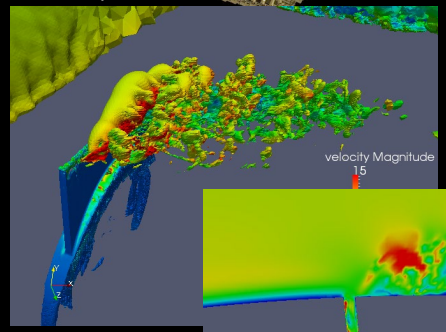
Jerry Clarke, US Army Research Laboratory

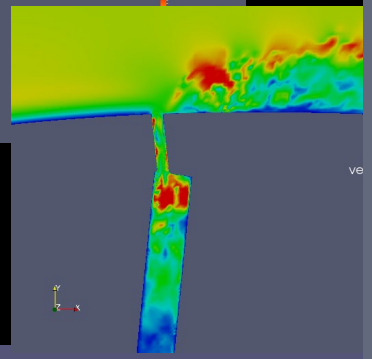


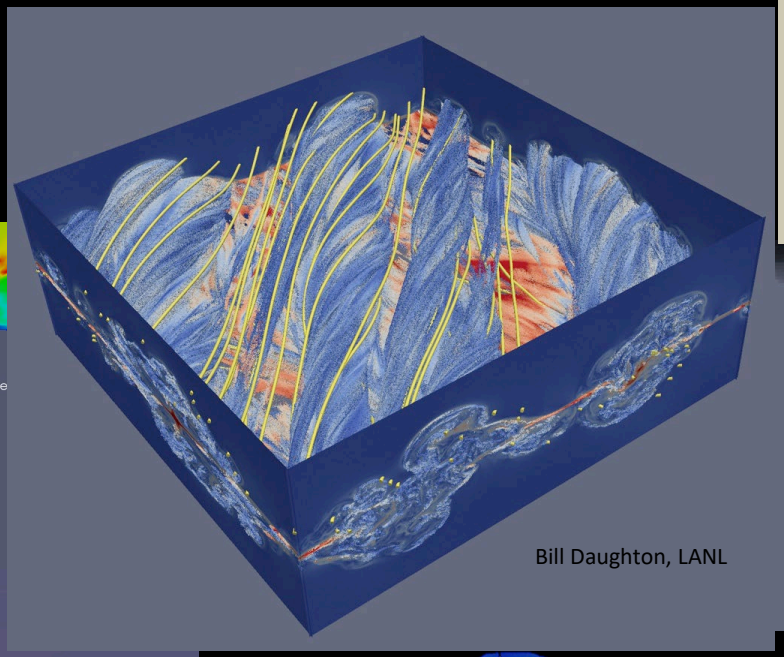
Renato N. Elias, NACAD/COPPE/UFRJ, Rio de Janerio, Brazil



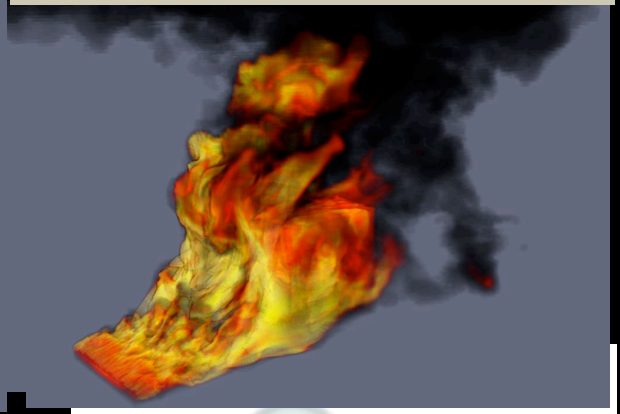
Swiss National Supercomputing Centre

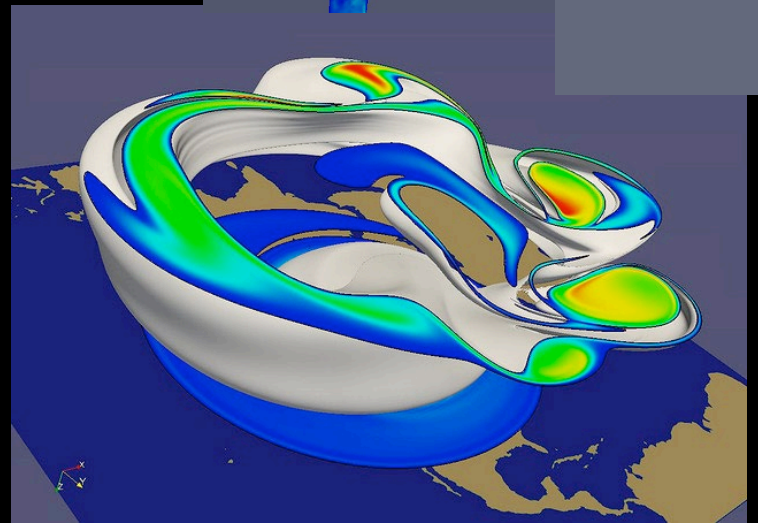


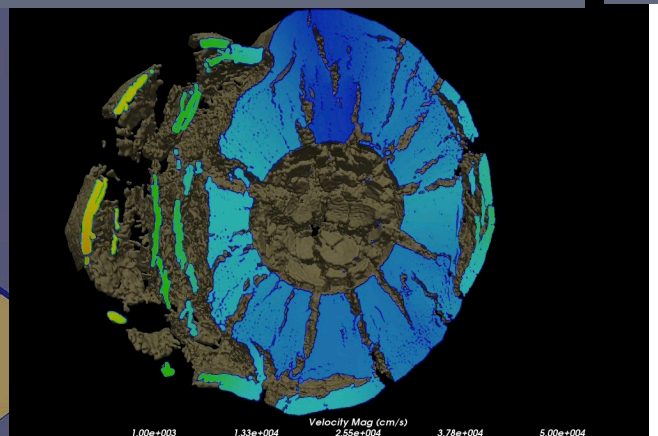


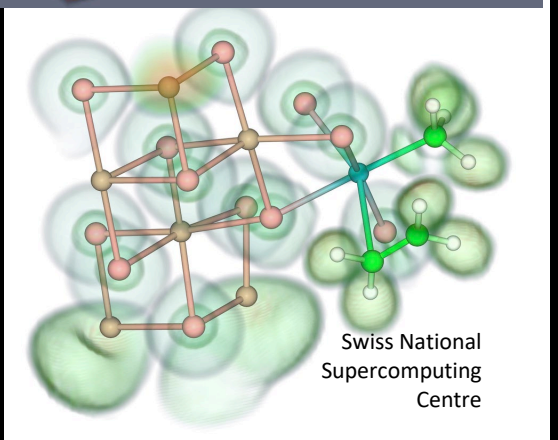


Bill Daughton, LANL





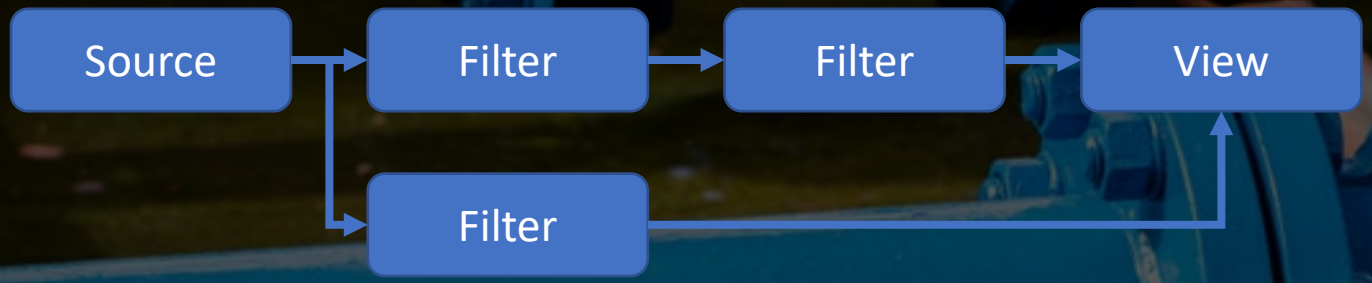




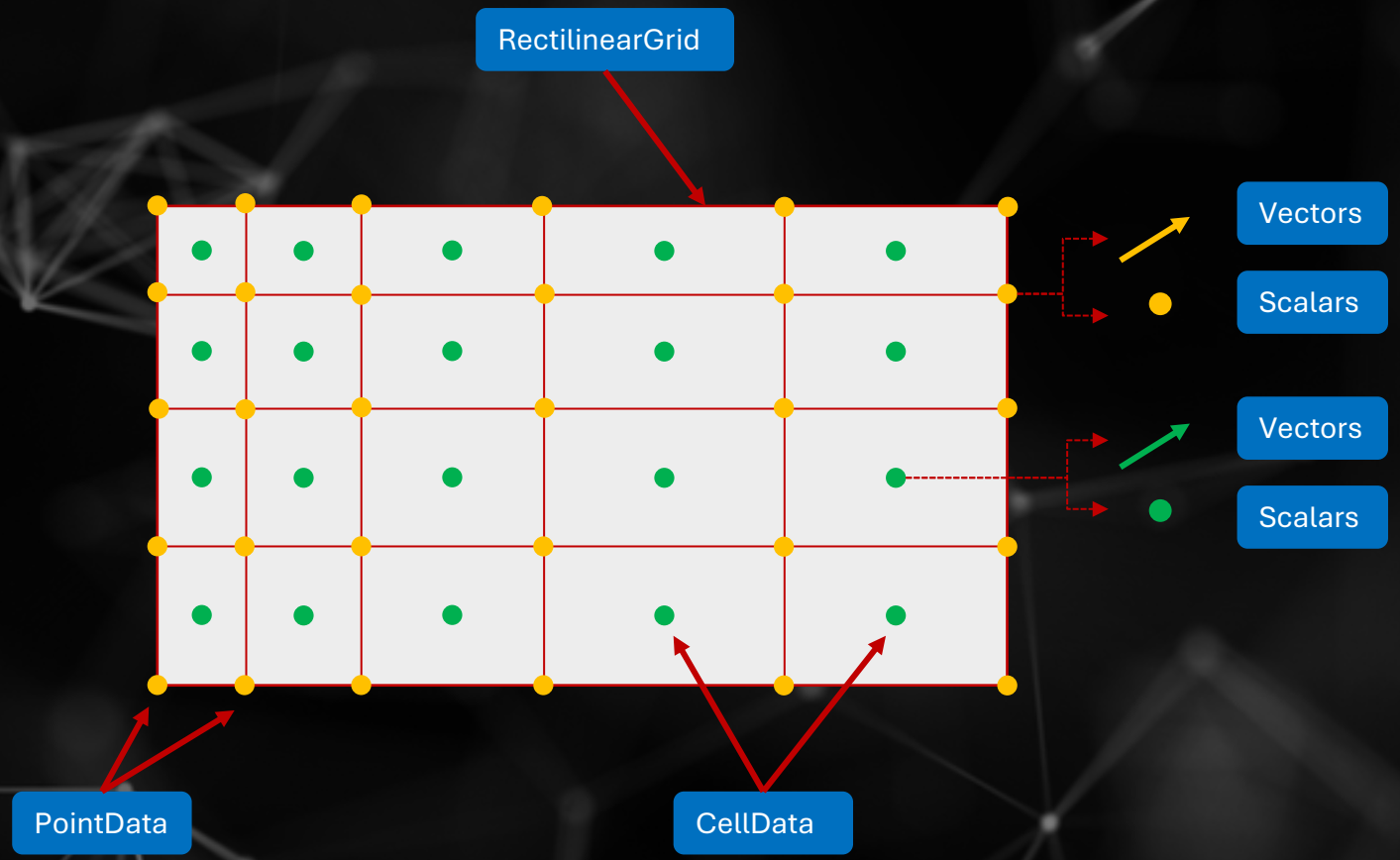
Swiss National Supercomputing Centre

Basic concept

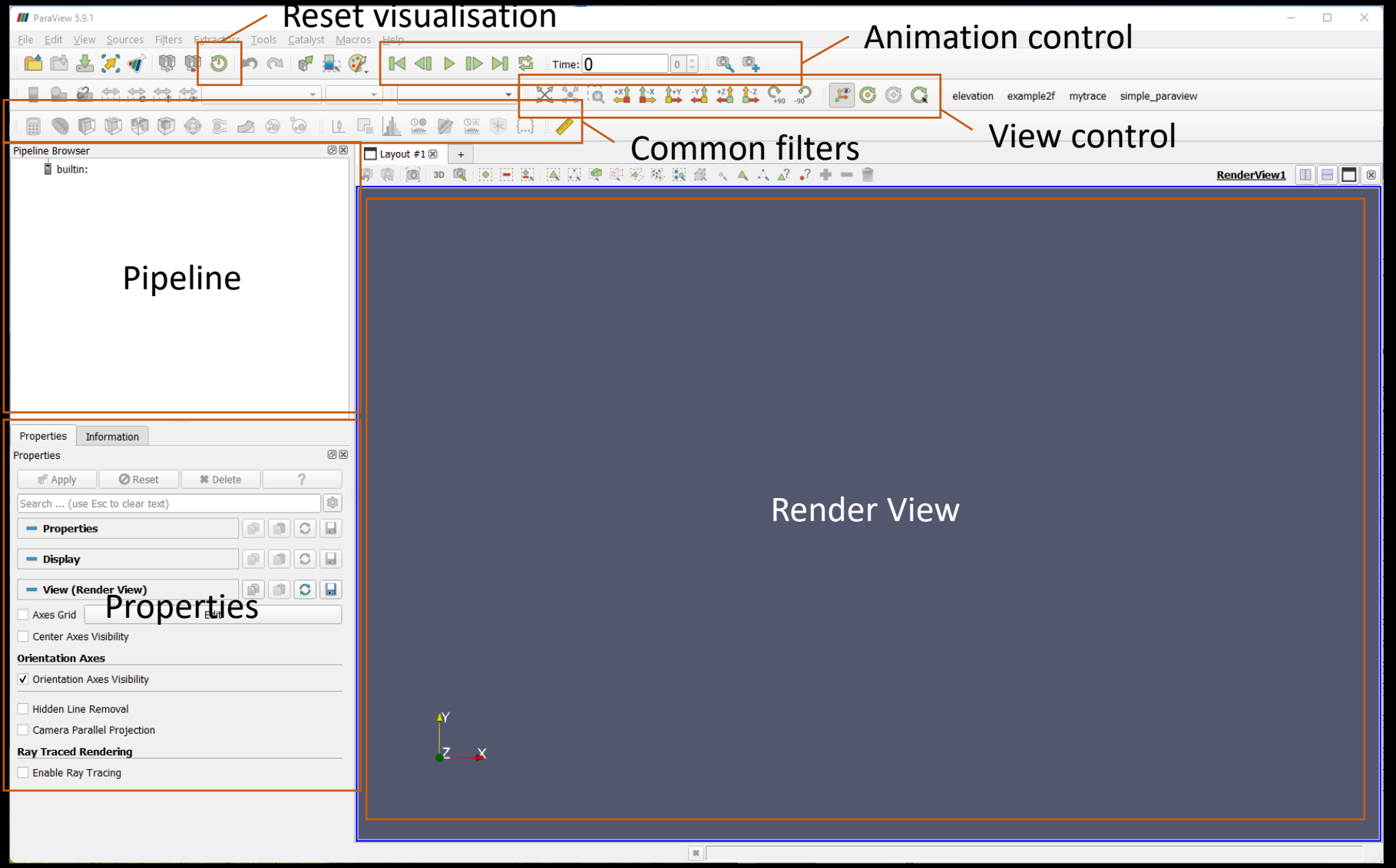
- Visualisation application developmed by KitWare
- Uses the Visualisation ToolKit (VTK) as foundation
- Uses a flow concept for generating visualisations



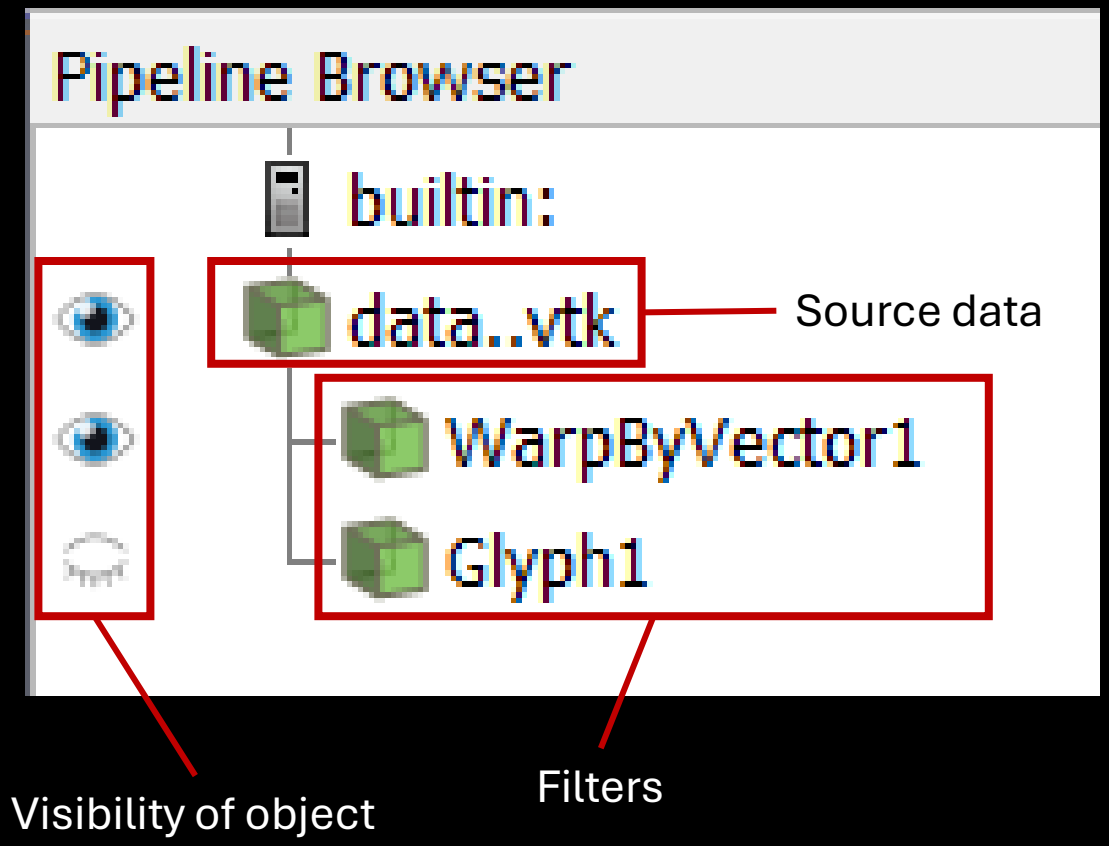
Data structures and data – High level



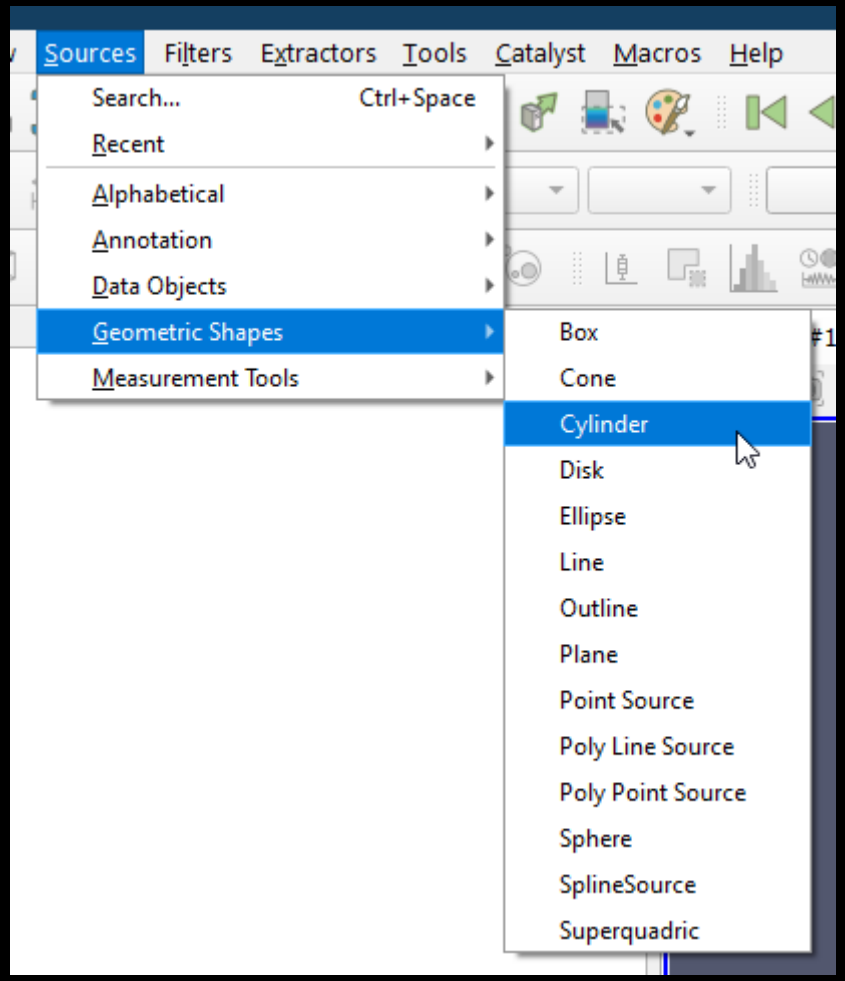
ParaView user interface



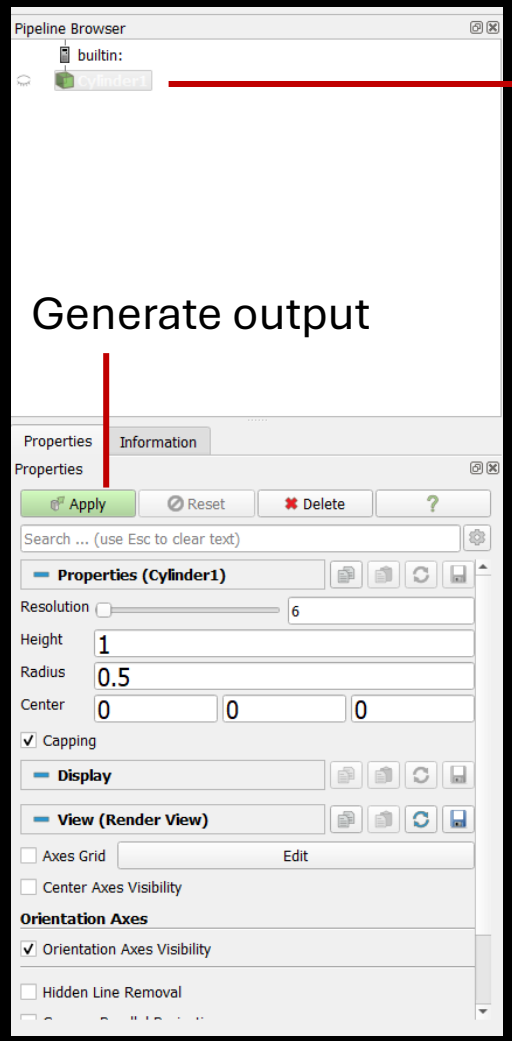
Flow in ParaView



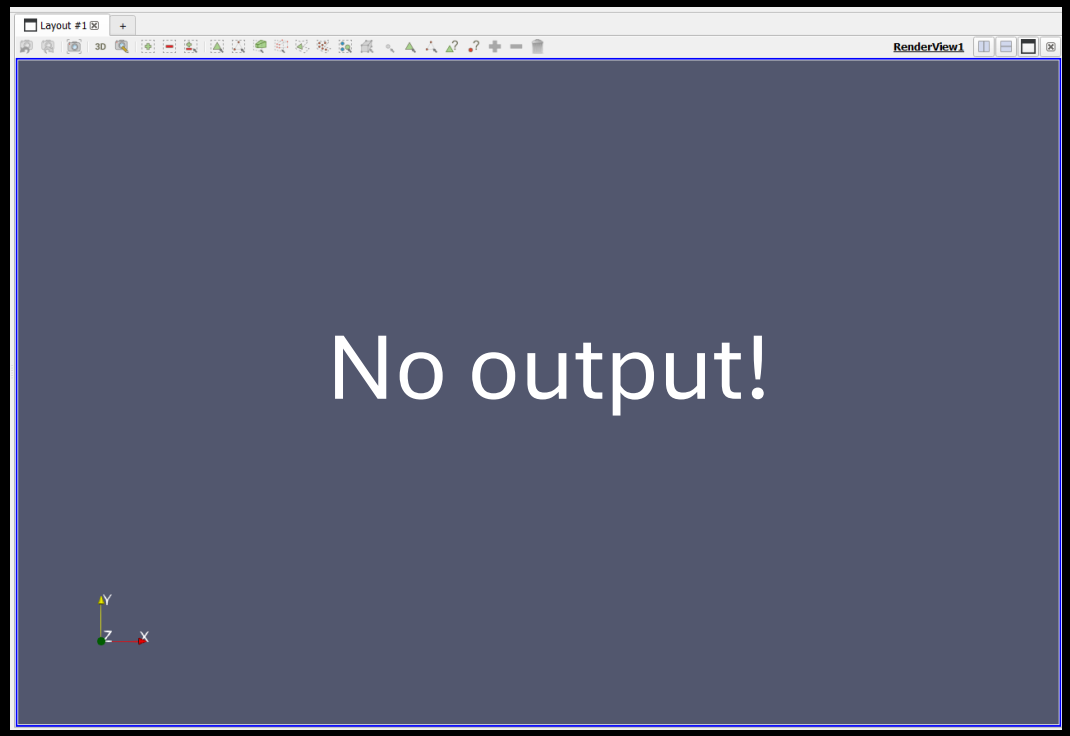
Basic usage – Creating a source



Creating a Cylinder

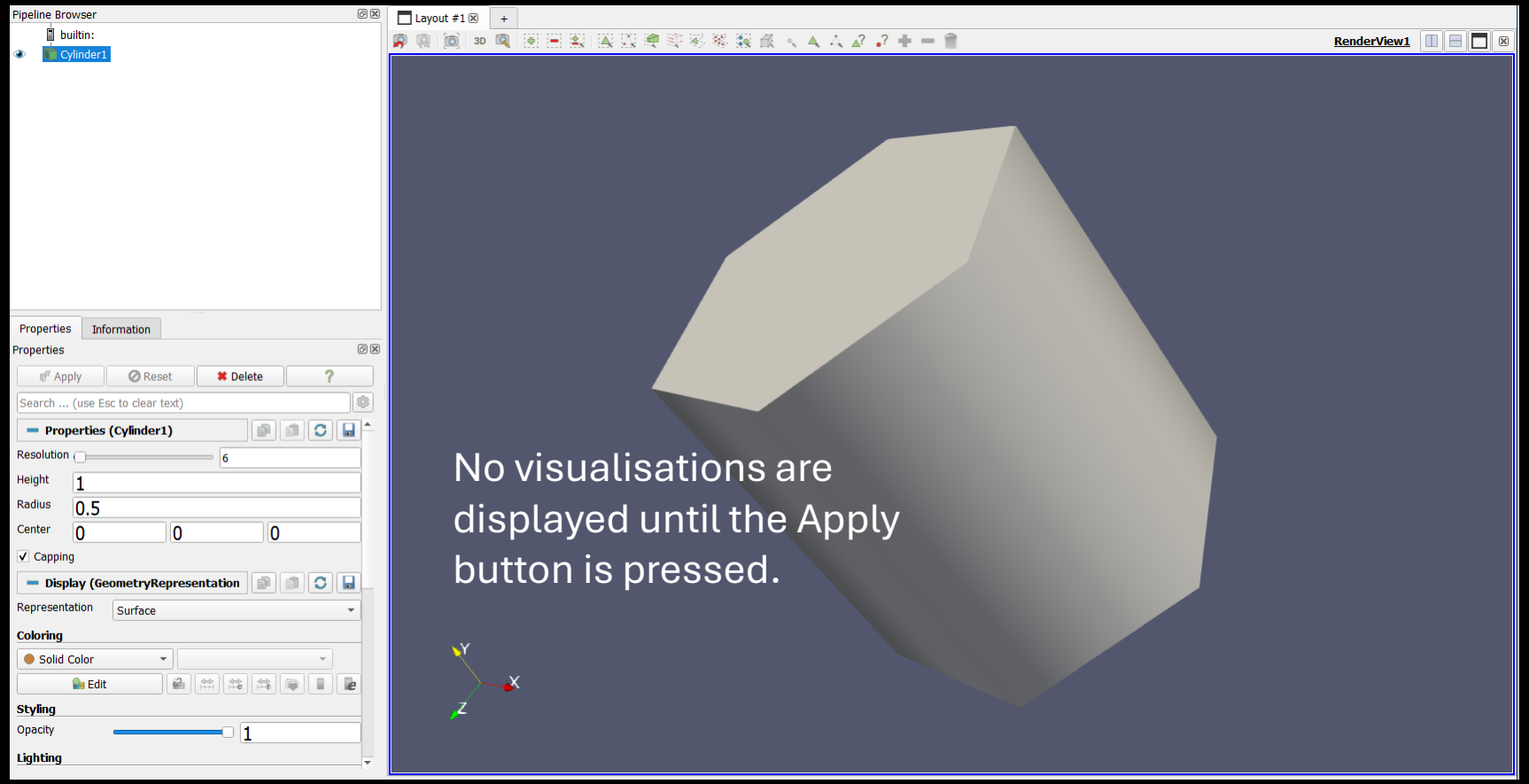


Cylinder source

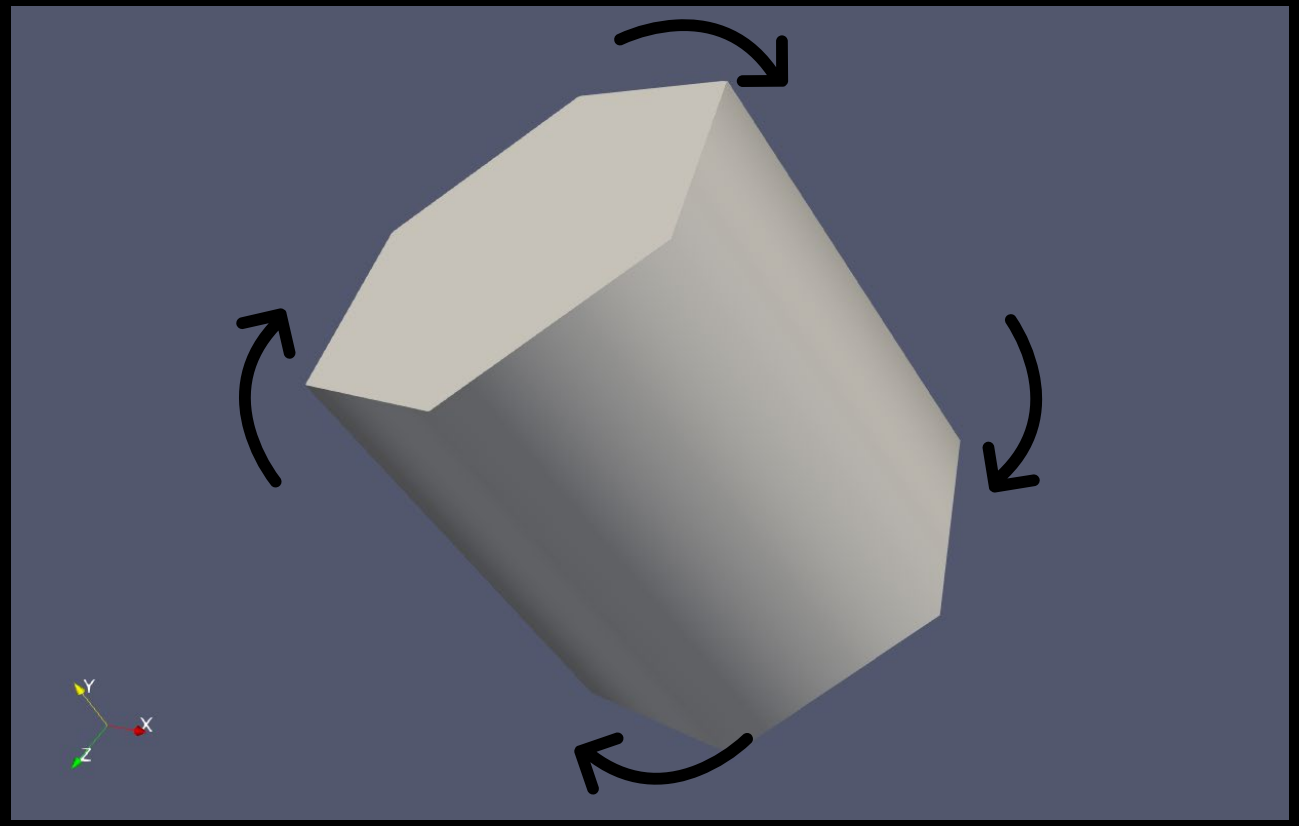


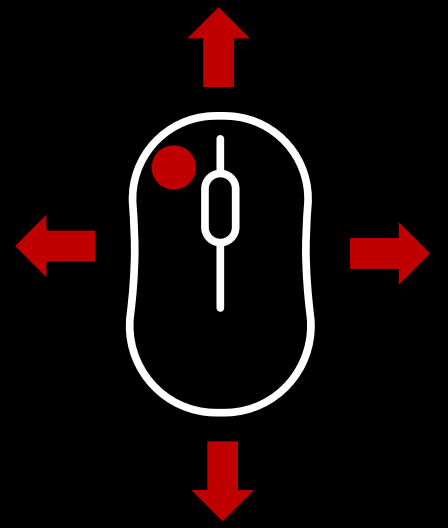
Creating a Cylinder

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



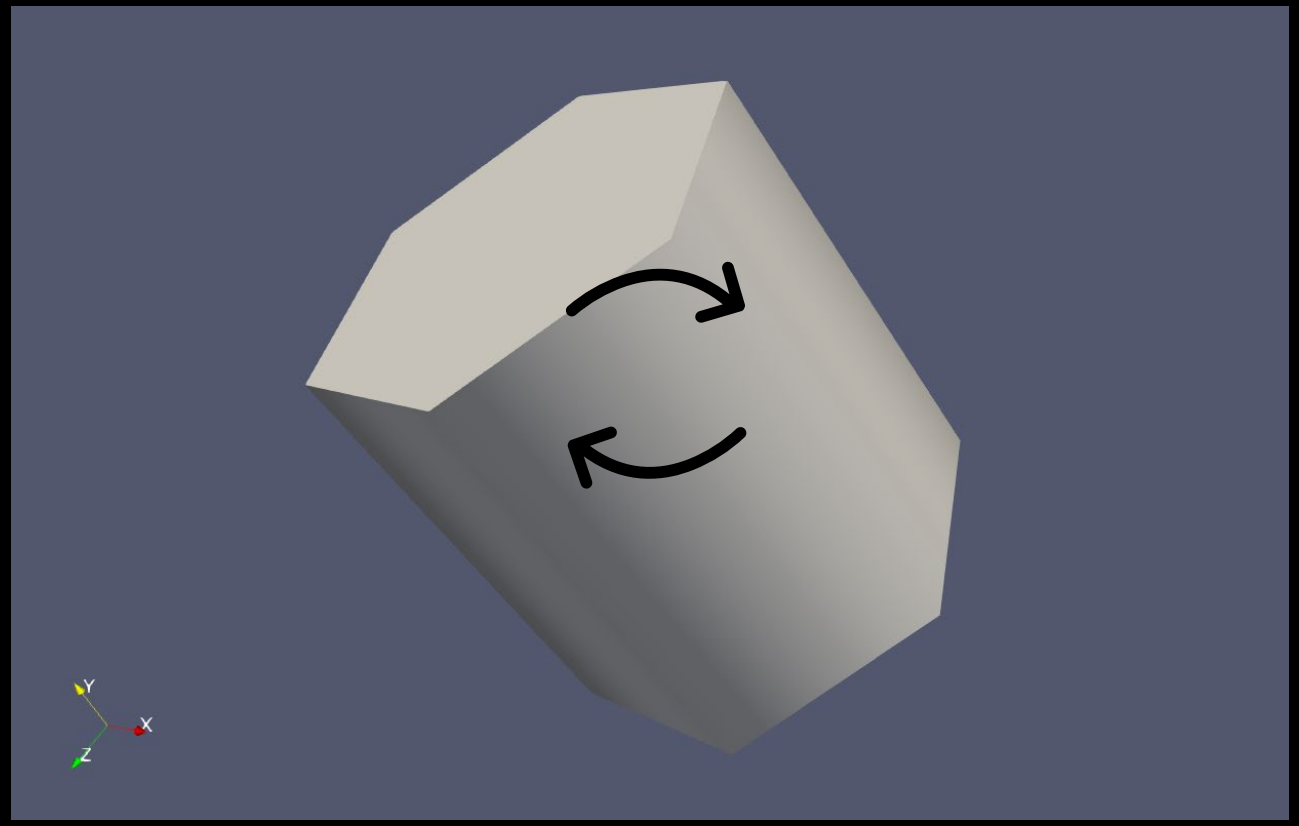
3D view manipulation



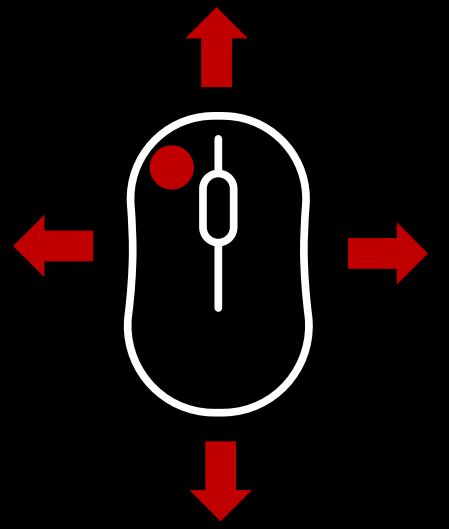


Rotate view

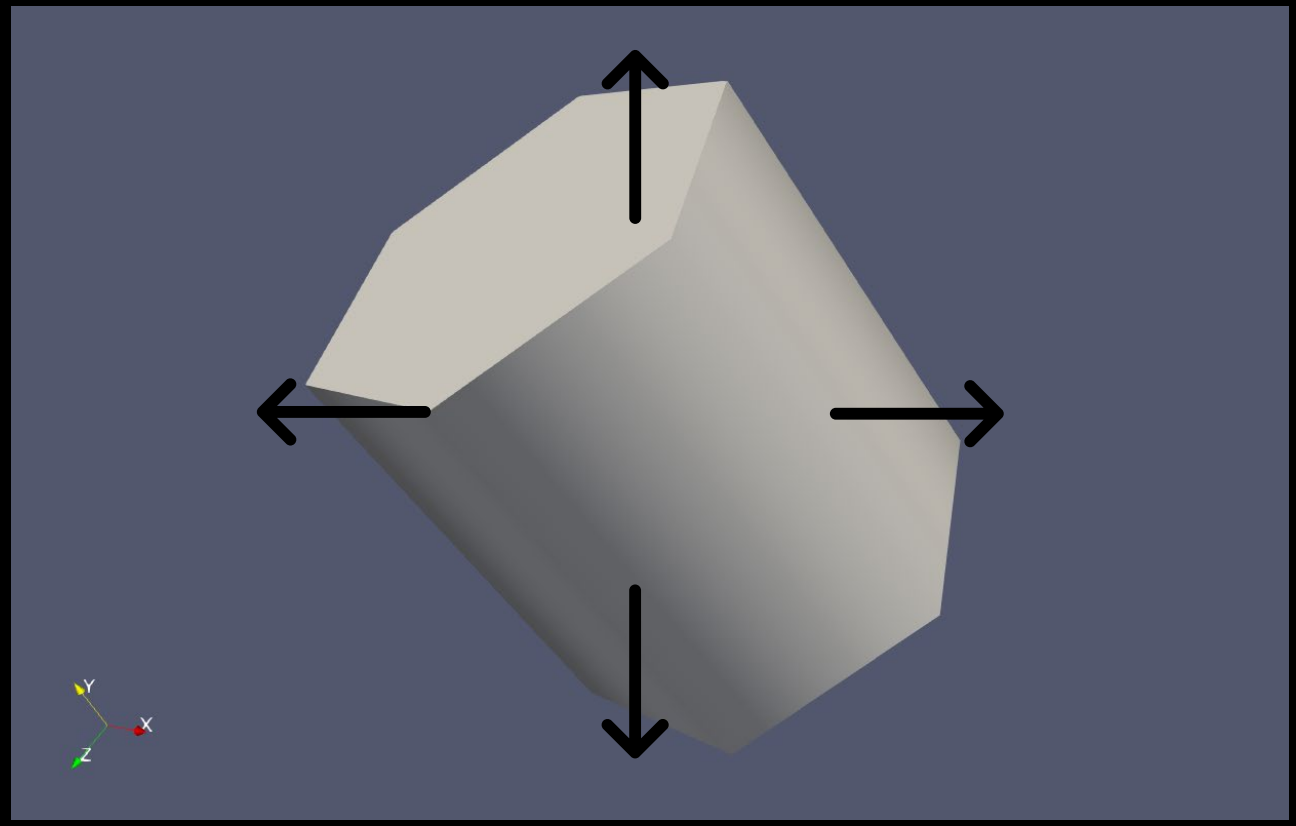
3D view manipulation

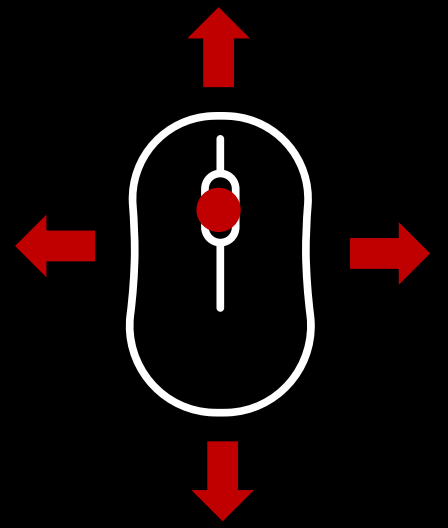


Roll view



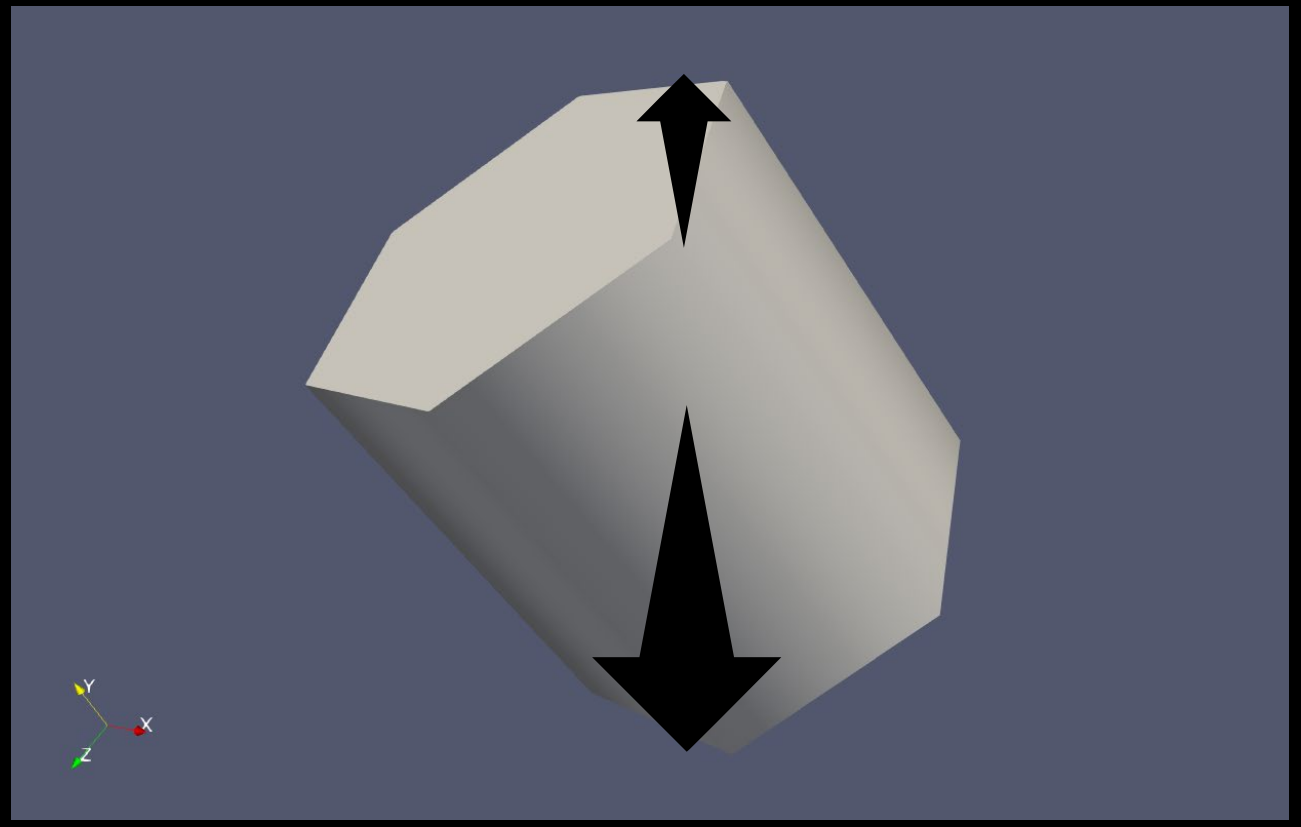
3D view manipulation

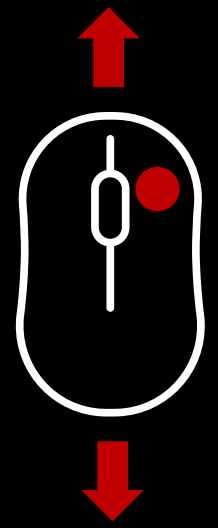




Pan

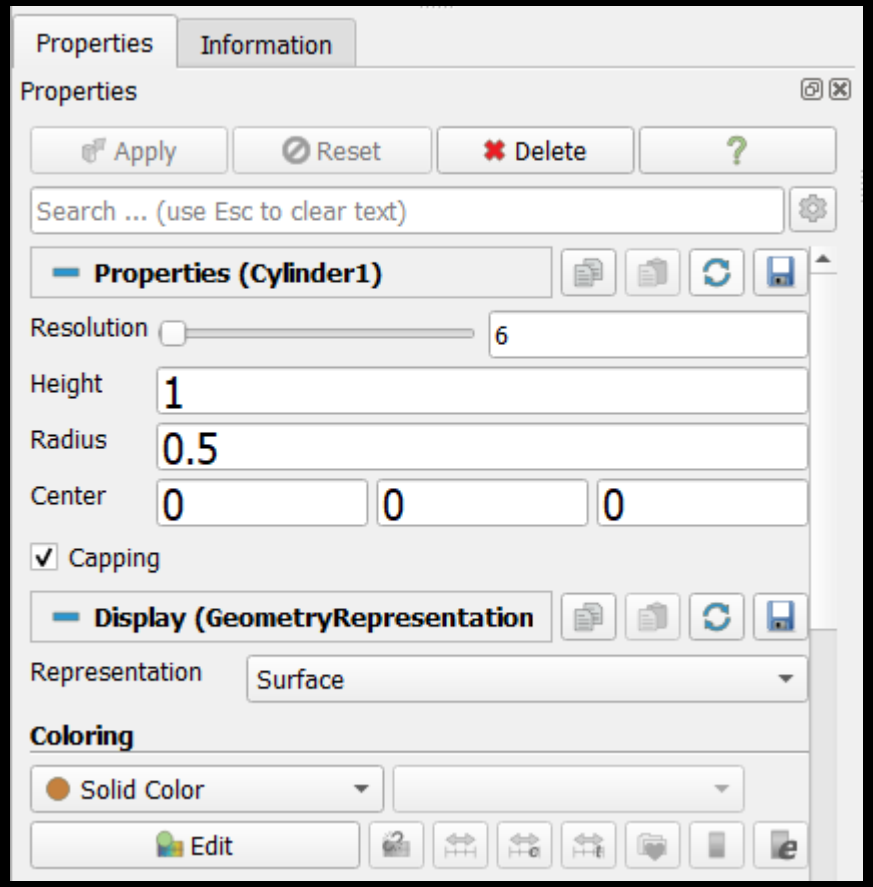
3D view manipulation





Zoom in / Zoom out

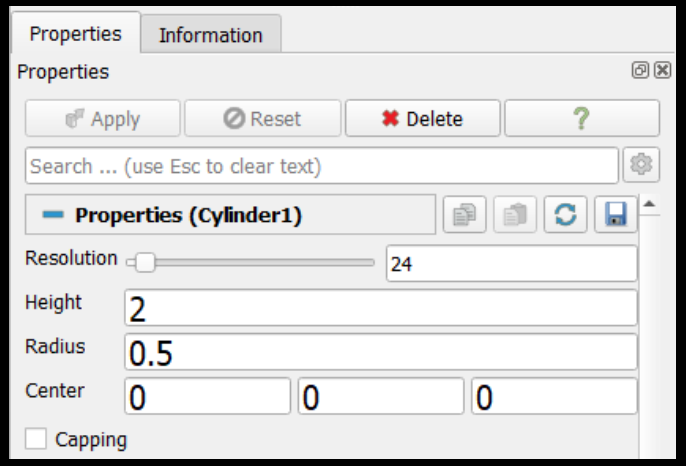
Specific object properties

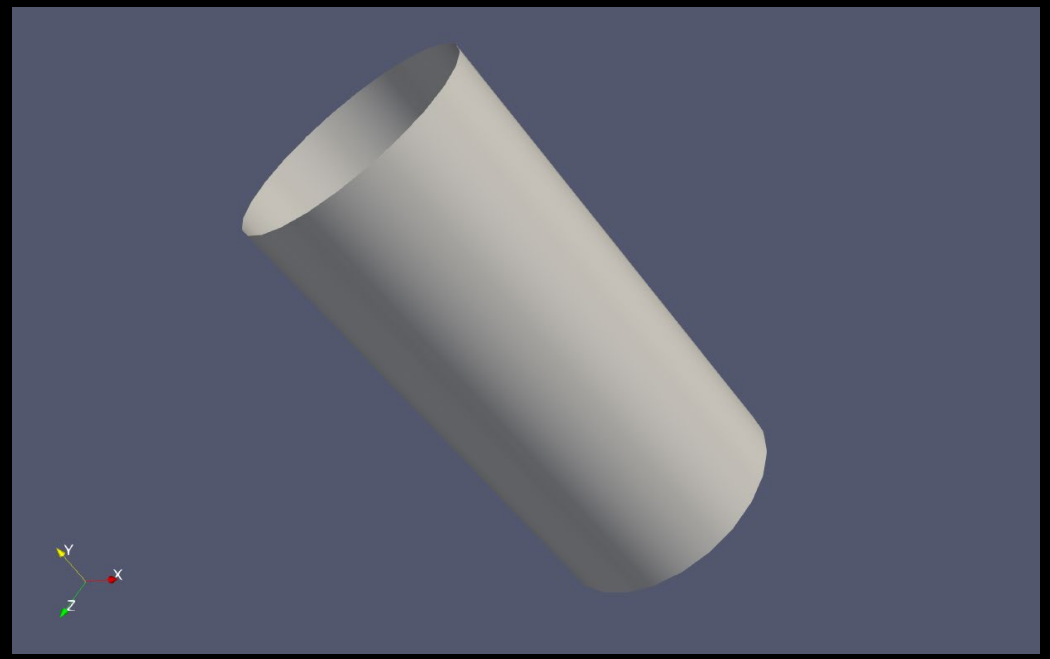


All objects in ParaView have properties that can be changed in the property inspector.

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

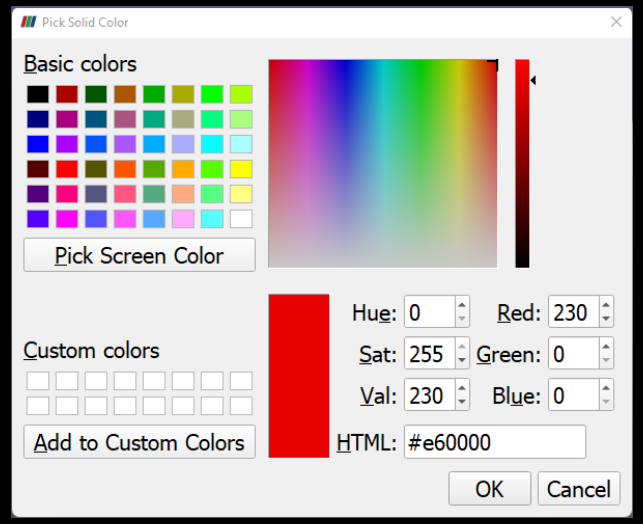
Specific object properties

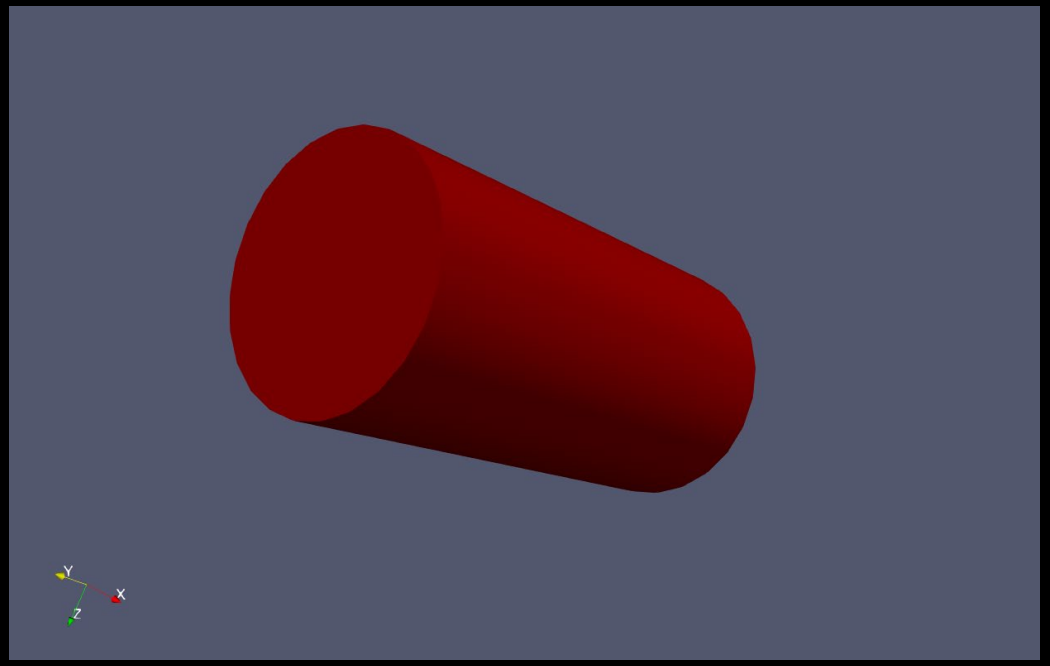




Don’t forget to press apply after a change or use the auto-apply button

Color properties

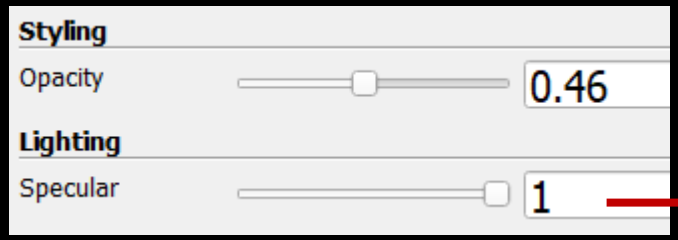




All objects can be given colors either solid or by values / vectors

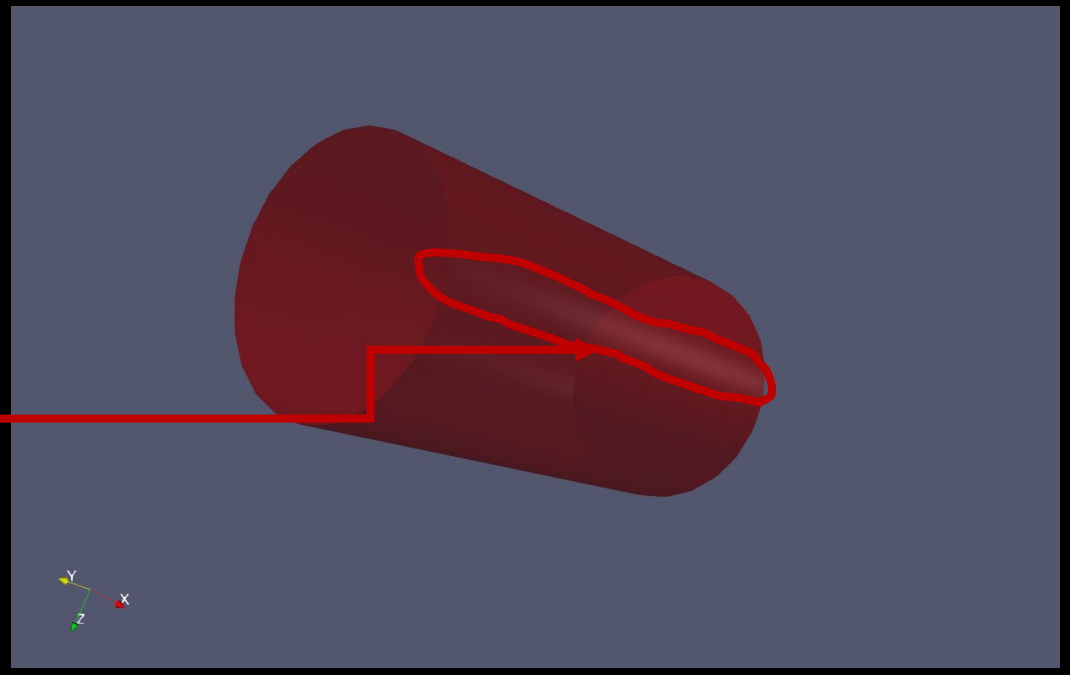
Material properties

Objects can also be given opacity and specular properties.



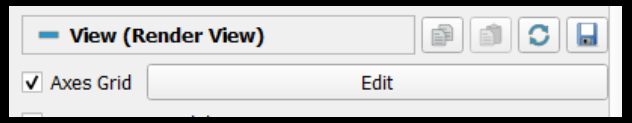
Specular controls how shiny an object is.

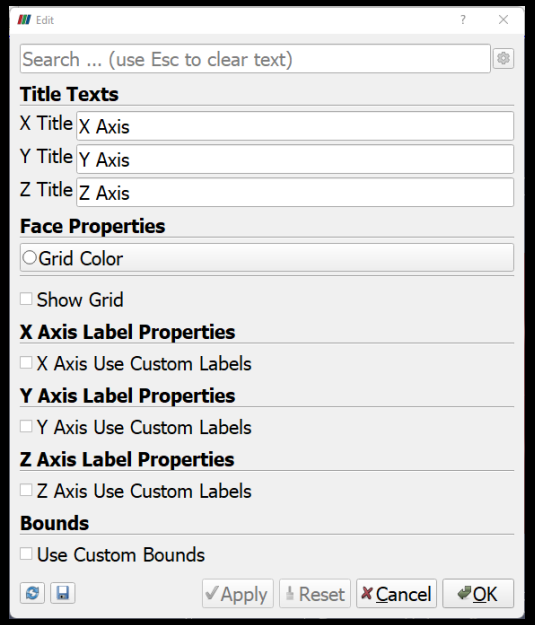
Opacity controls the transparency of an object. Can be useful to show outlines of objects.

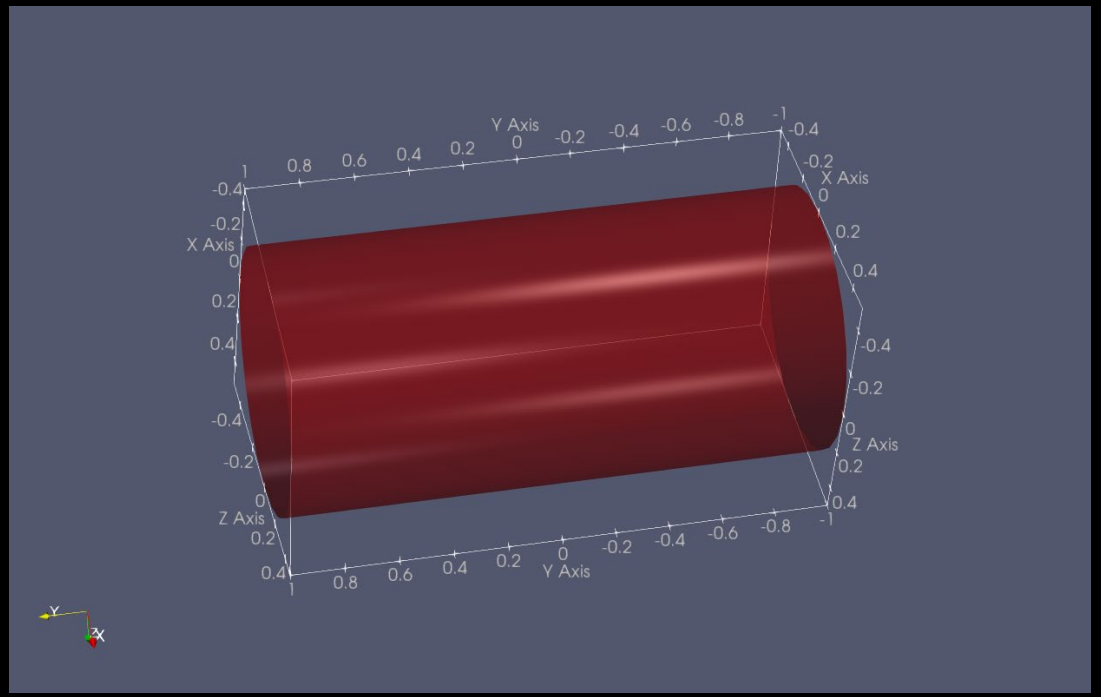


Axes

Axes can be used to give objects a sense of scale

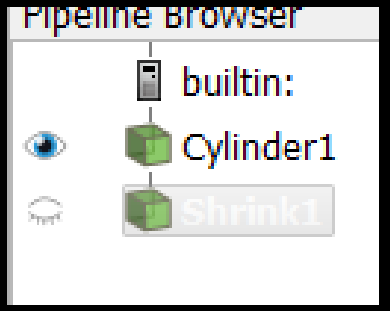


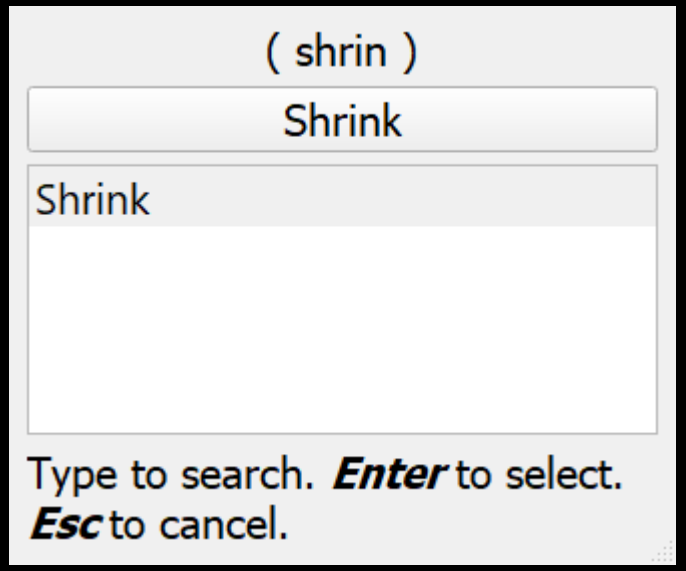




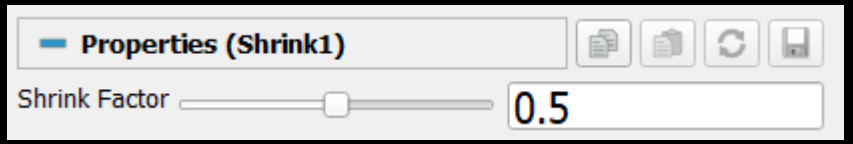
Applying a shrink filter

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

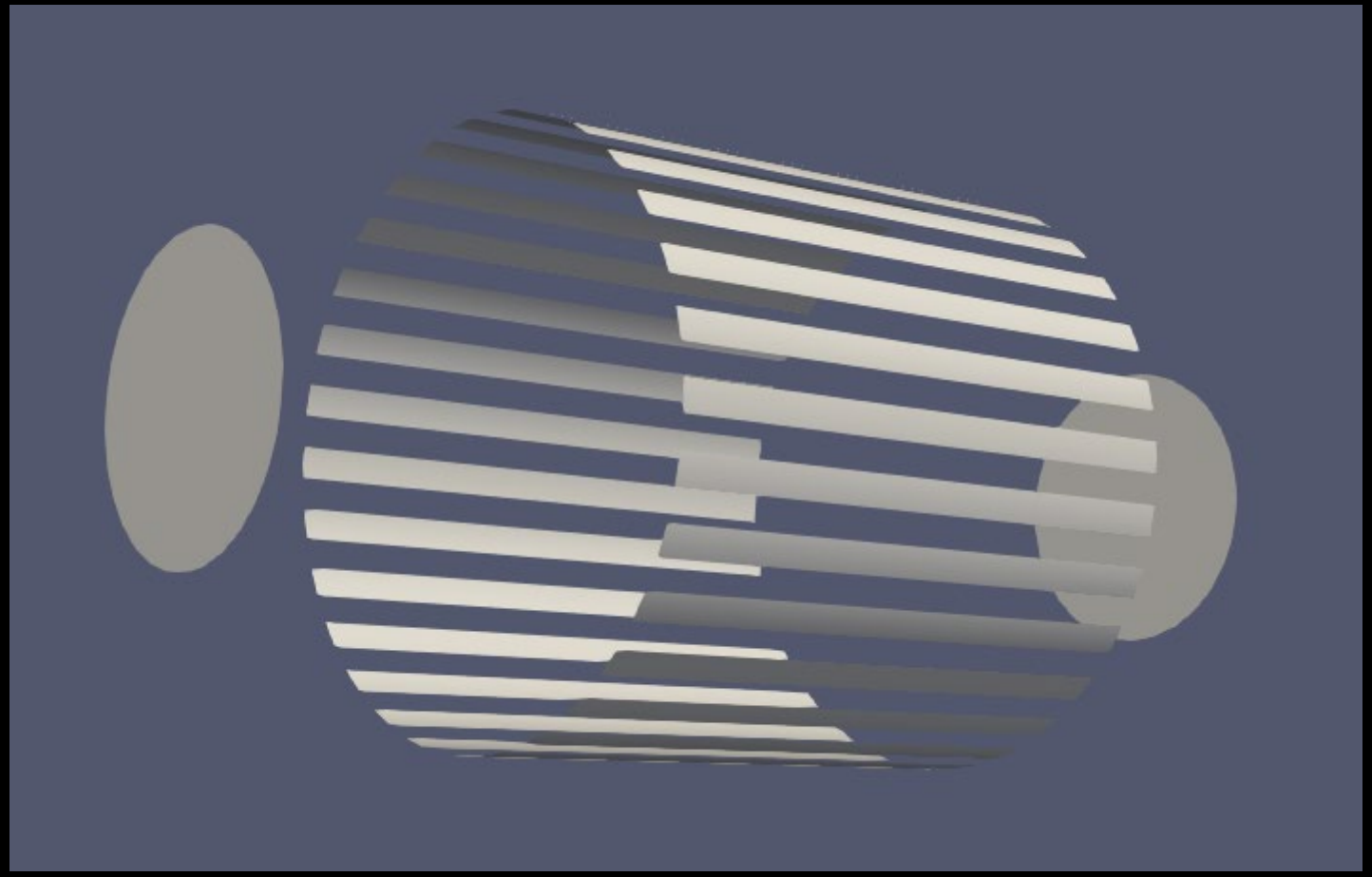




- Edit filter properties
- Click Apply



Applying a shrink filter



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


Demo time

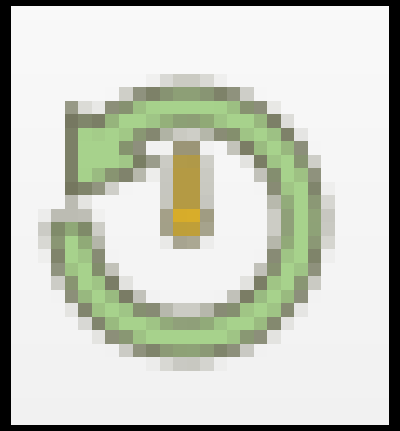
Exercise

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 shrinked sphere green. It should also be shiny.

Reset ParaView



Reading scalar data

Open the scalar.csv file

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

Reading CSV

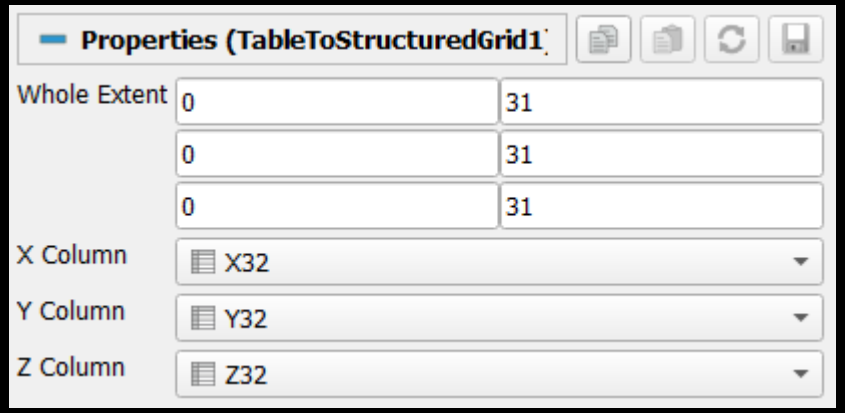
Showing scalar.csv Attribute: Row D

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

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

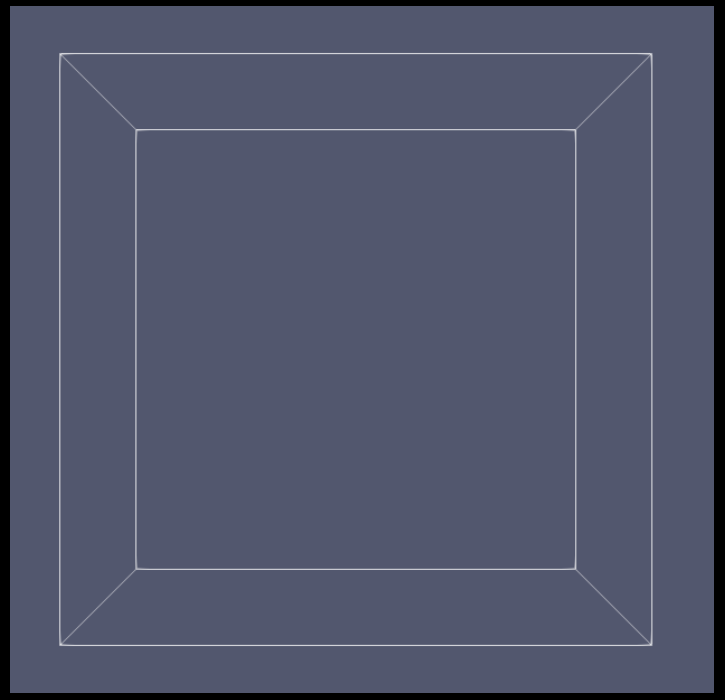
Todo this, we can use the **TableToStructureGrid** filter.

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.



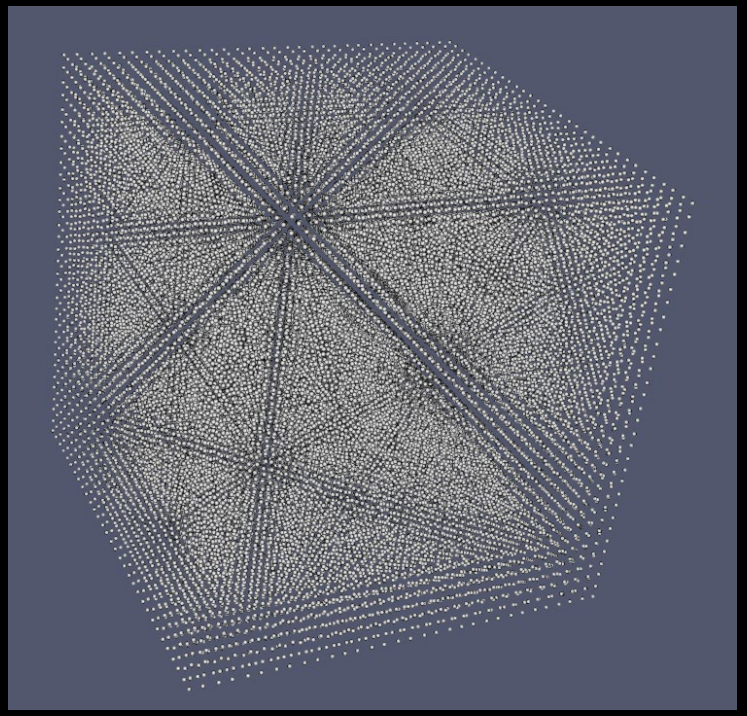
Visualising Scalar data

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



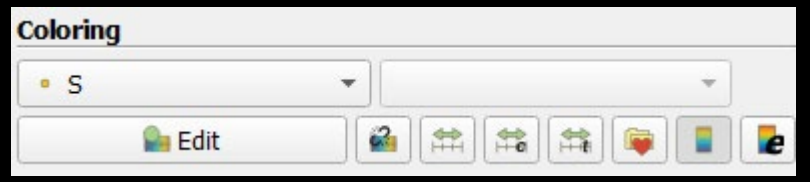
By default data is shown as an outline.

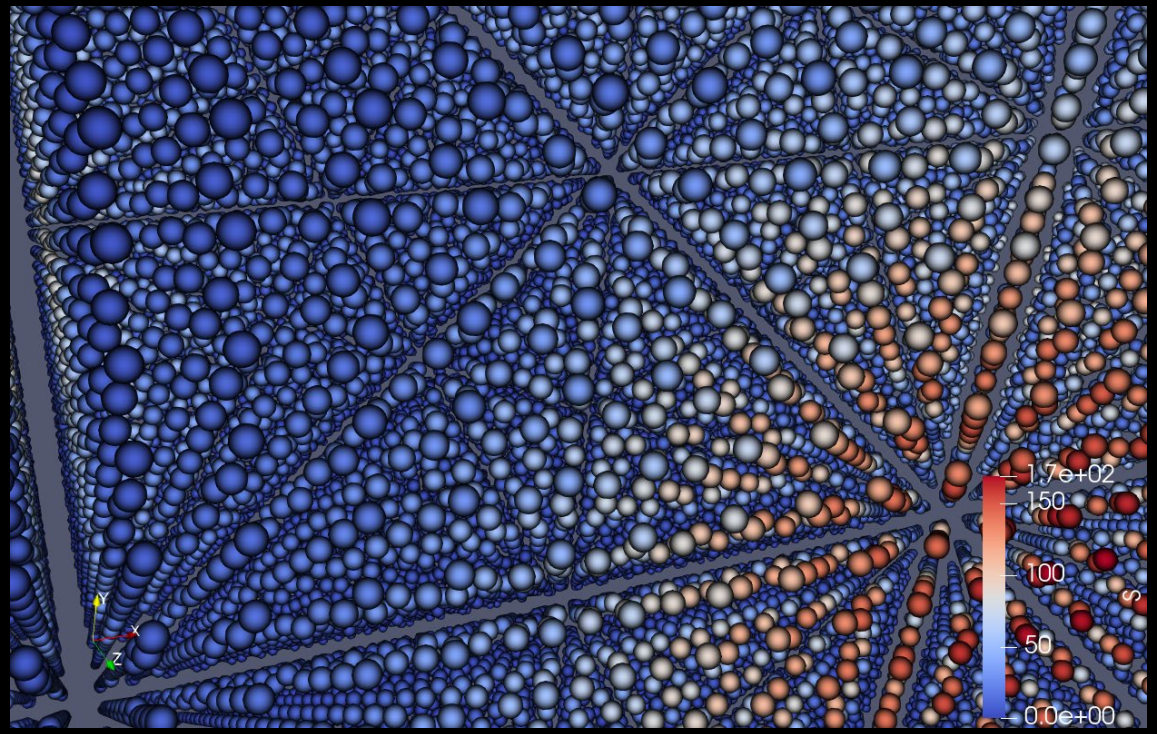
Change the representation to Point Gaussian to display the values in the grid.



Displaying colors

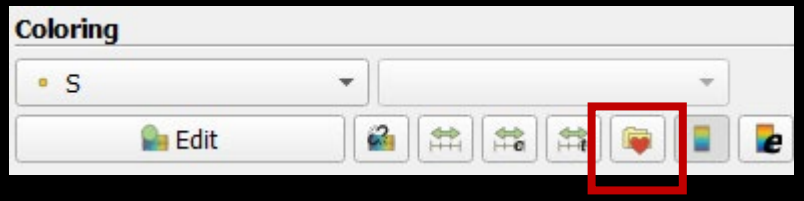
The CSV file contained an S column of values. To display these using a color scale we need to select the data field in the Coloring section:



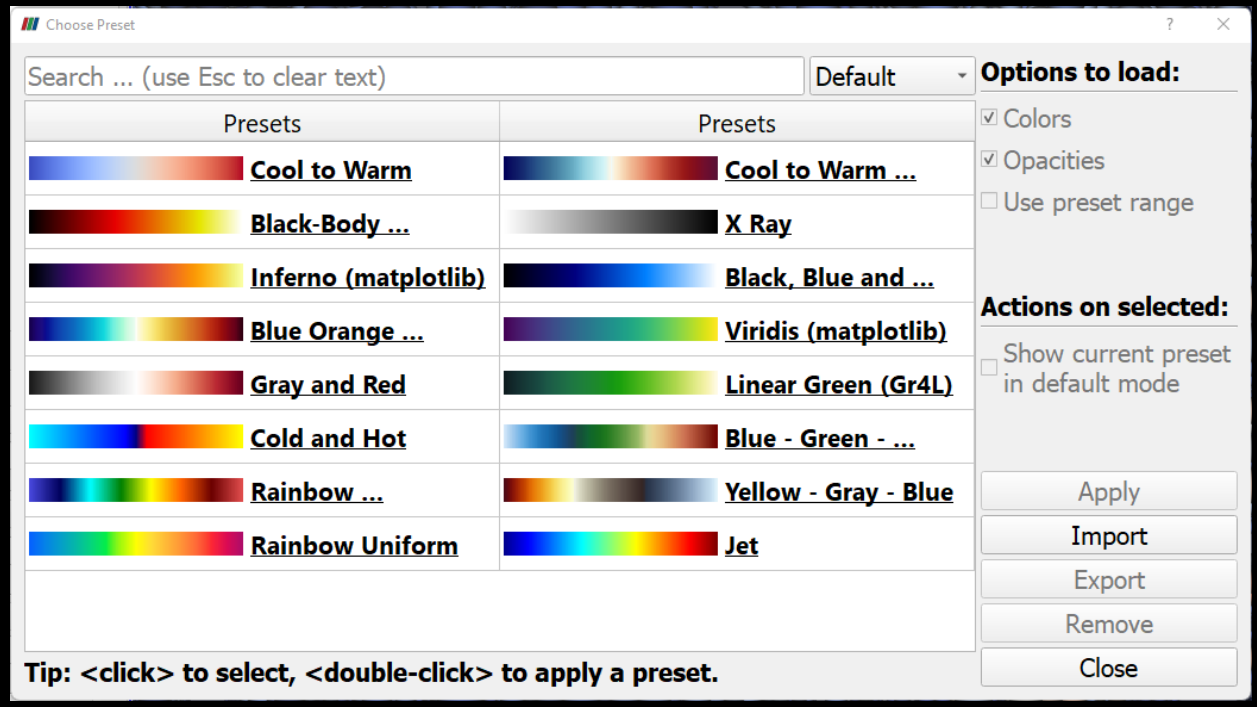


Changing color scale

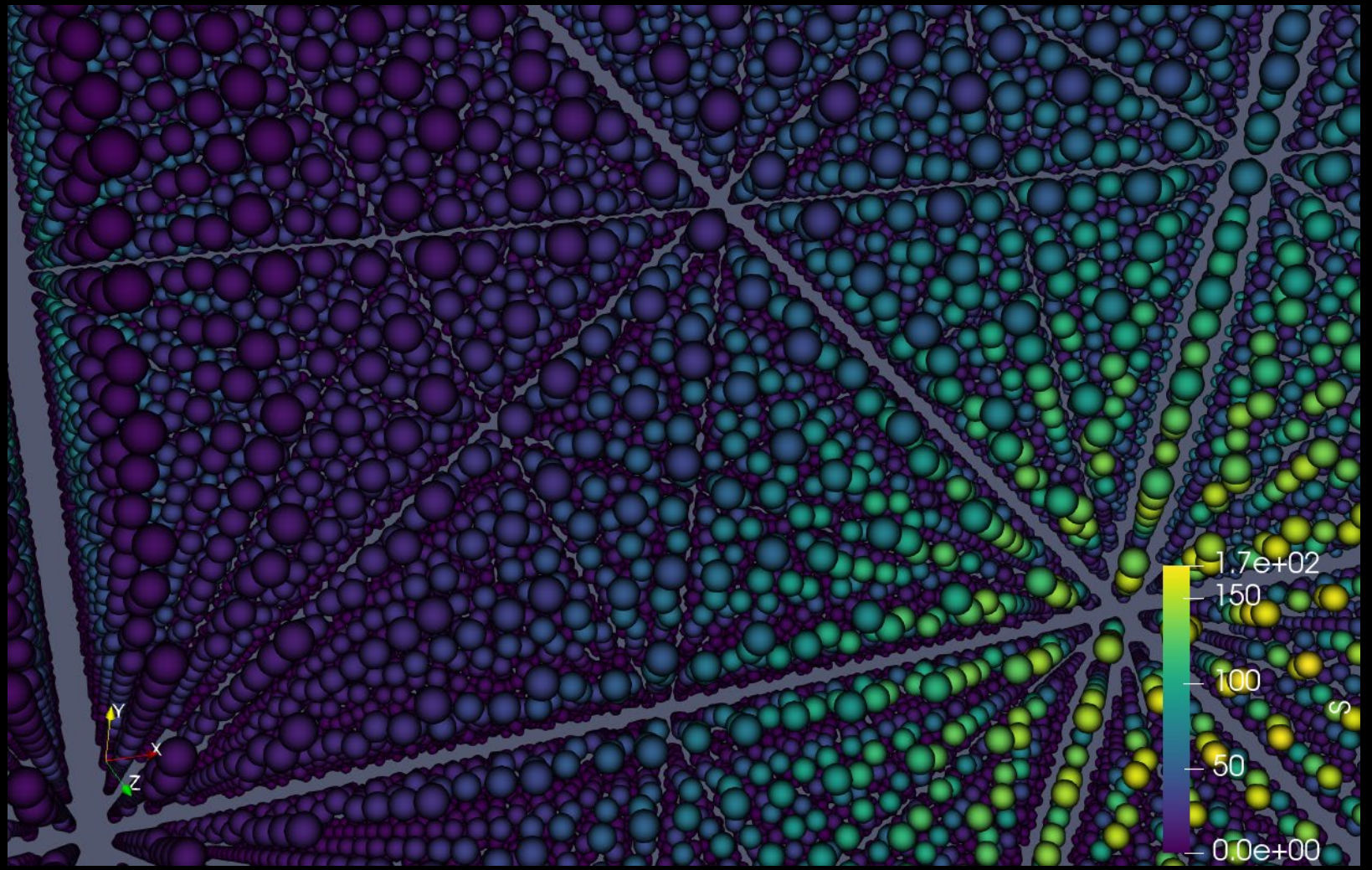
Changing color scale can be done by selecting the folder with a heart.



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

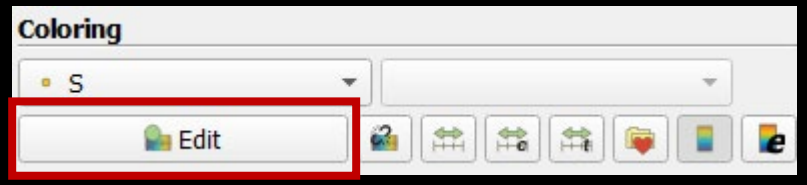


Changing color scale

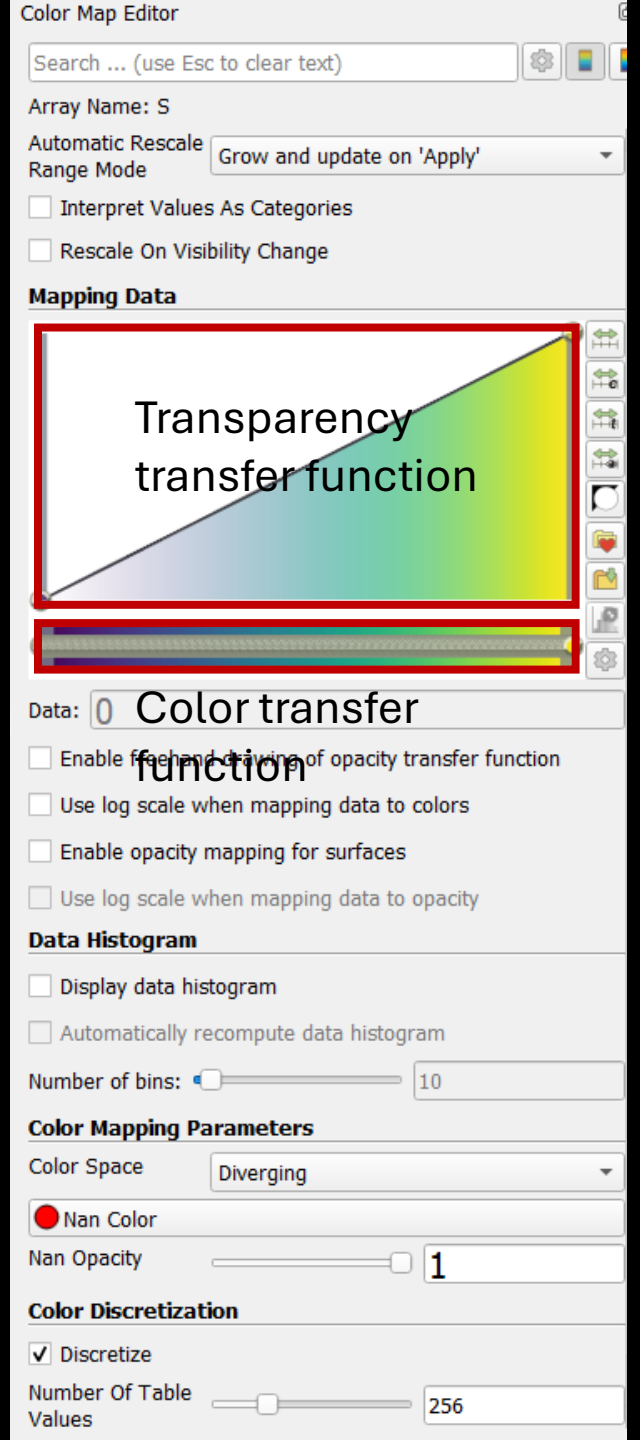


Changing color scale

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

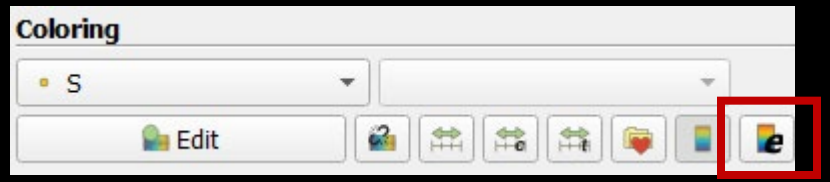


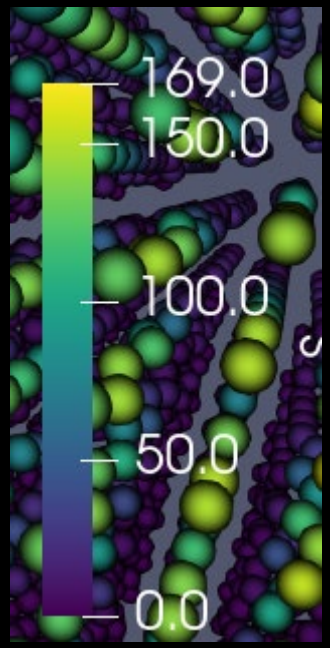
A selection of color scales are displayed. Select a color map and click Apply to change.



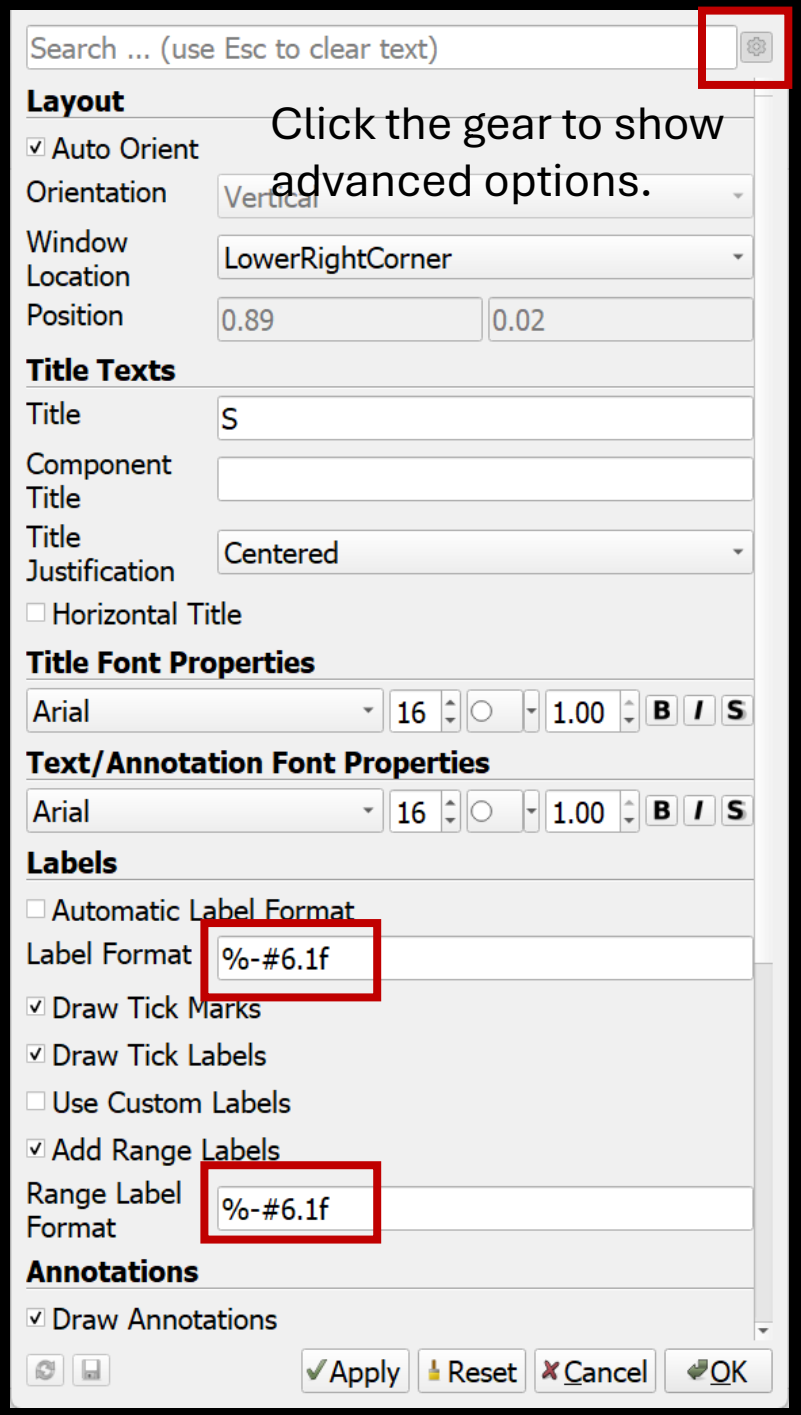
Changing the legend

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





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



Click the gear to show advanced options.


Demo time

Exercise

Change the color scale to "Inferno"

Map the range from 60-160

Use opacity in the transfer function

Using filters with scalar data

Common filters

Contour. Creates contour lines or surfaces.

Slice. Create a slice plane.

Threshold. Extracts data for a certain criteria

Glyph. Generate glyphs at data points

Warp by vector. Warps data mesh with a vector field

Calculator. Computes new fields based on existing fields.

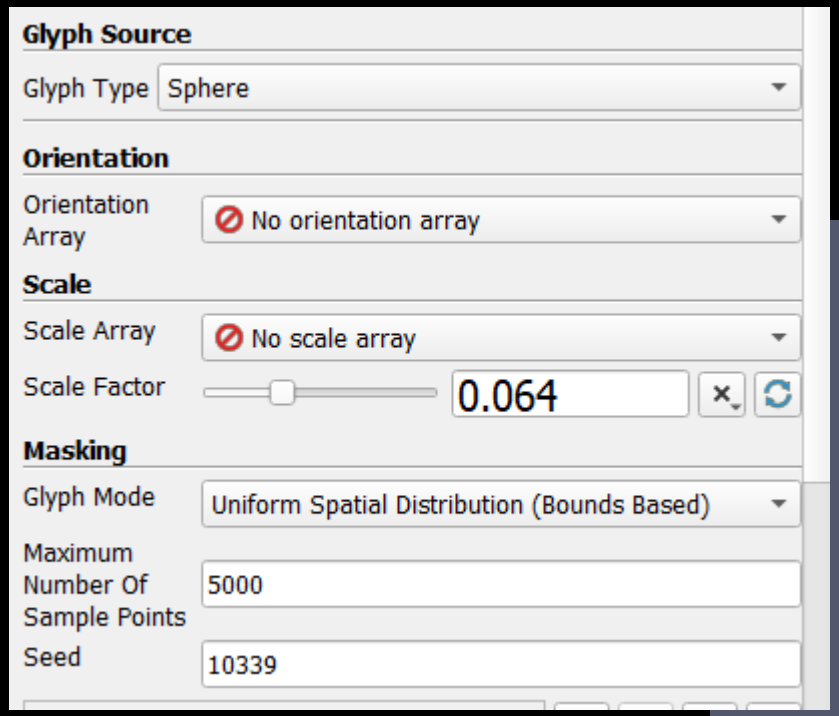
Clip. Clips the data at a certain plane

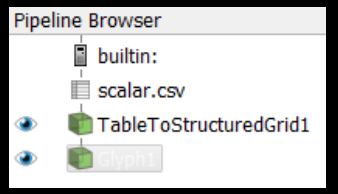
Subset. Extracts a sub grid of the data

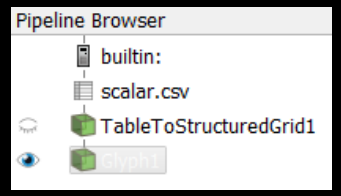
Stream tracer. Generate glyphs at data points

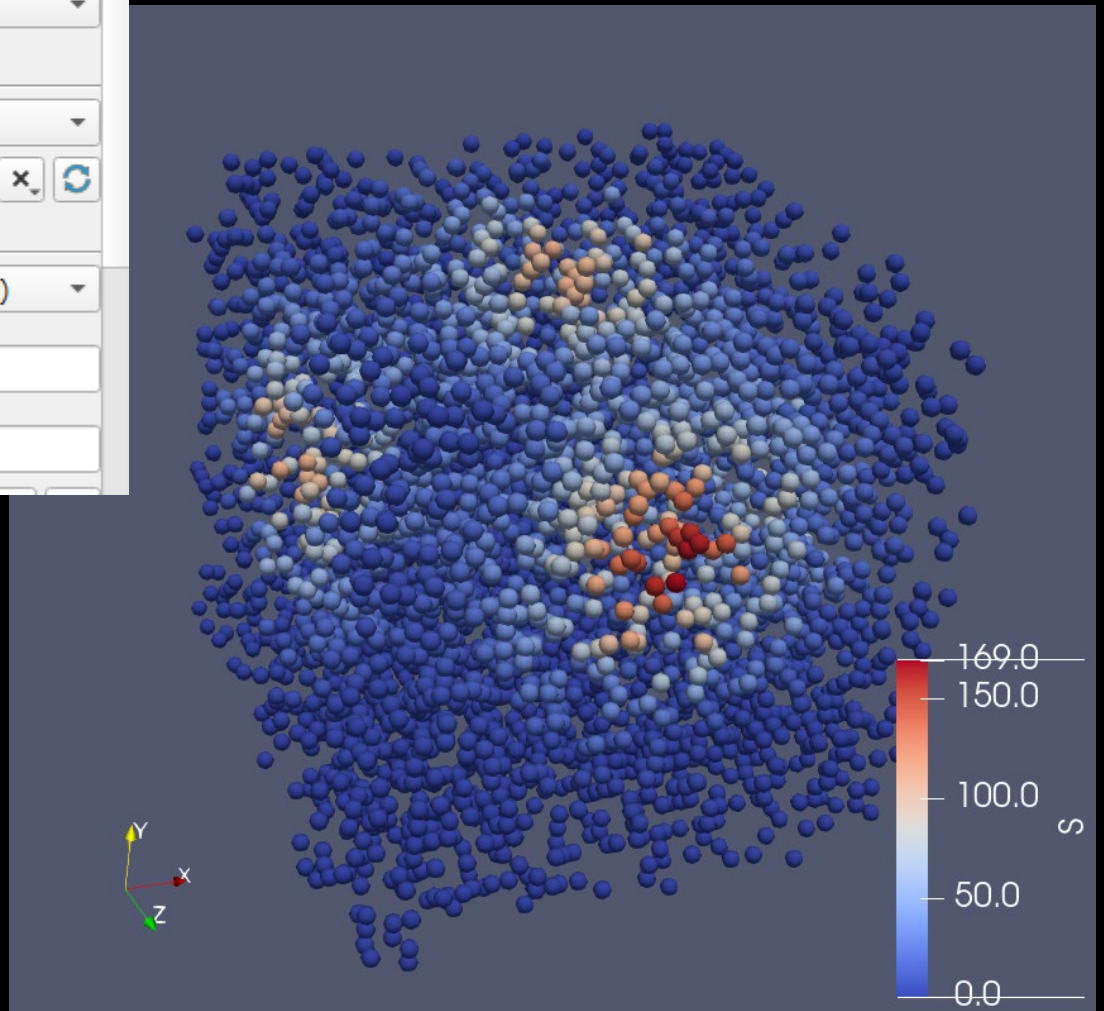
Glyph cloud

We need to turn off the TableToStructuredGrid

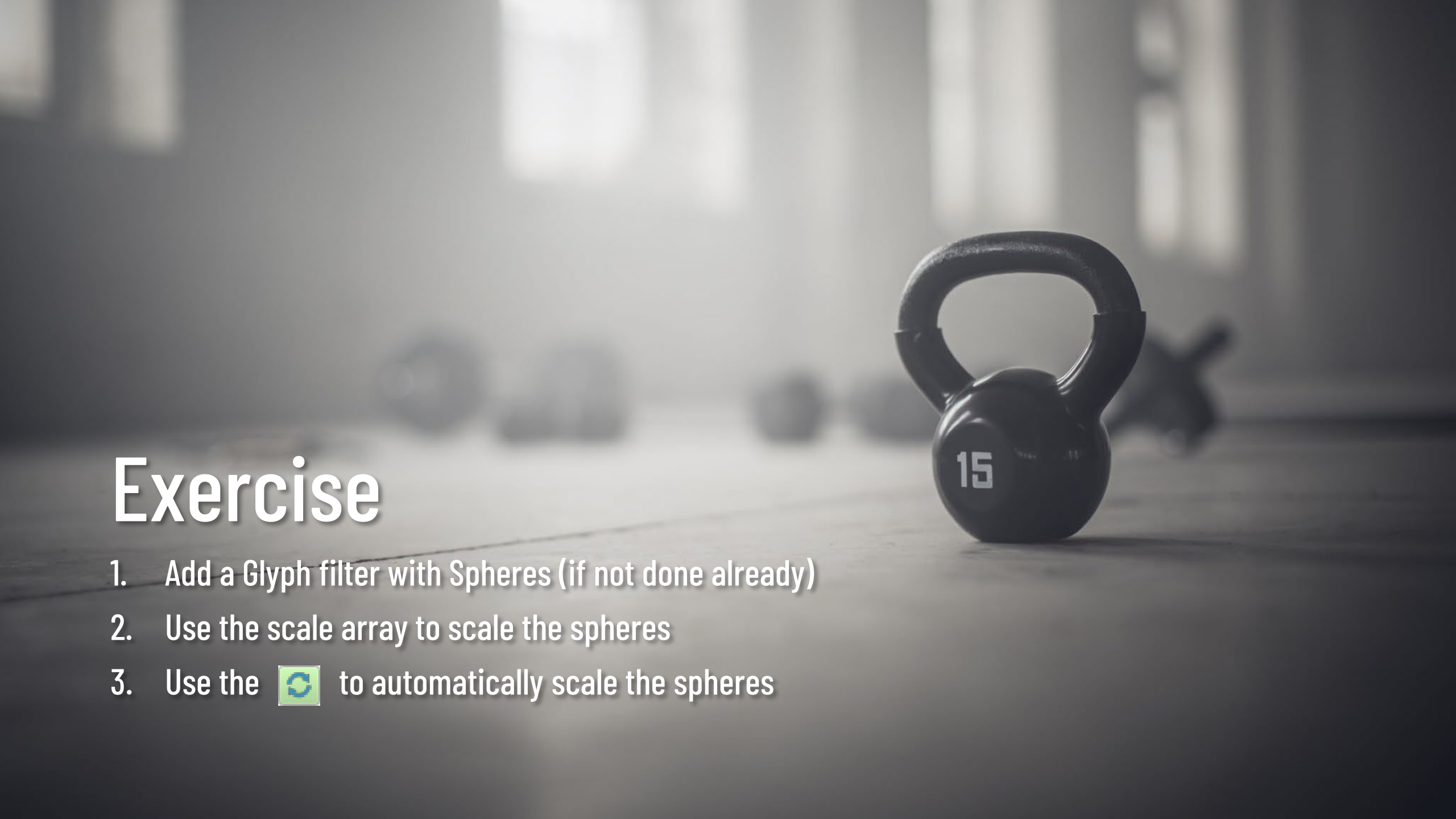









Demo time



Exercise

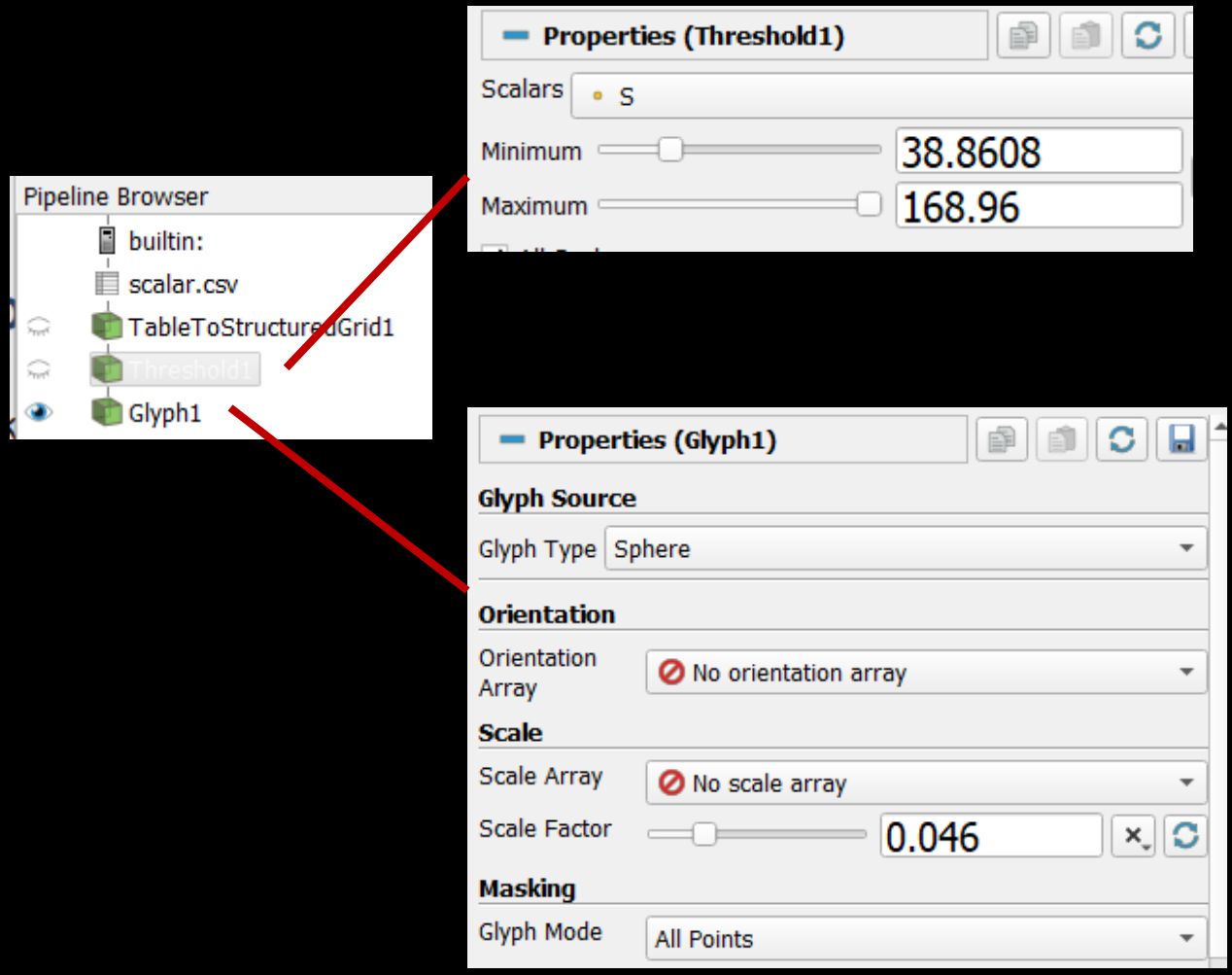
1. Add a Glyph filter with Spheres (if not done already)
2. Use the scale array to scale the spheres
3. Use the to automatically scale the spheres

Tresholding data

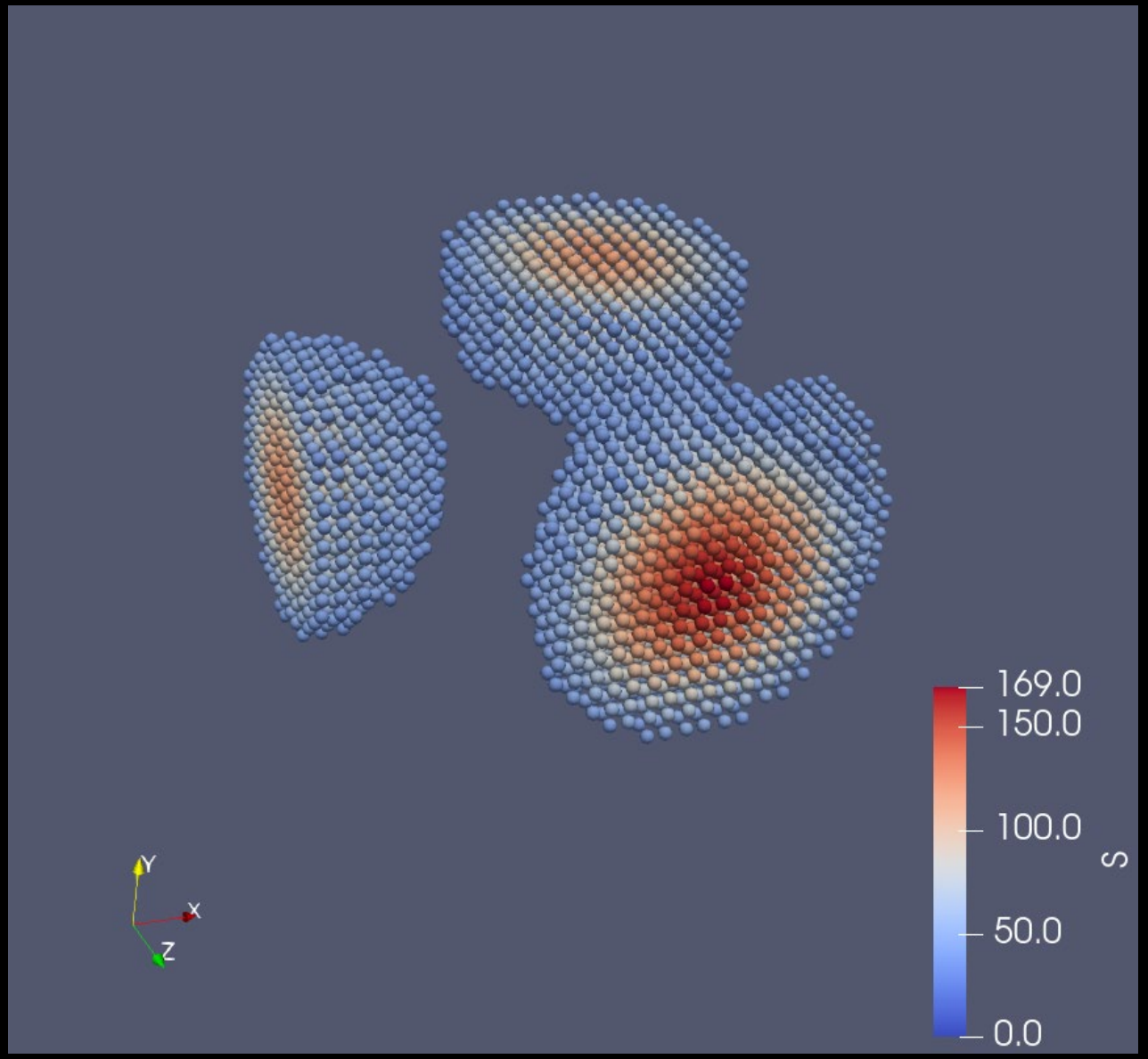
We want to limit data in the glyph visualisation.

We Create the following pipeline

We use **Delete** to remove the existing filters and apply



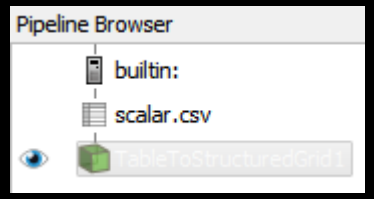
Thresholding data



Creating a cutting plane

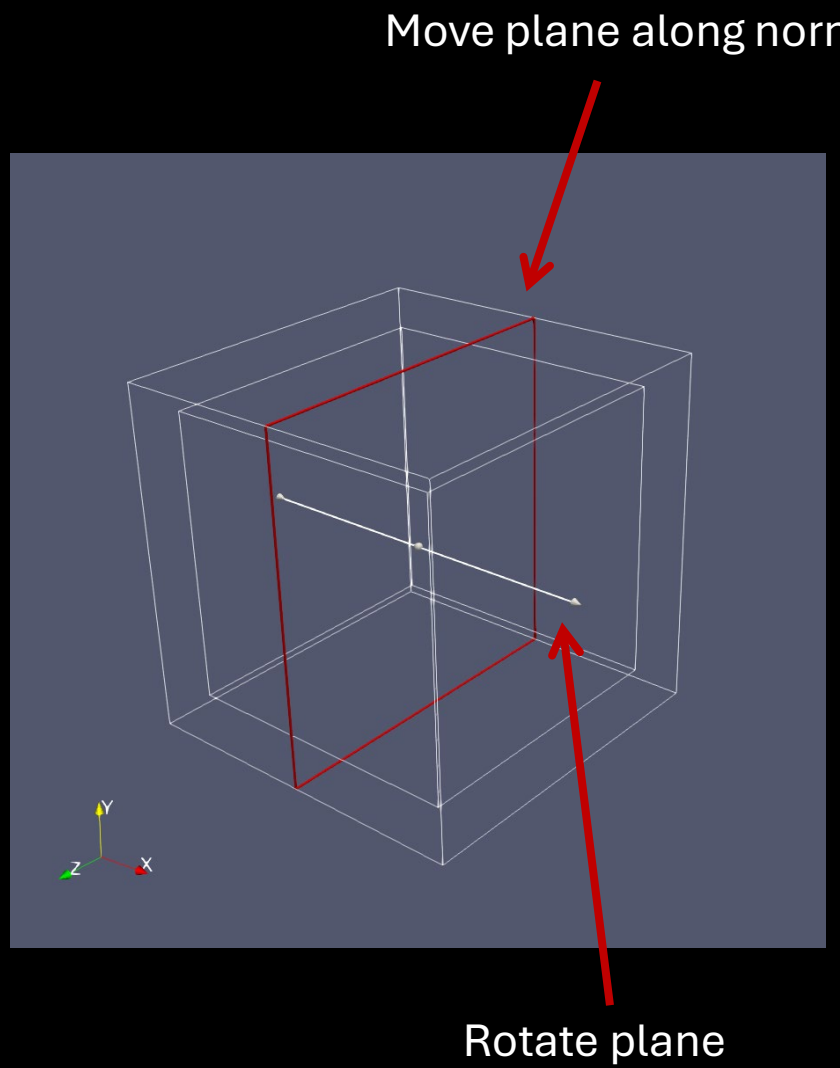
Delete all filters except the TablesToStructuredGrid

Set the filter representation to Outline

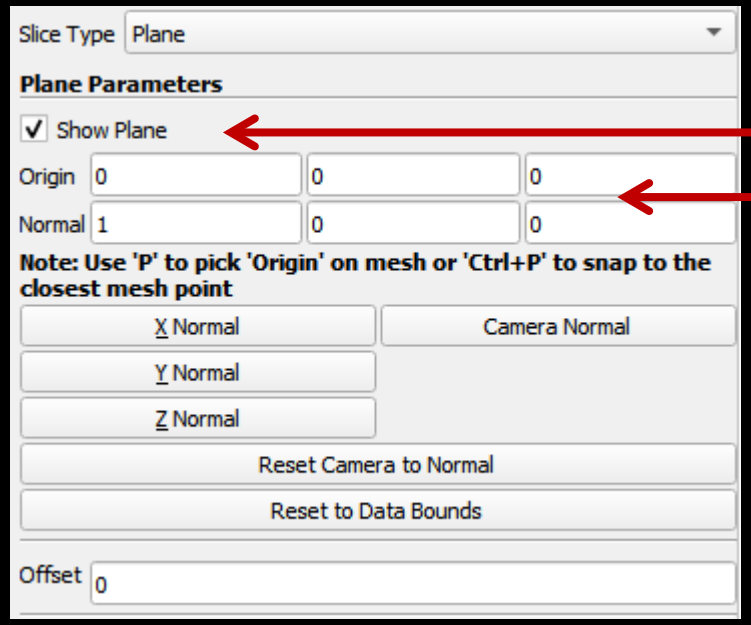


Add a Slice filter 

The view shows a representation of the slice plane.



Creating a cutting plane

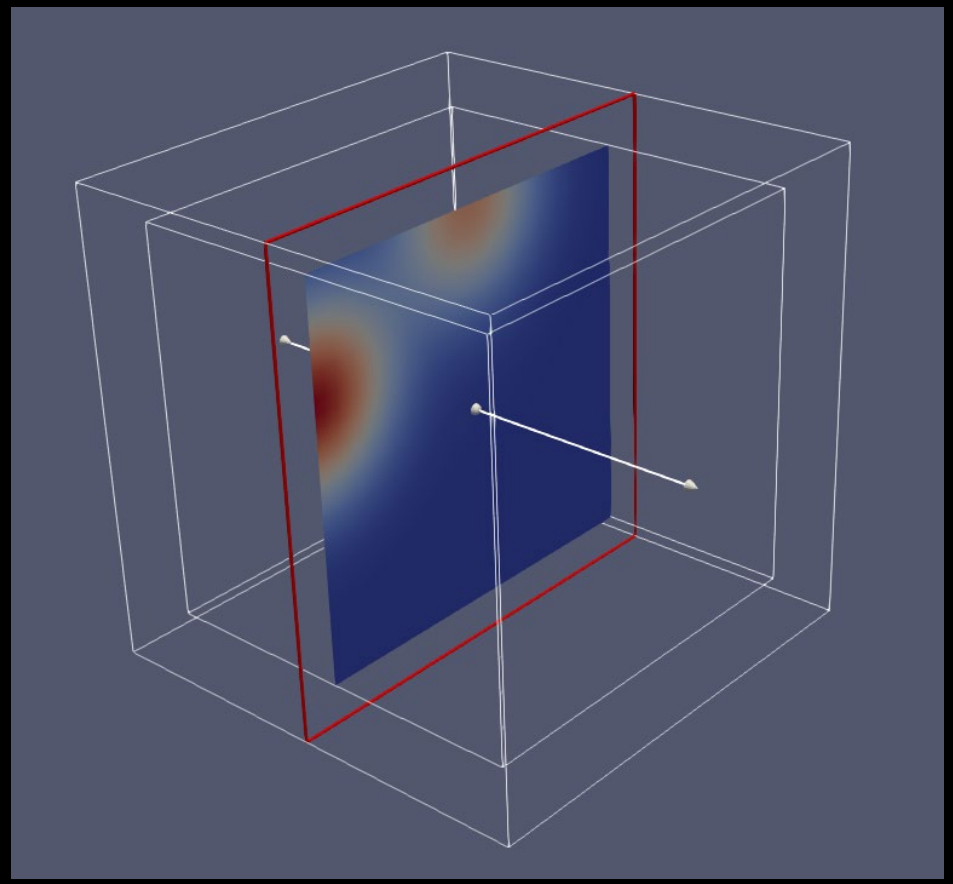


Show a visual representation of the plane

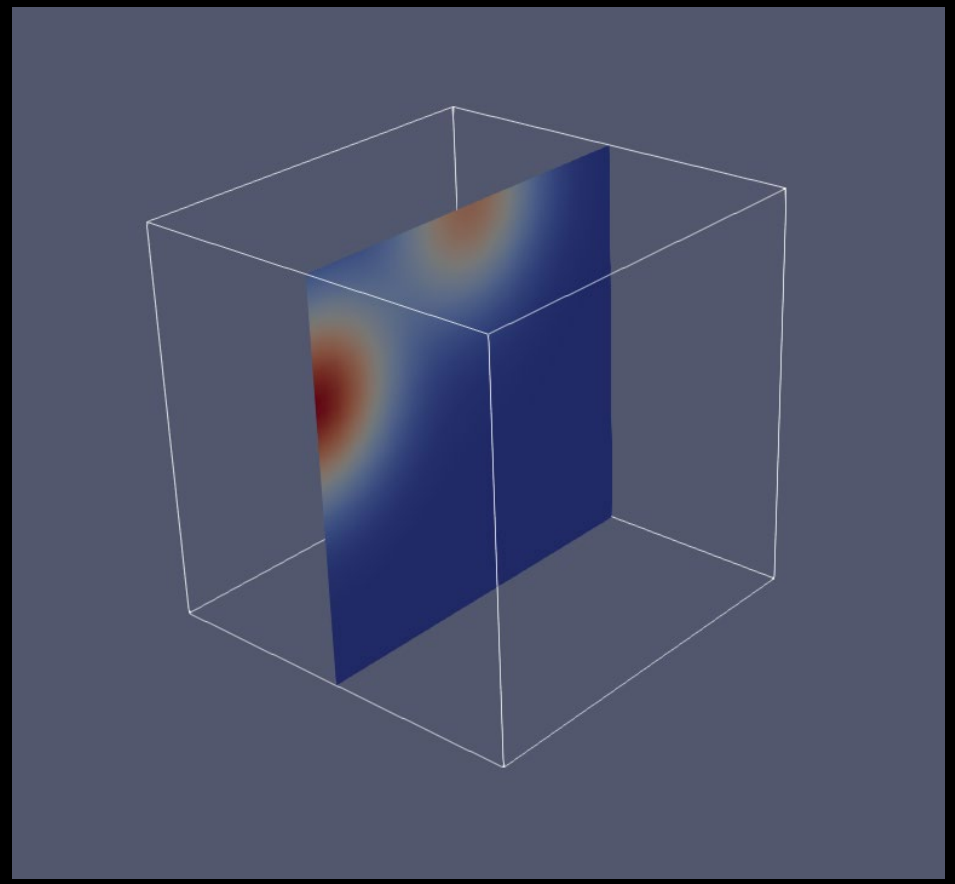
Manual plane parameters

Make sure to color the plane according to the S field

Creating cutting plane



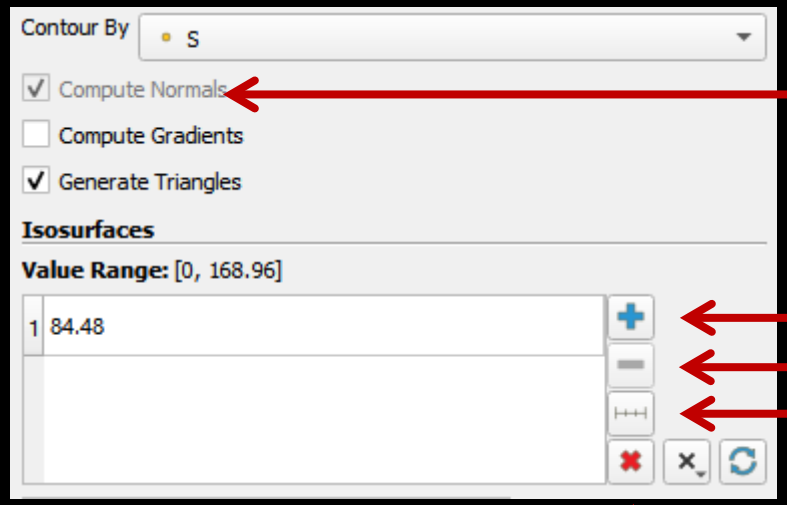
☑ Show Plane



☐ Show Plane

Displaying cutting planes with contours

Add a Contour filter 



Show a visual representation of the plane

Add contour line

Remove contour line

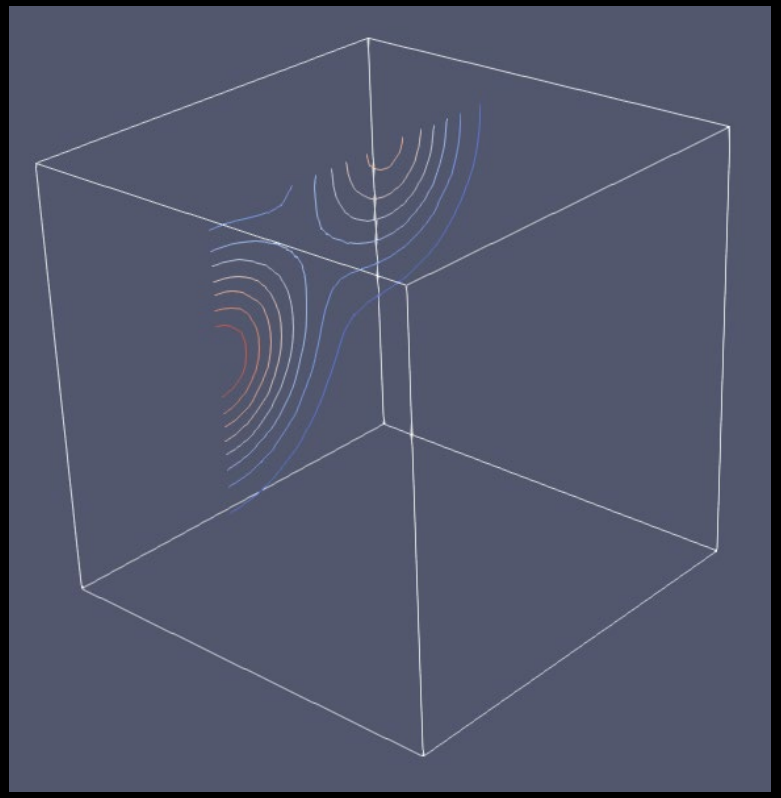
Create multiple contour lines

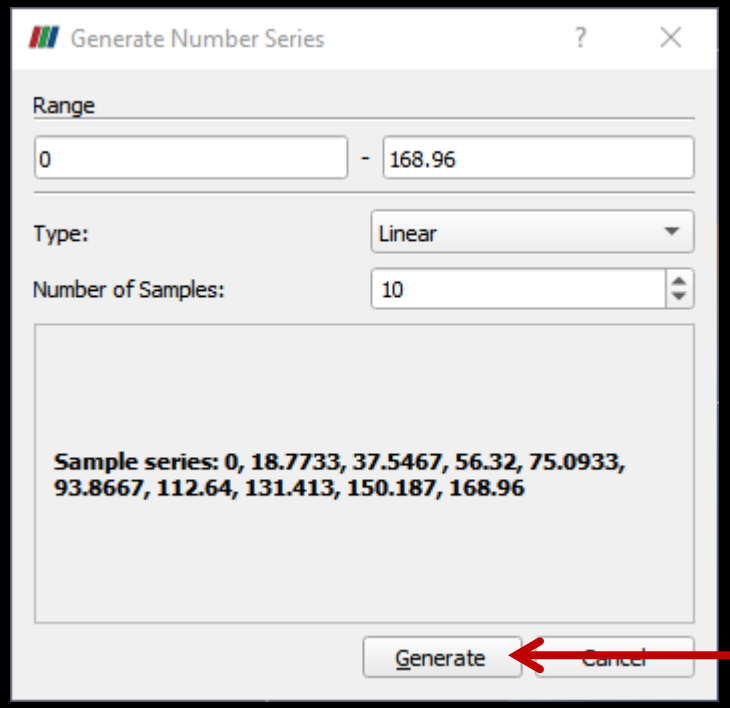
Clear all contour lines

Displaying cutting planes with contours

Now we erase all existing contour lines

Create a range of contour lines

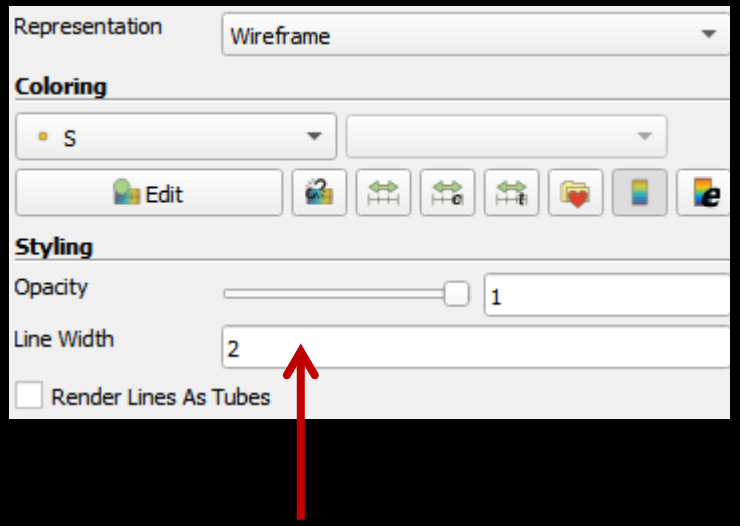




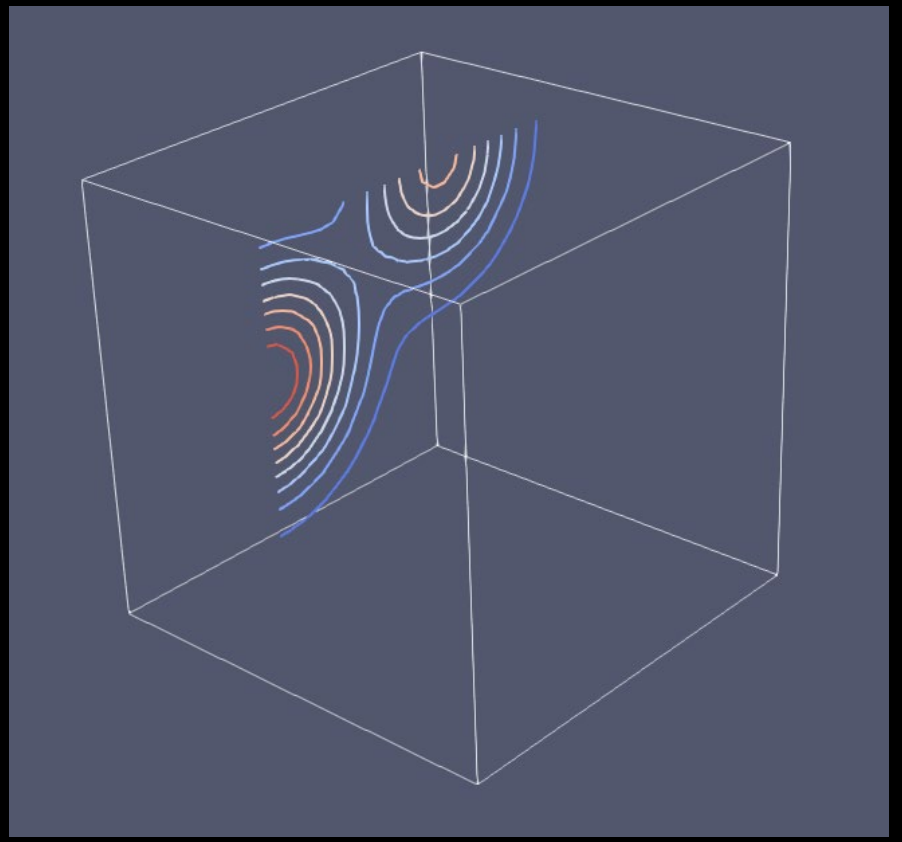
Click **Generate** and then **Apply**

Displaying cutting planes with contours

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

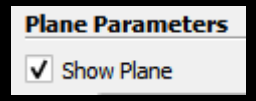


Line thickness

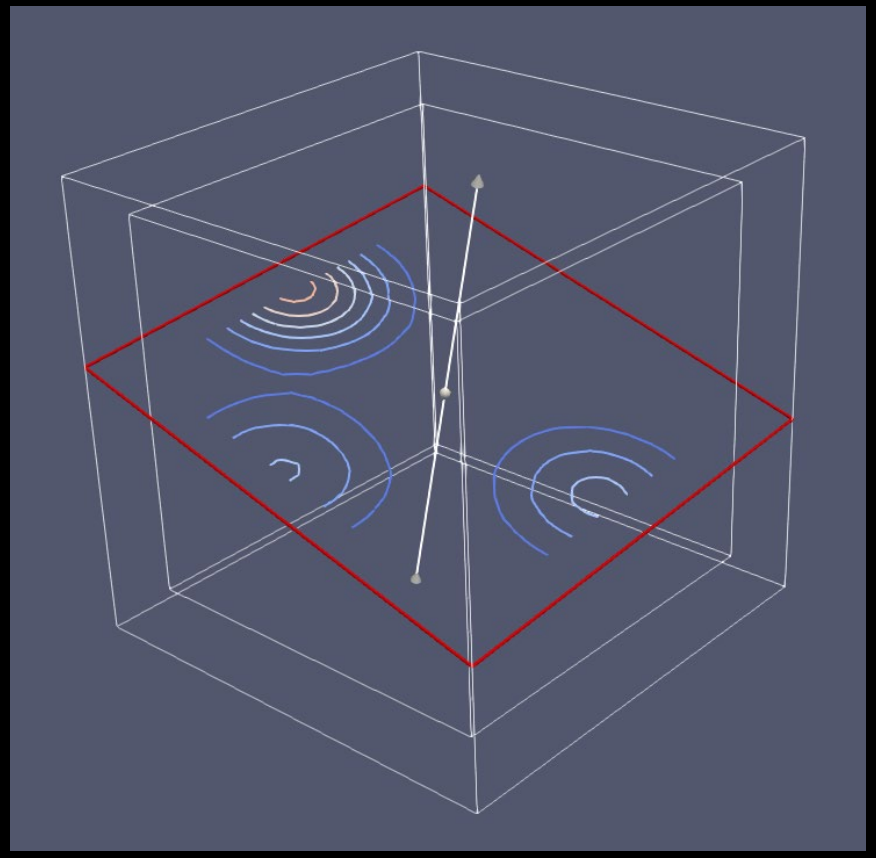


Displaying cutting planes with contours

Select the Slice filter and enable

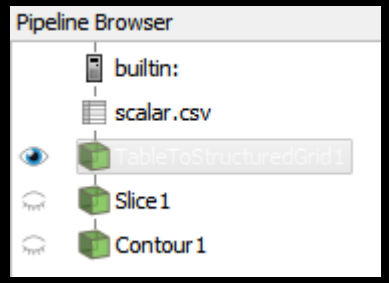


This enables you to move the cutting planes displaying contours on the plane.

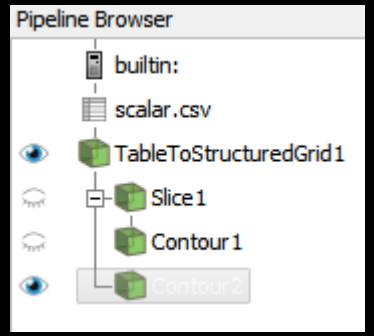


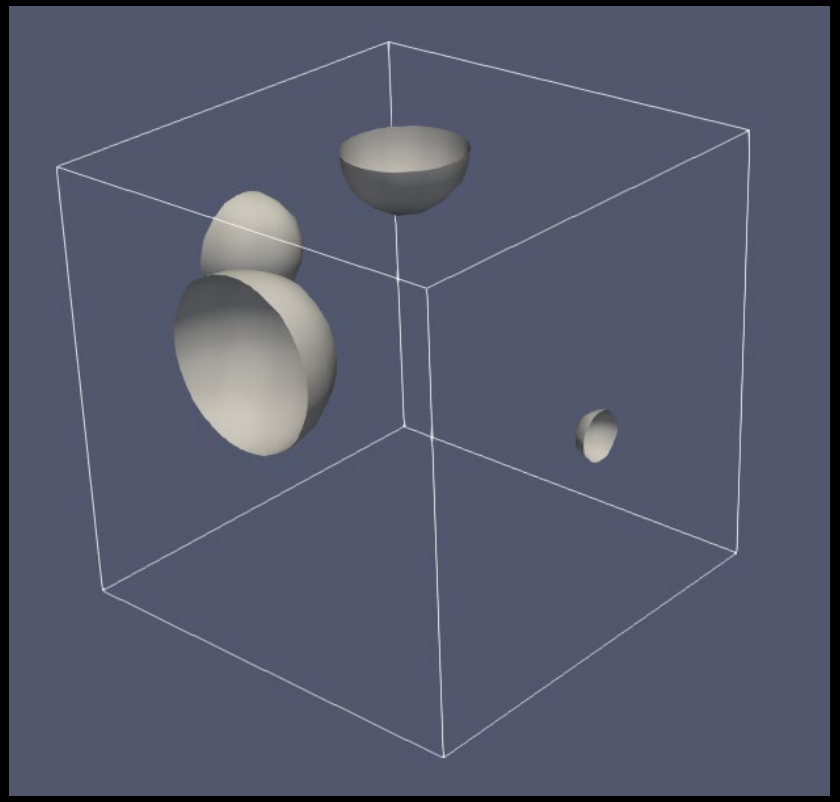
Displaying iso surfaces

Select **TableToStructuredGrid1** and disable the other filters



Create a **Contour** filter from the grid 



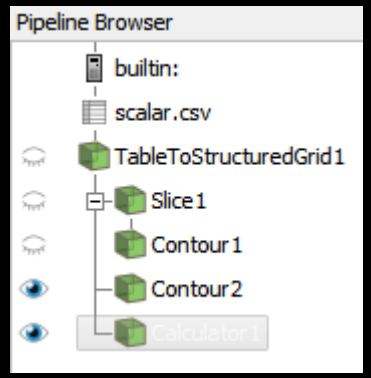


Notice that it is not possible to color the iso lines as S is not availble anymore.

We need a way of duplicating the S field

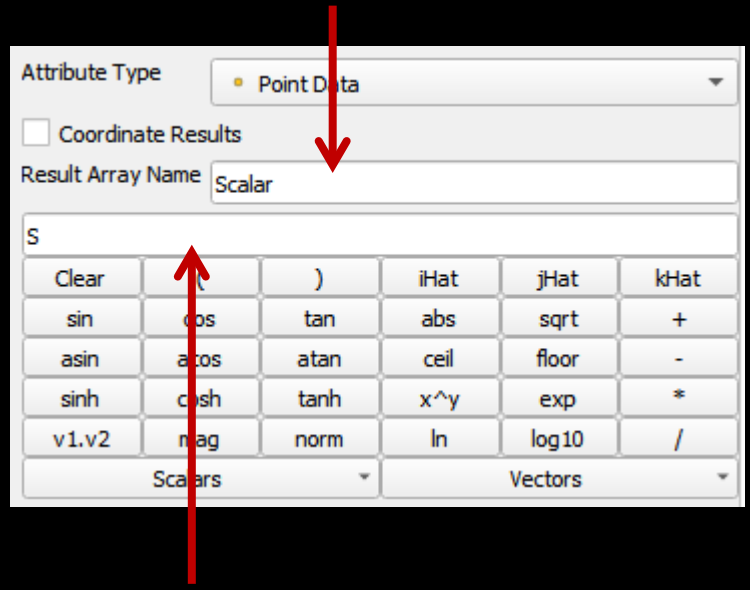
Displaying iso surfaces

Create a Calculator filter from the TableGrid



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

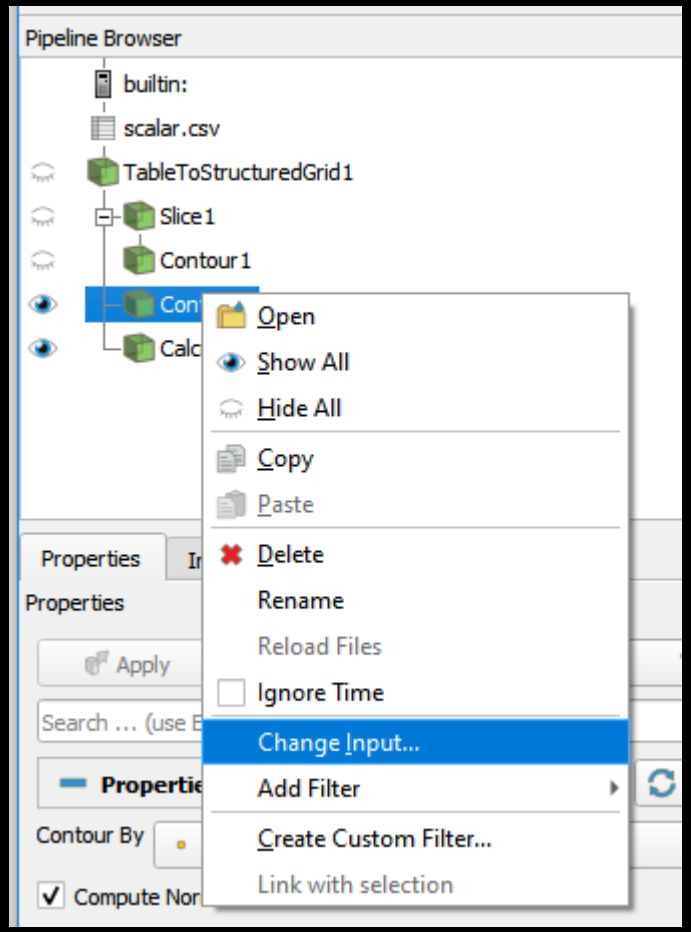
Name the result of the expression. We name it Scalar

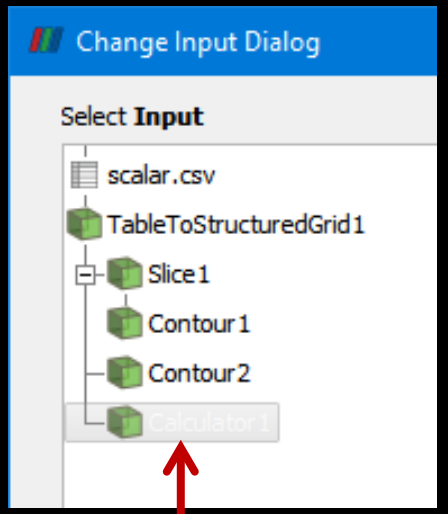


Enter the expression here. In this case just return the scalar value from S

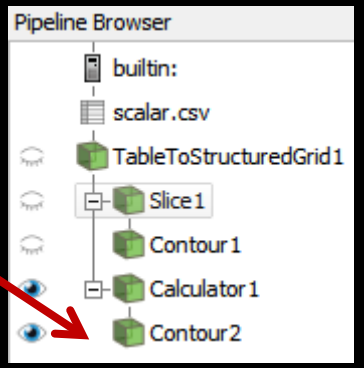
Displaying iso surfaces

Moving Contour2 after Calculator



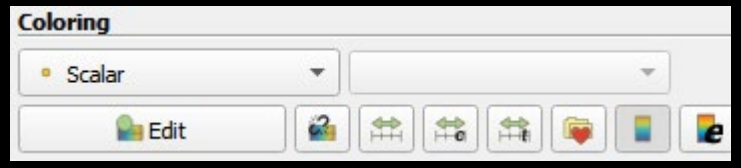


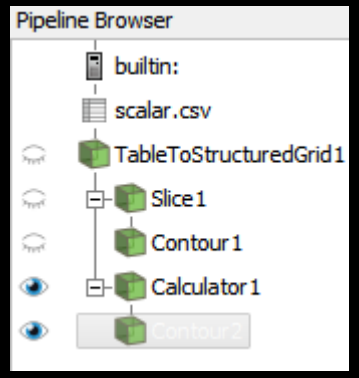
Move here

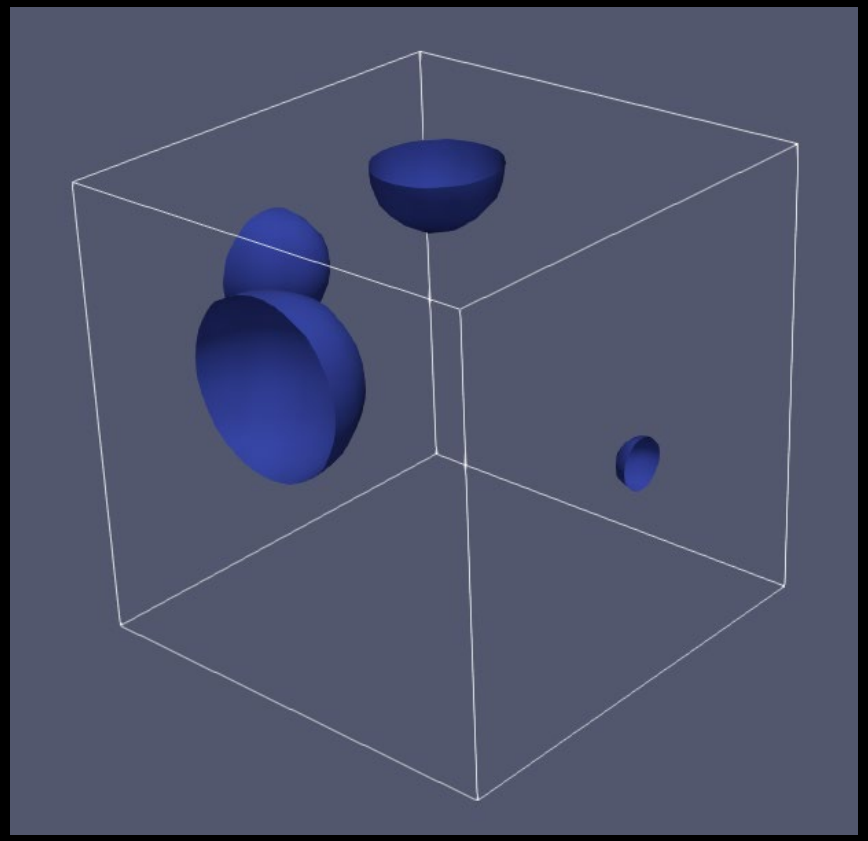


Displaying iso surfaces

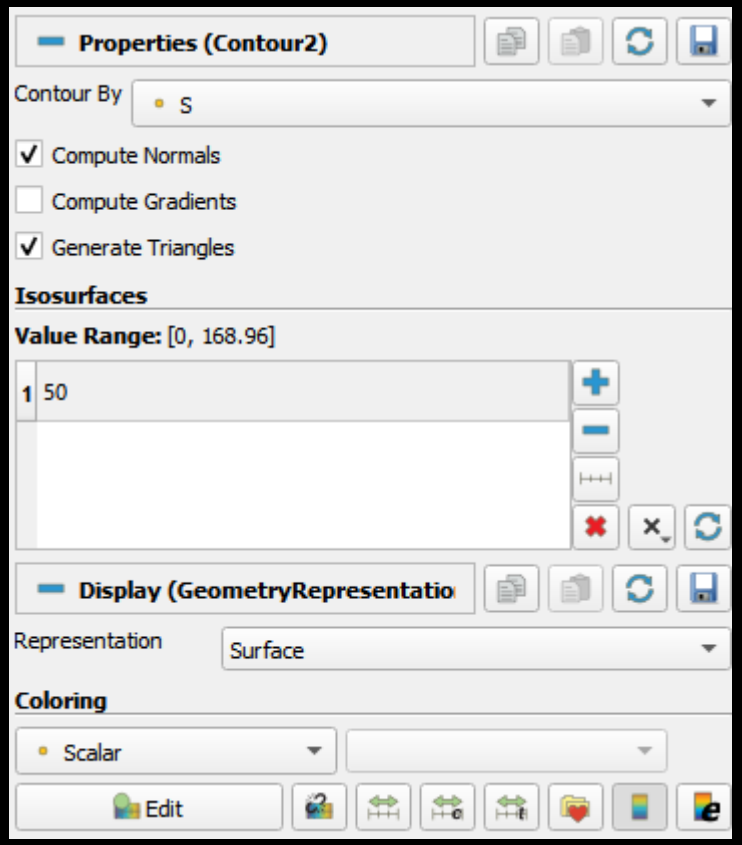
We can now select the computed field in the Contour filter

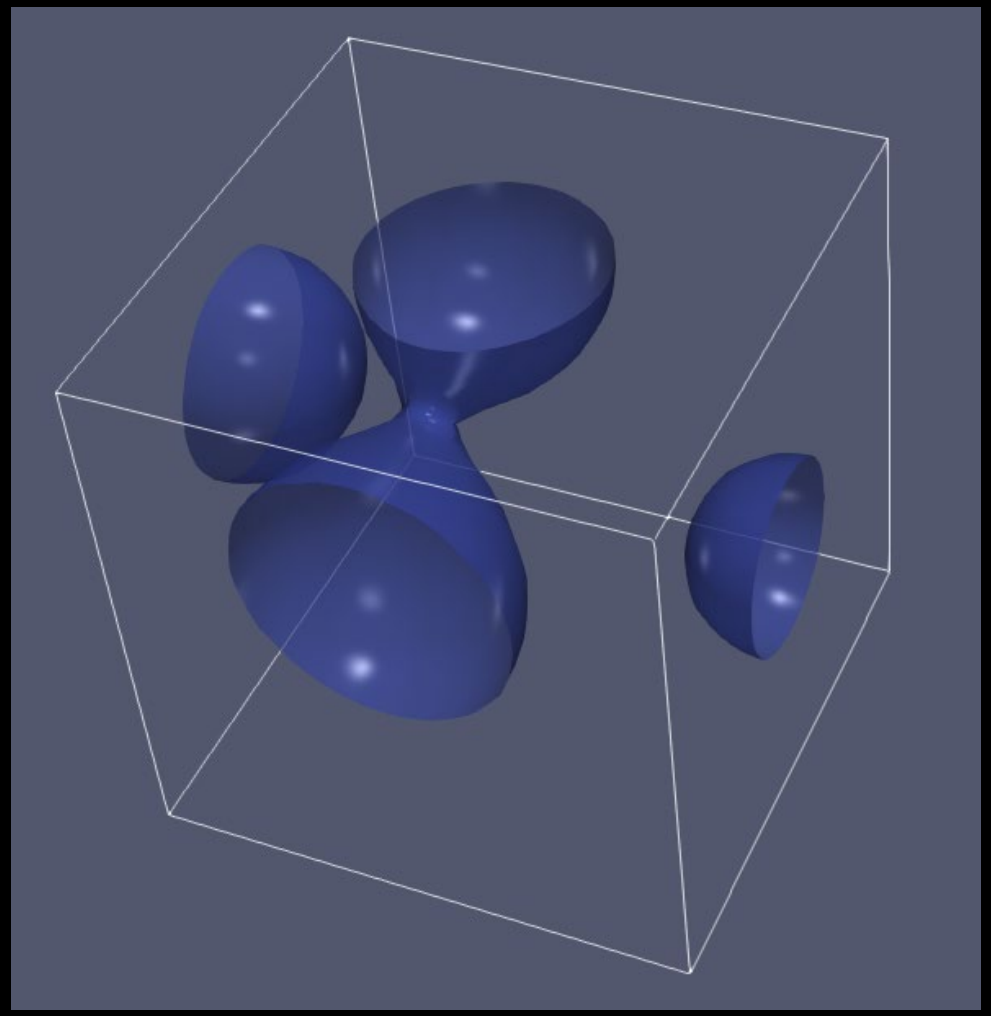




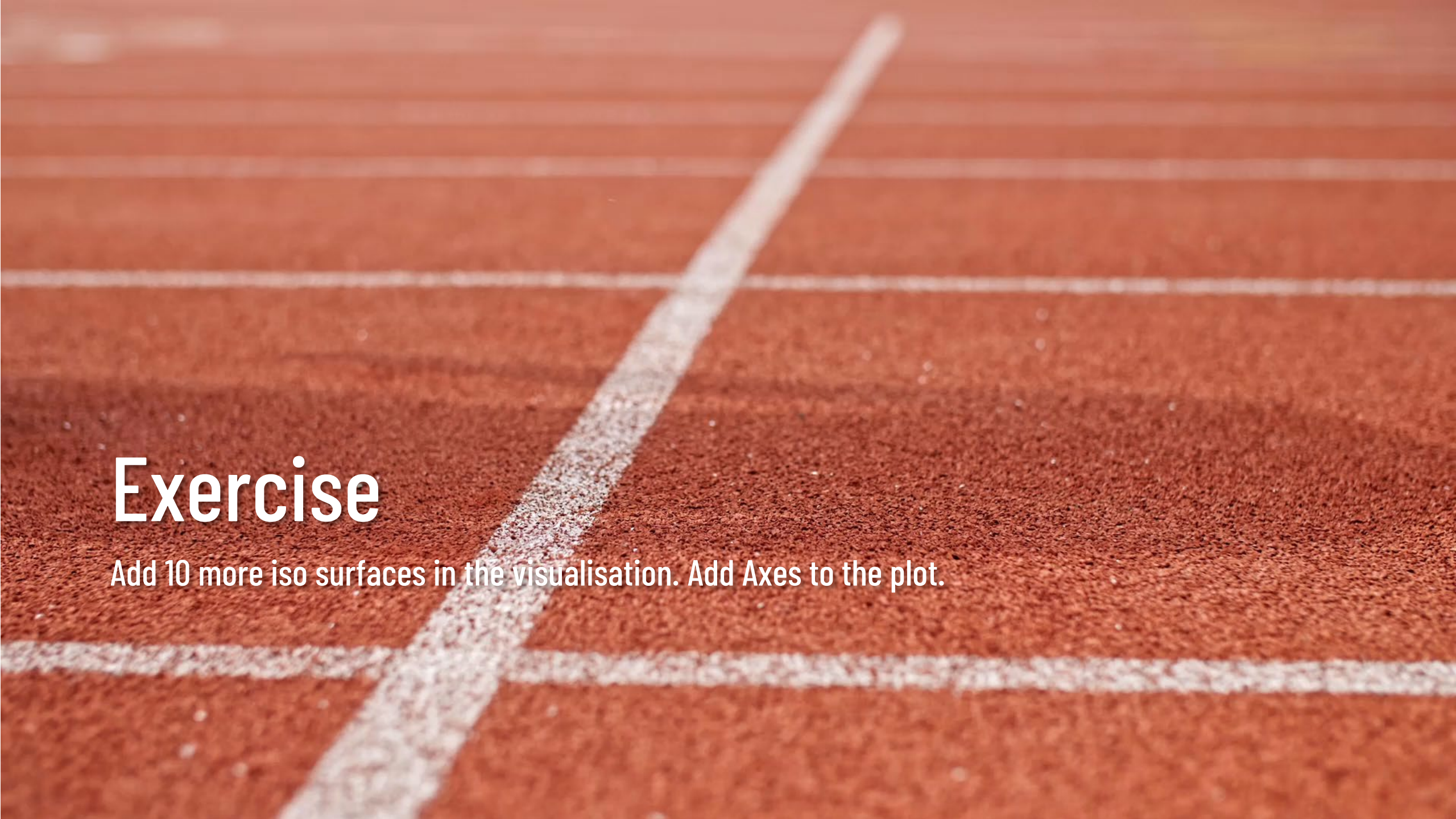


Displaying iso surfaces






Demo time

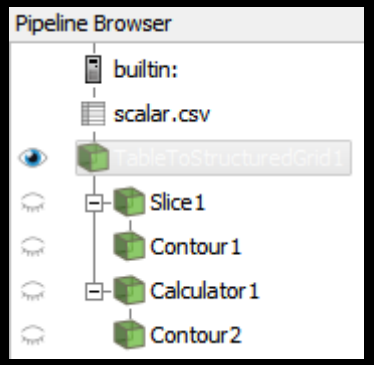


Exercise

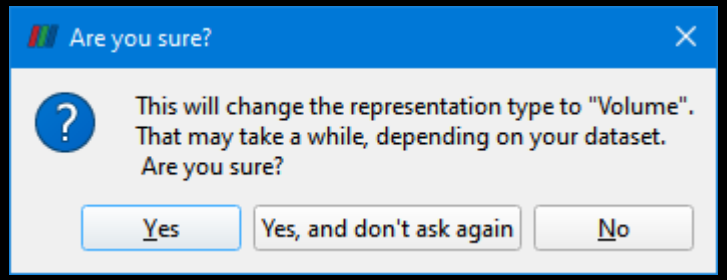
Add 10 more iso surfaces in the visualisation. Add Axes to the plot.

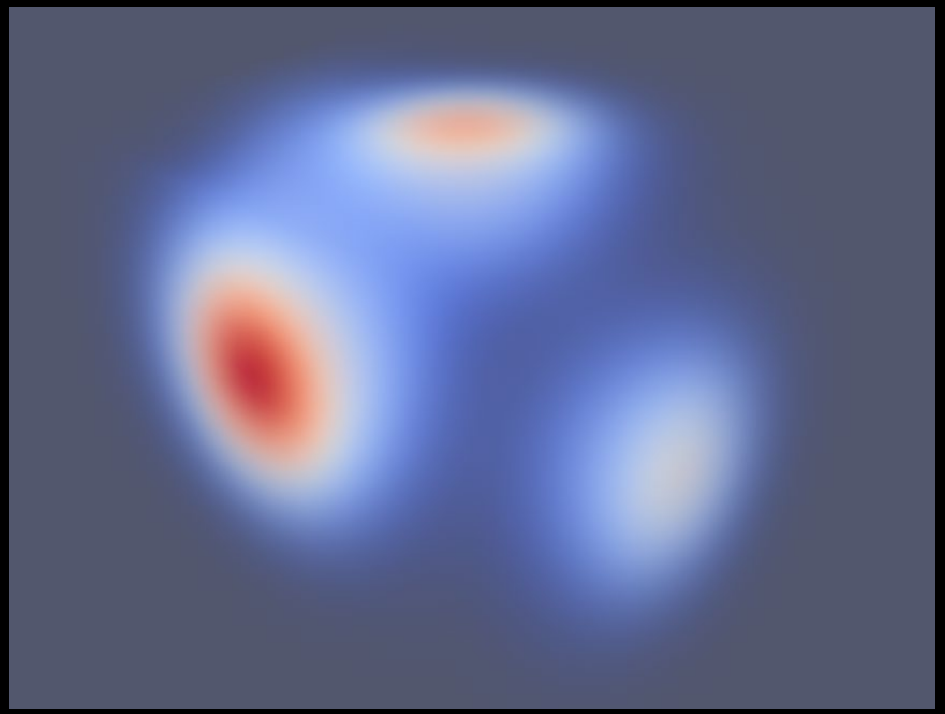
Displaying data as a volume

Select TableToStructuredGrid1 and disable the other filters



Set representation to ”Volume”

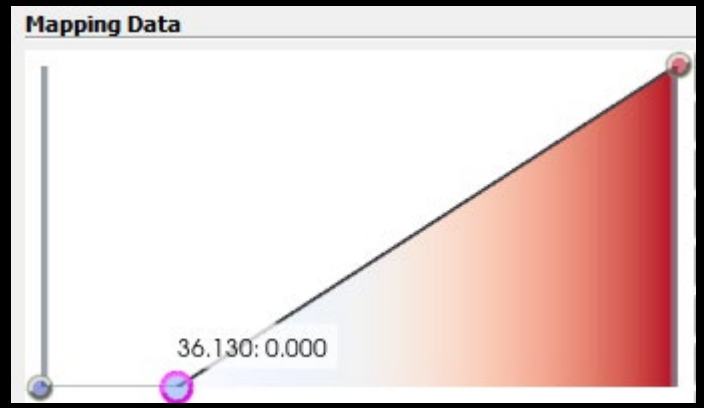




Controlling the opacity properties

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

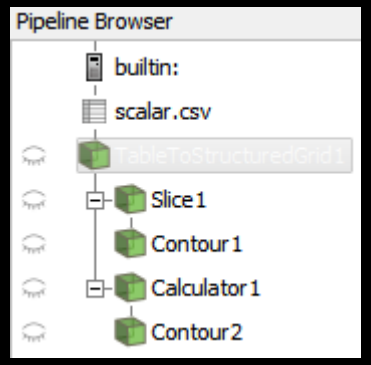
Click 



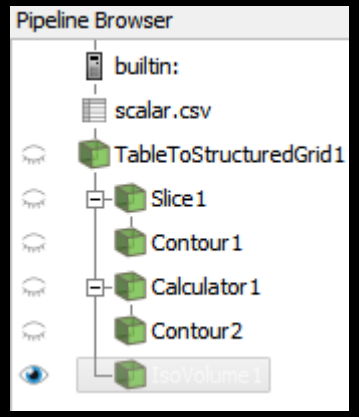
Add a point by clicking on the curve.
Drag the point down to make values left of the point transparent

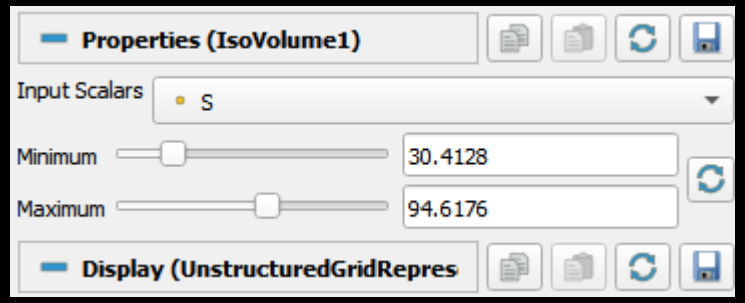
Creating iso volumes

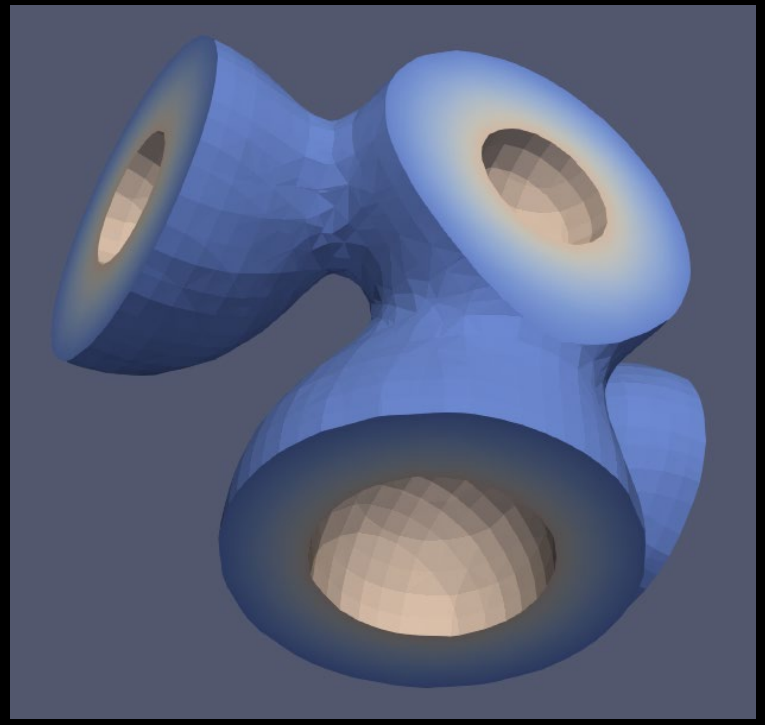
Select TableToStructuredGrid1 and disable the other filters



Add a IsoVolume filter from the menu or searching for it

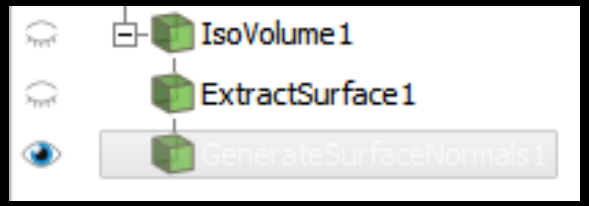


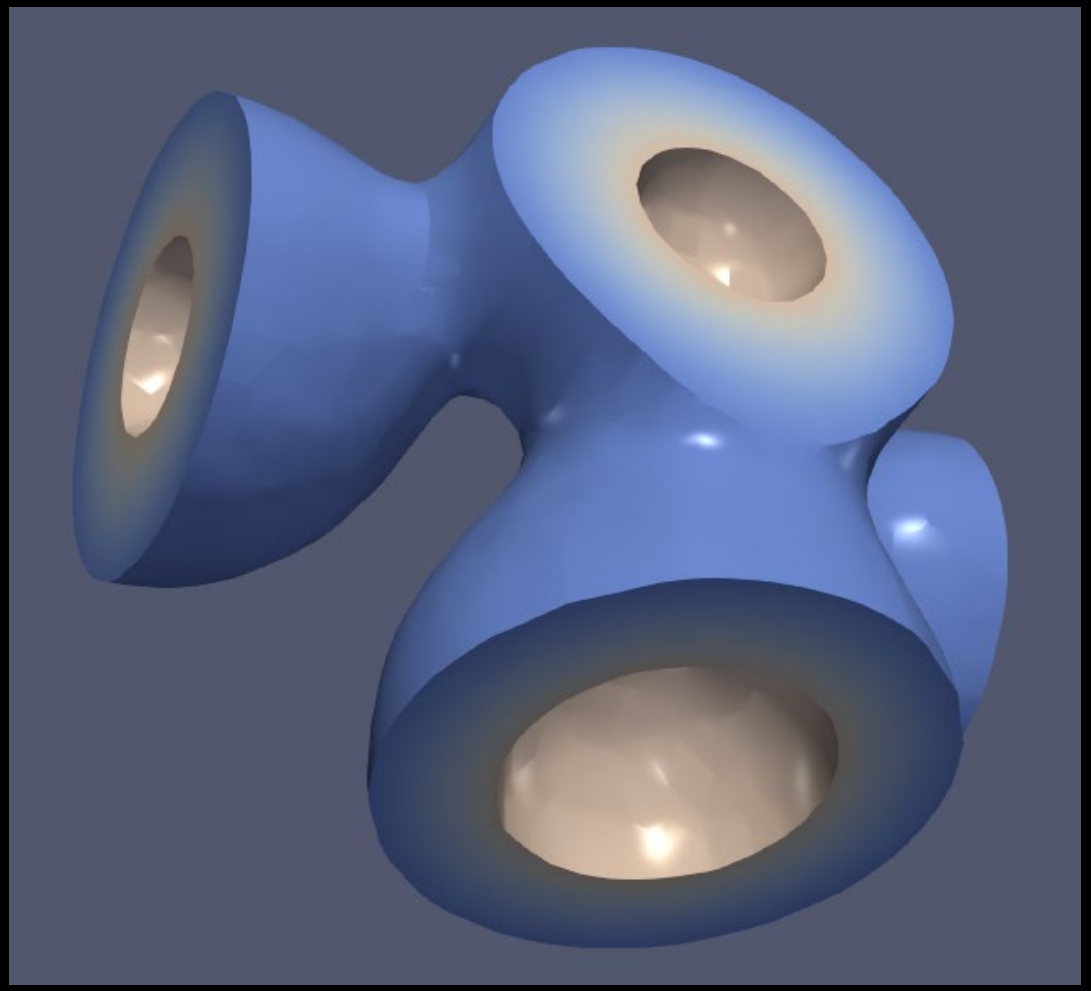




Creating smooth iso volumes

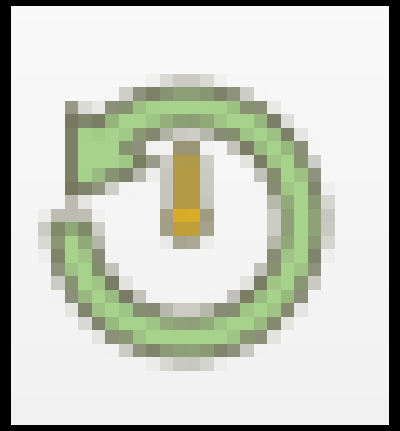
Add ExtractSurface and a GenerateNormals filter to the IsoVolume filter






Demo time

Reset ParaView

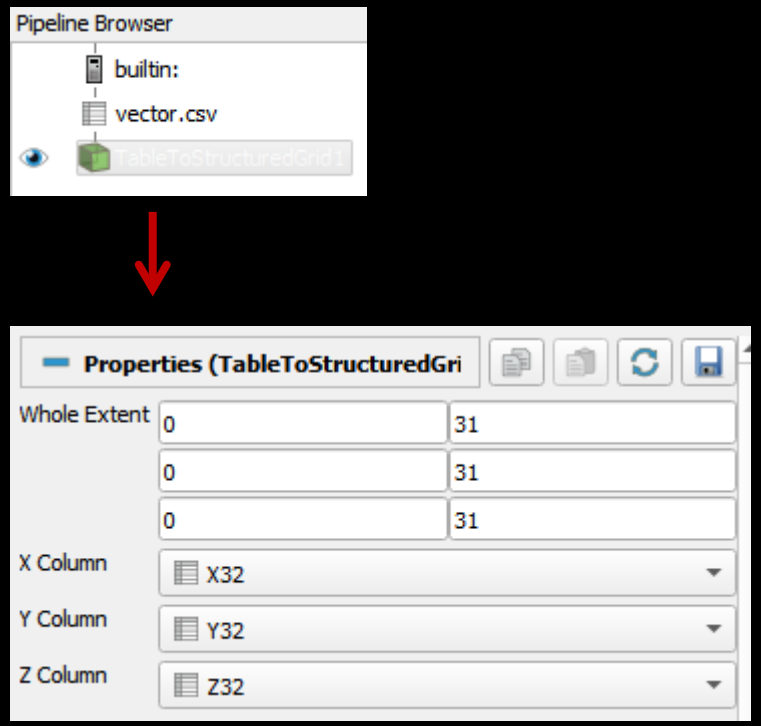


Reading vector data

Open the vector.csv file

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

Create a Grid from the data

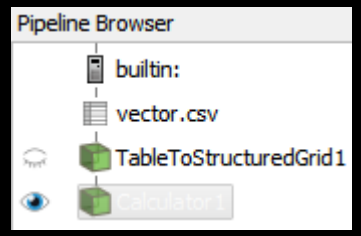


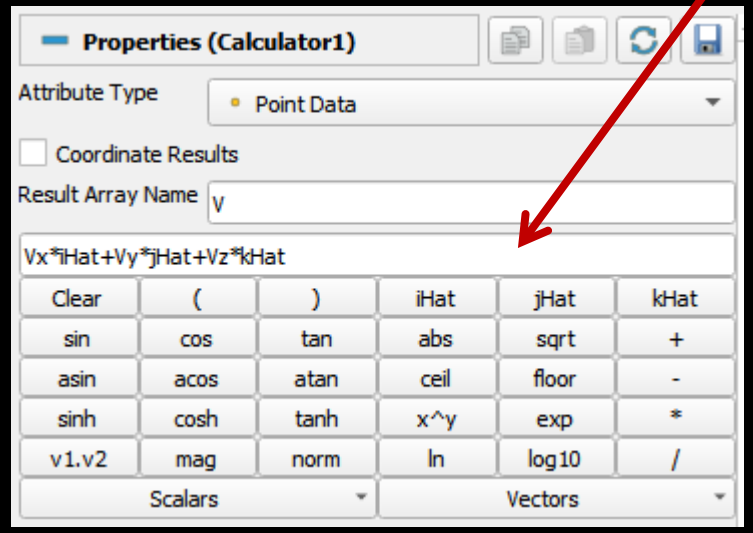
Vector data from the CSV file is provided as 3 scalar fields.

We need to convert these to vectors

Calculator the the rescue

Combine scalar fields to vectors





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

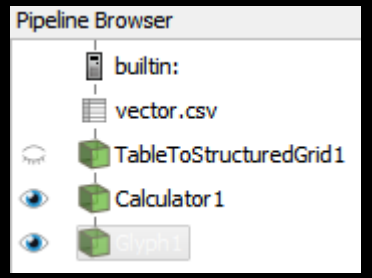
We now have a vector field we can visualise. We start with glyphs

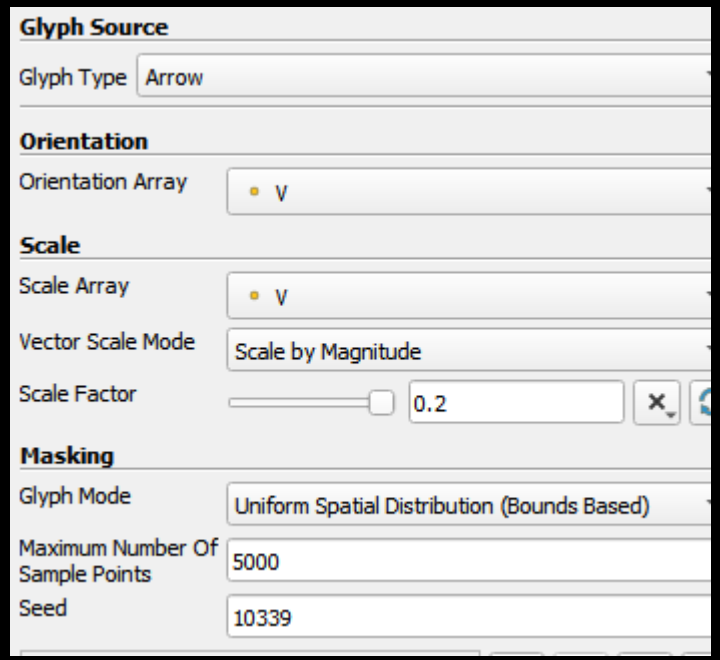

Demo time

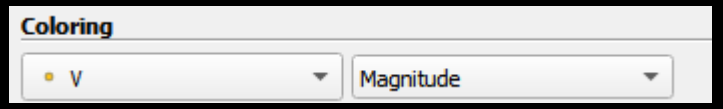

Using filters with vector data

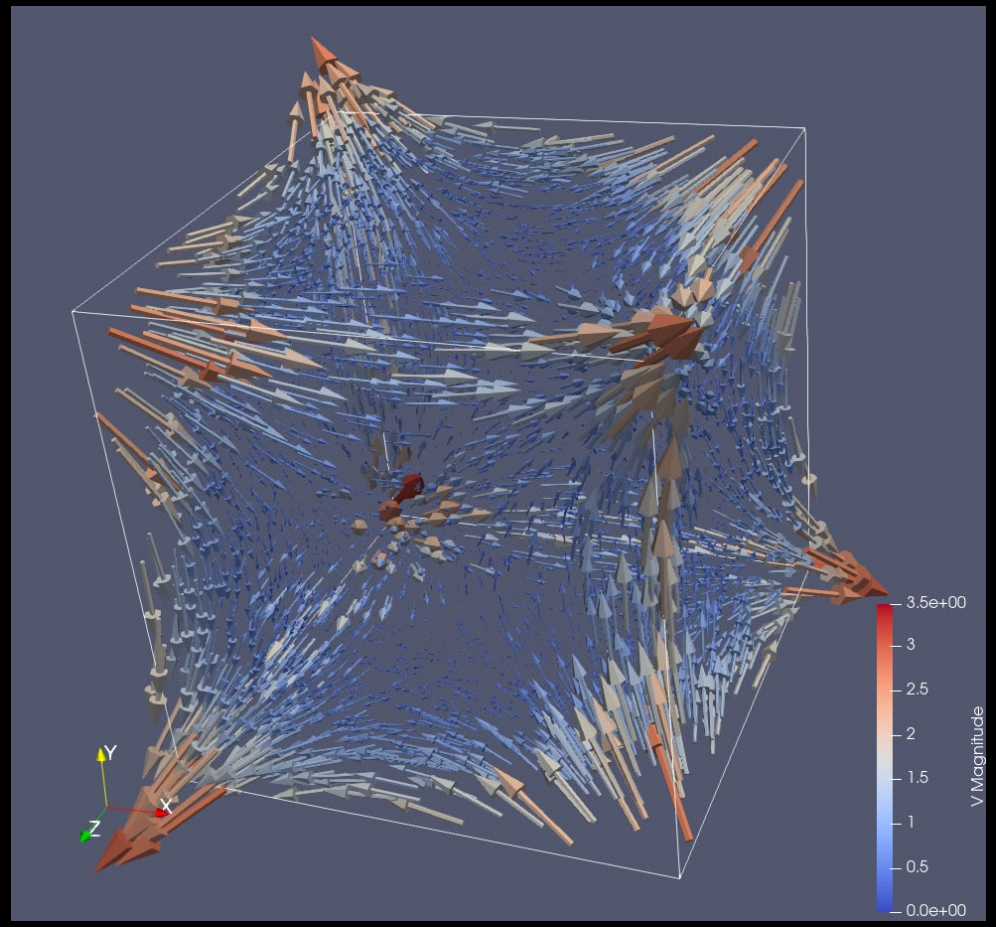
Visualise vectors using oriented glyps

Add a Glyph to the **Calculator**





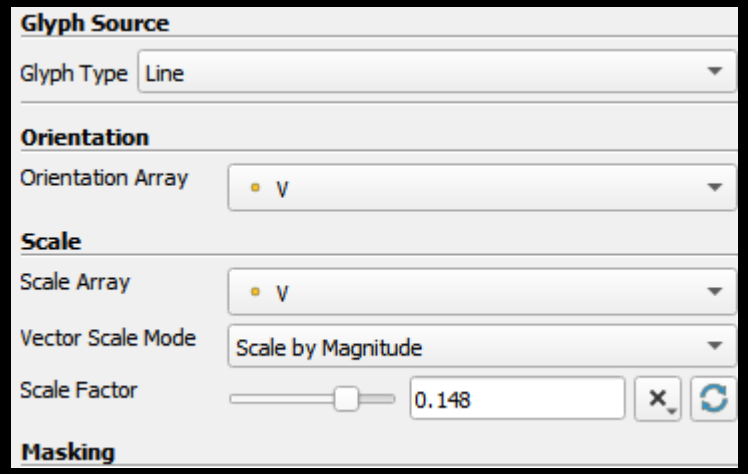


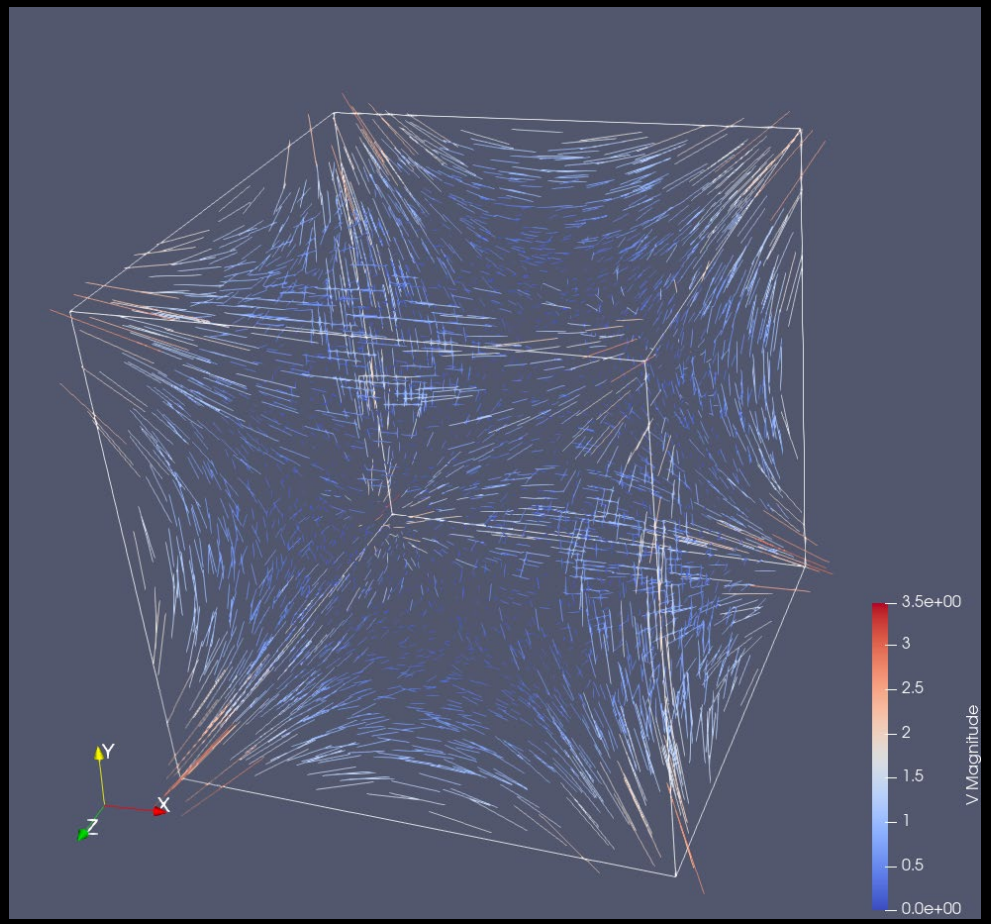


Visualise vectors using oriented glyps

Sometimes it can be beneficial to visualise glyps as thin lines

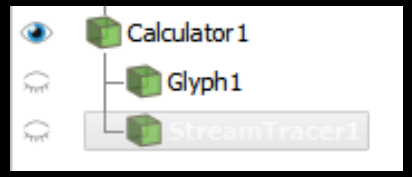
Change representation to ”Line”

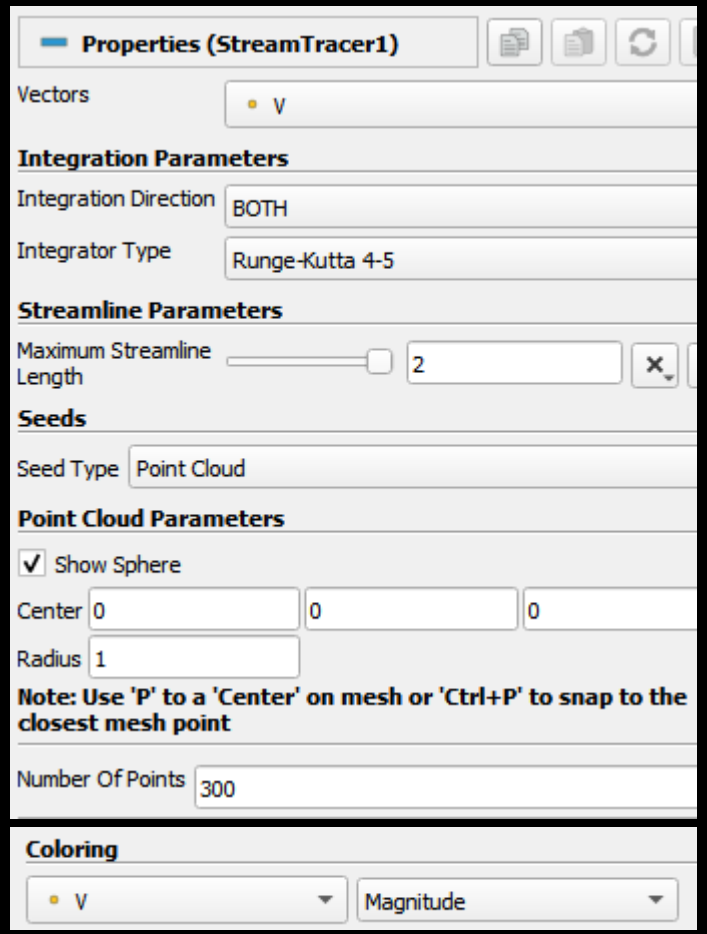


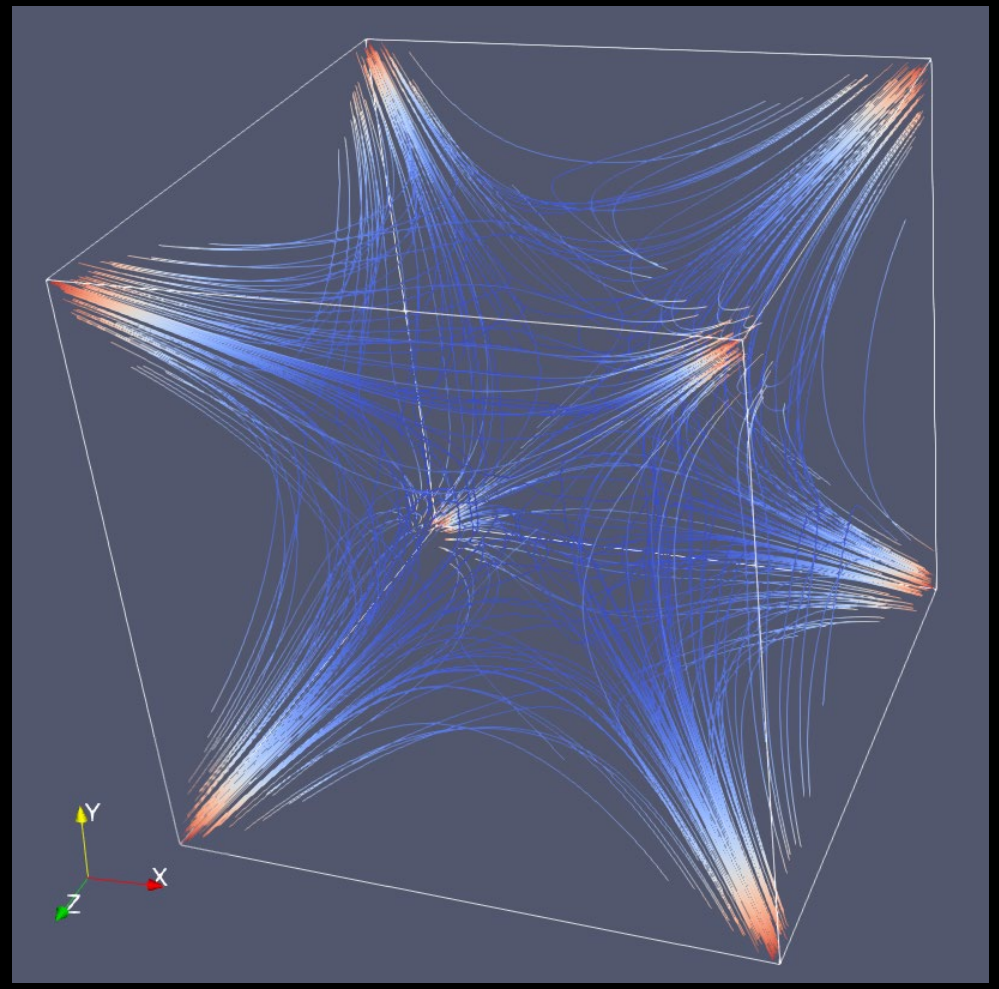


Visualise vectors stream tracer

Add a StreamTracer to the Calculator filter

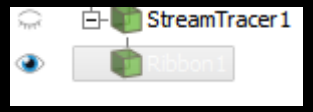


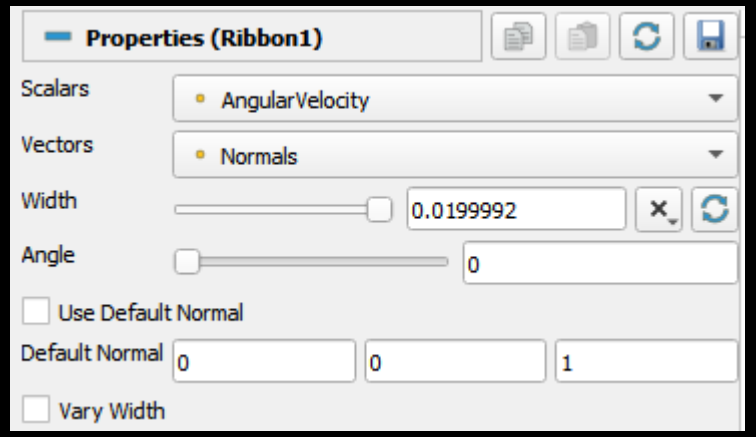


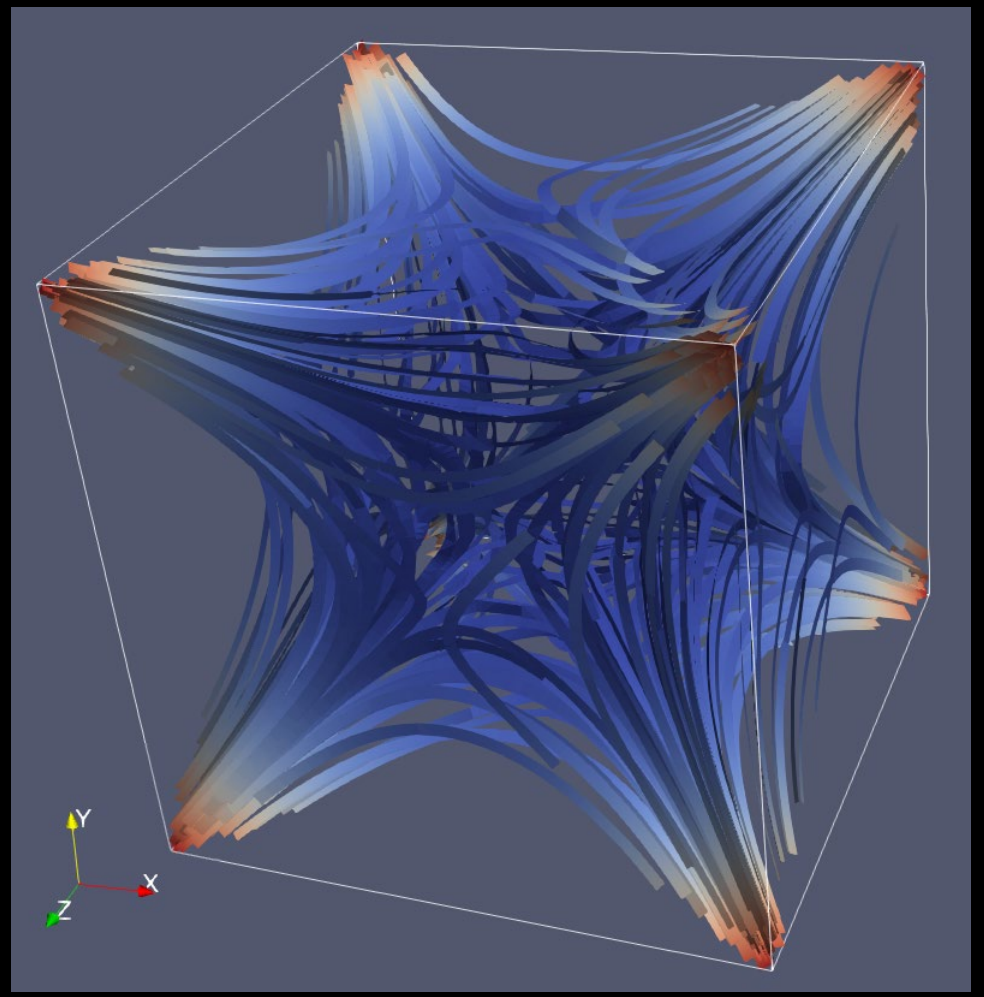


Visualise vectors as ribbons

Add a Ribbon filter to the StreamTracer filter

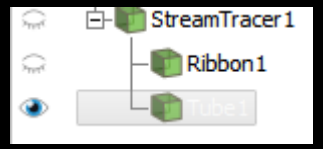


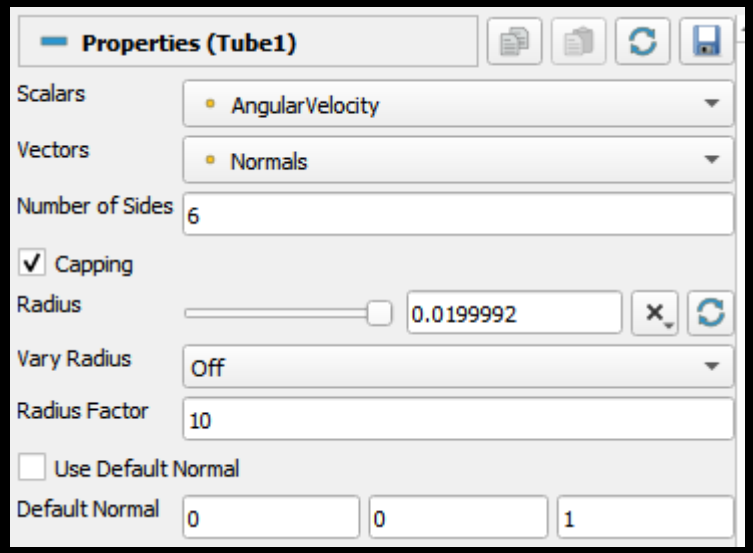


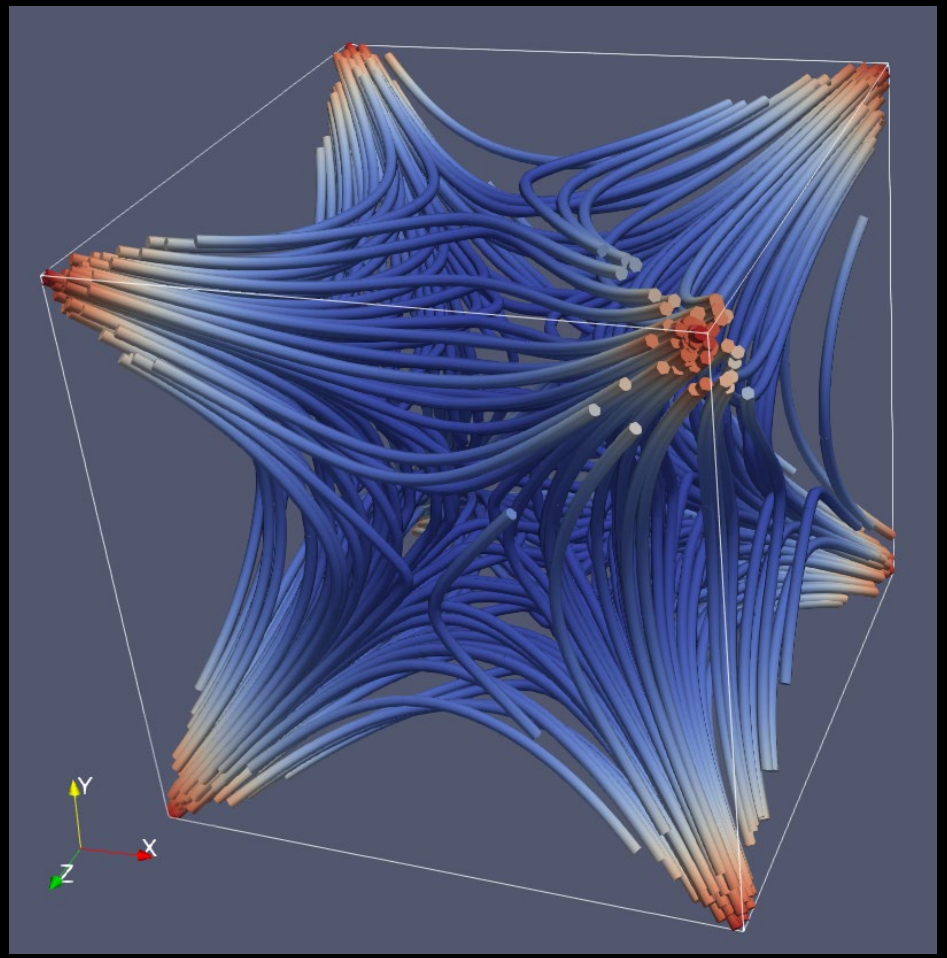


Visualise vectors as tubes

Add a **Tube** filter to the **StreamTracer** filter

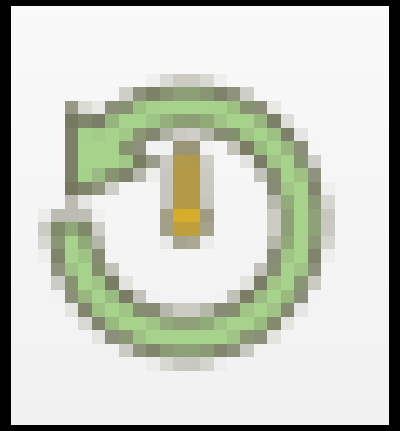







Demo time

Reset ParaView



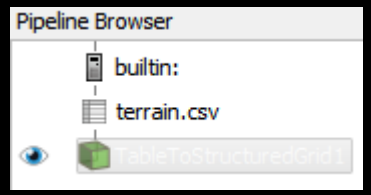
Reading terrain data

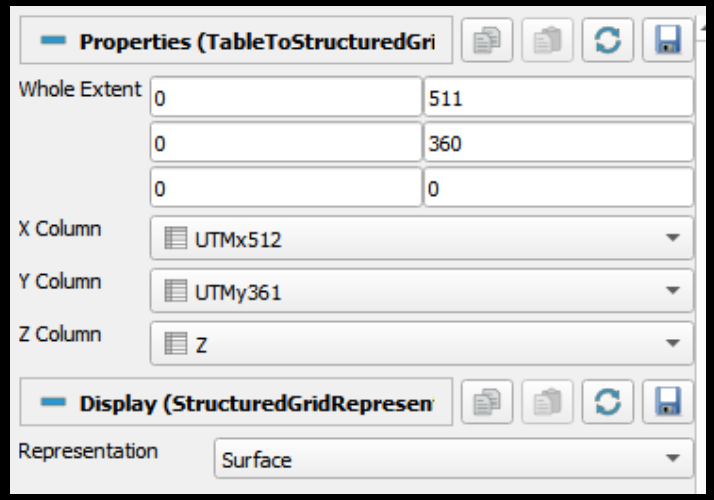
Open the terrain.csv file

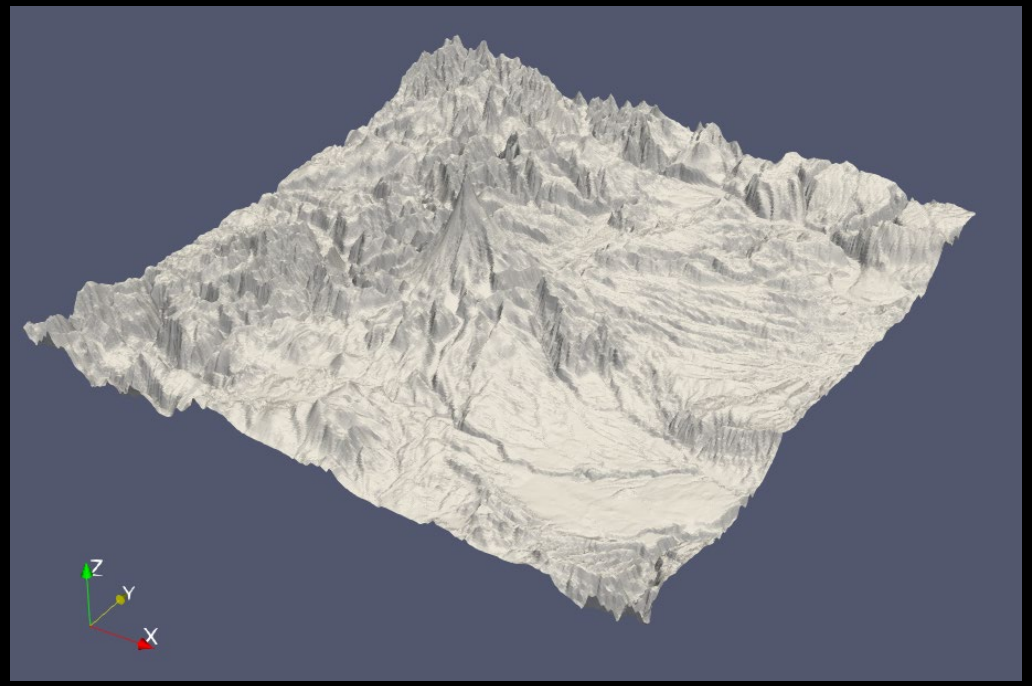
- File/Open...
- Find the terrain.csv file.
- Click Apply to load the file

Visualising terrain data

Create a 2D grid from terrain data



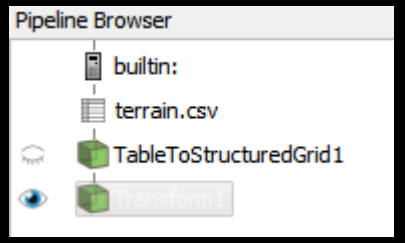


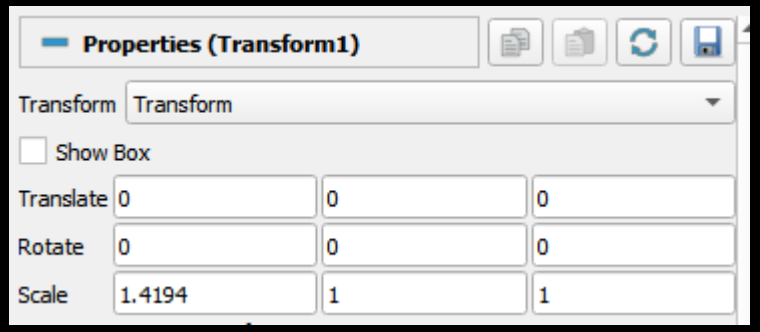


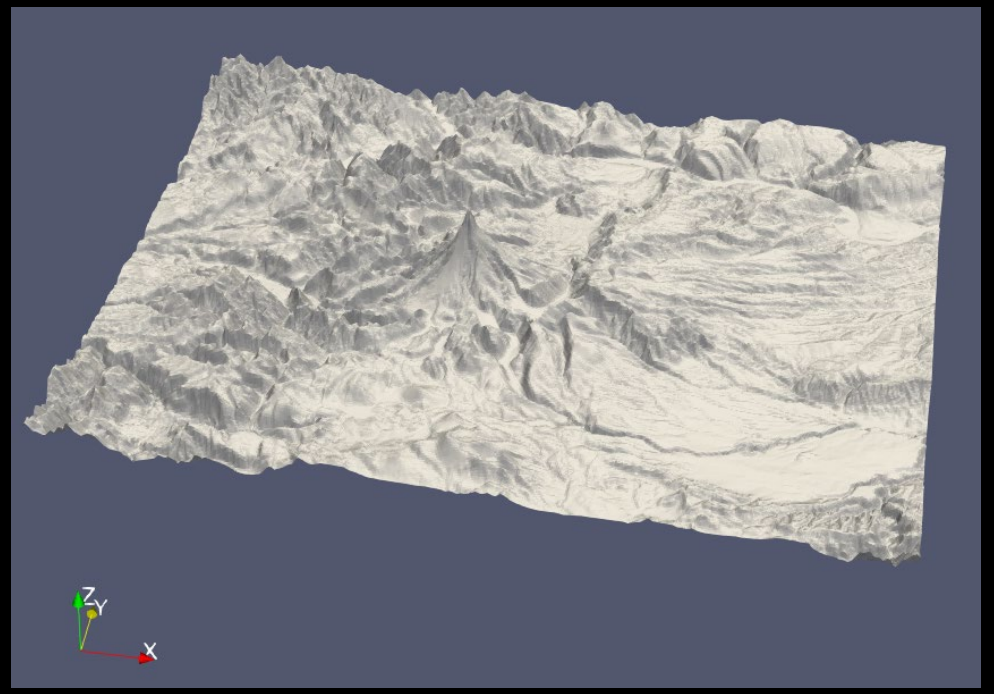
Adjusting scale factor

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

Add a Transform filter to the grid








Demo time

Exercise

1. Add a color scale for the elevation.
2. Add contours to the terrain.
3. Visualise them as black lines. Reduce the height exaggeration (Transform/Calculator)

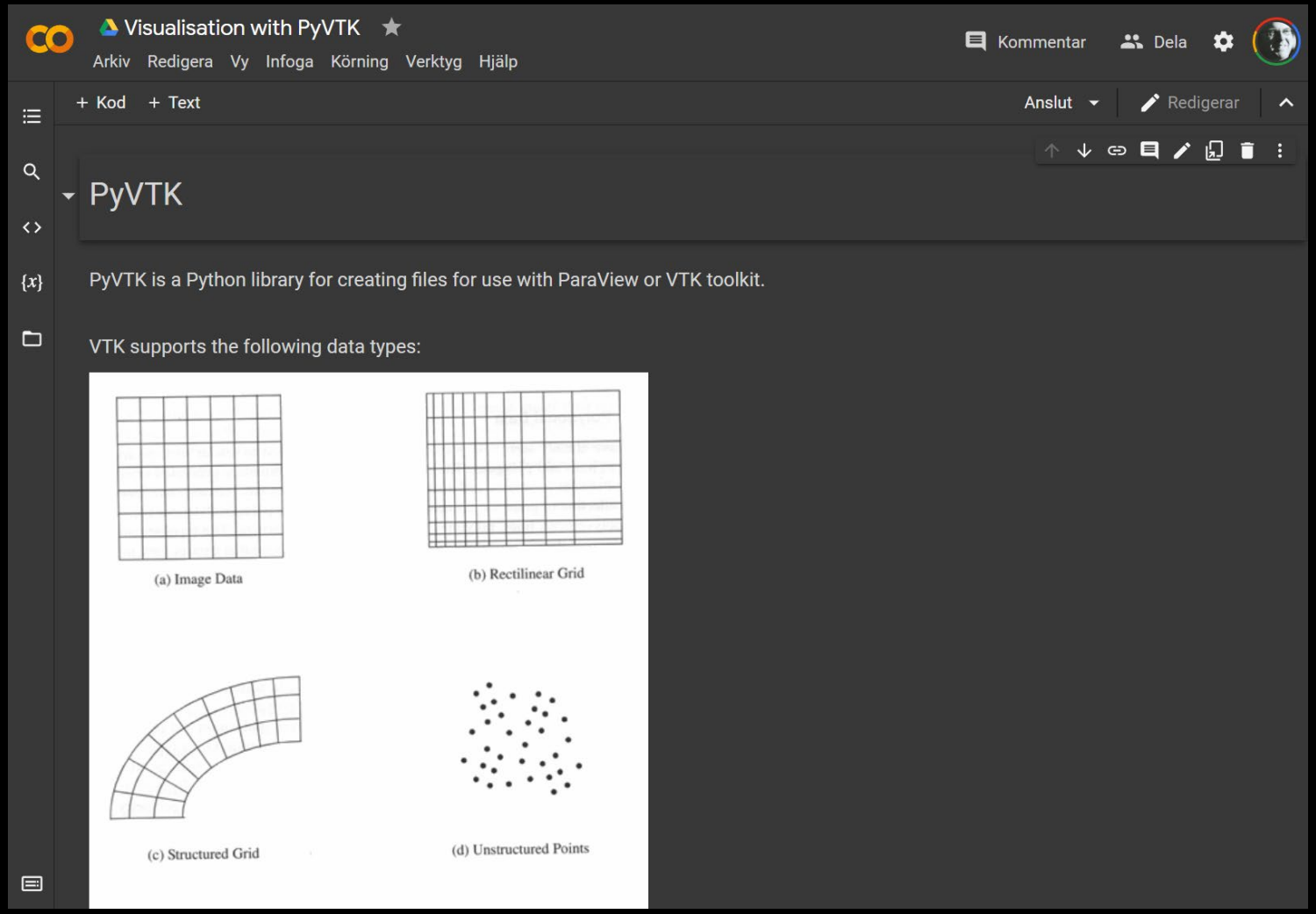
Getting data into ParaView

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.

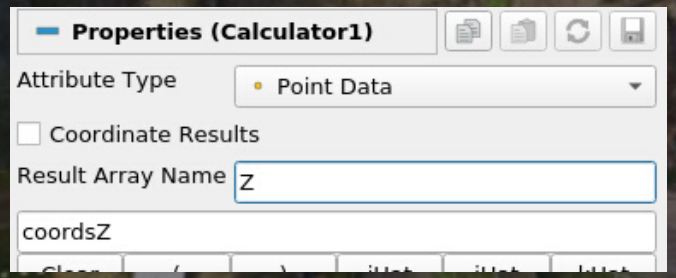
PyVTK Notebook



Exercises with real data

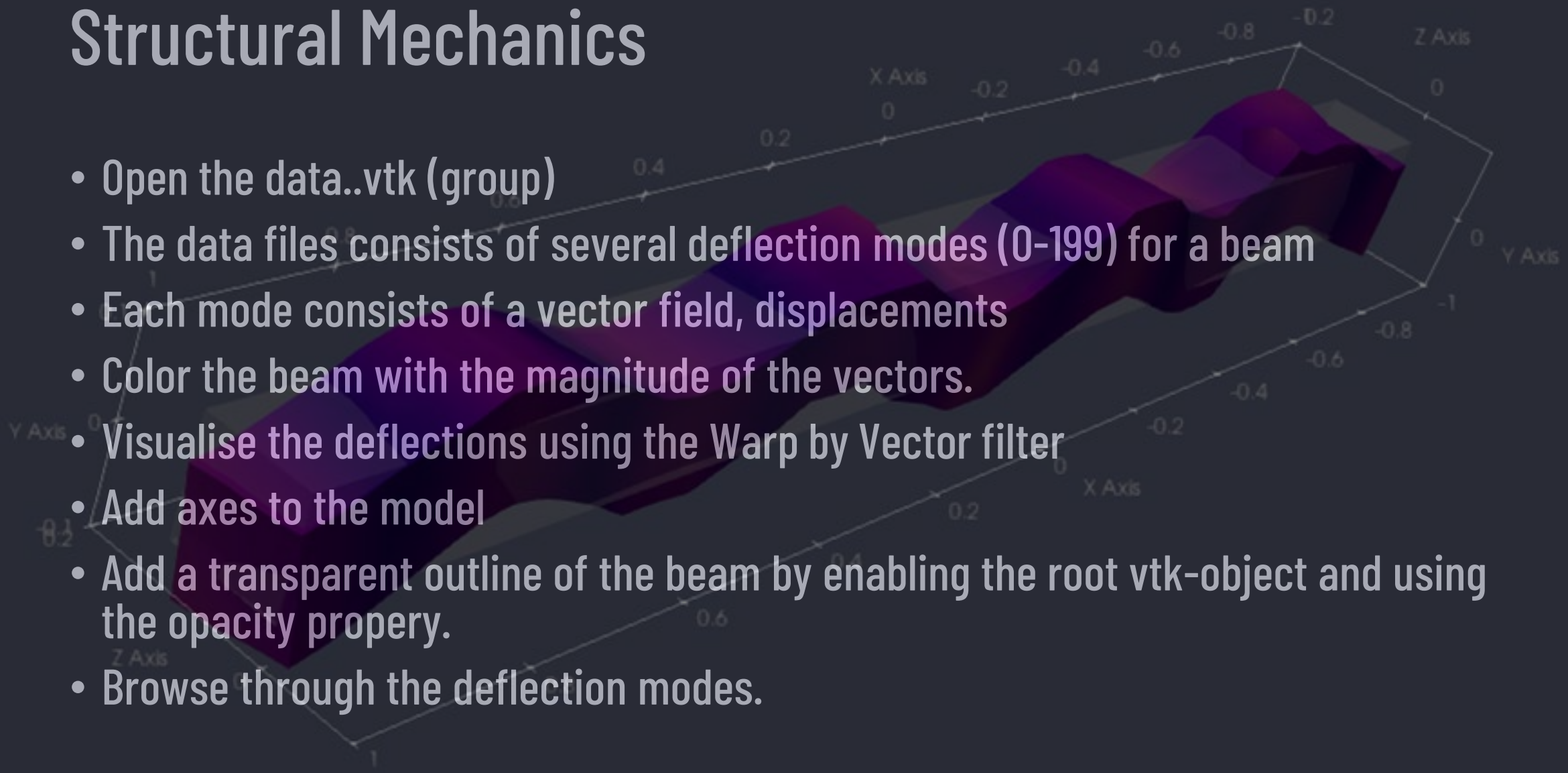
Photogrammetry

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



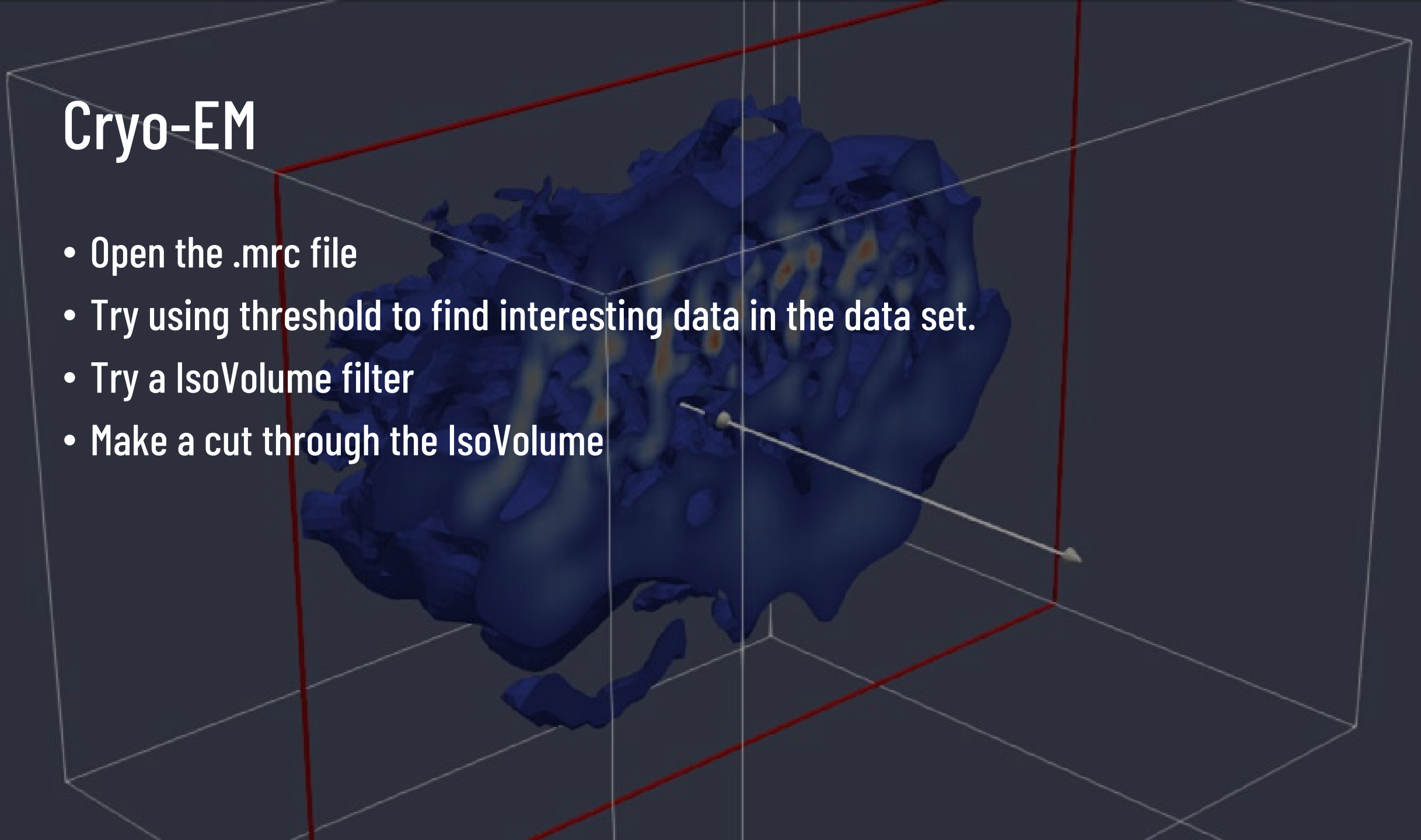
- Visualise elevation data using a suitable color scale
- Add black contour lines with a interval of 1 m

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 propery.
- Browse through the deflection modes.

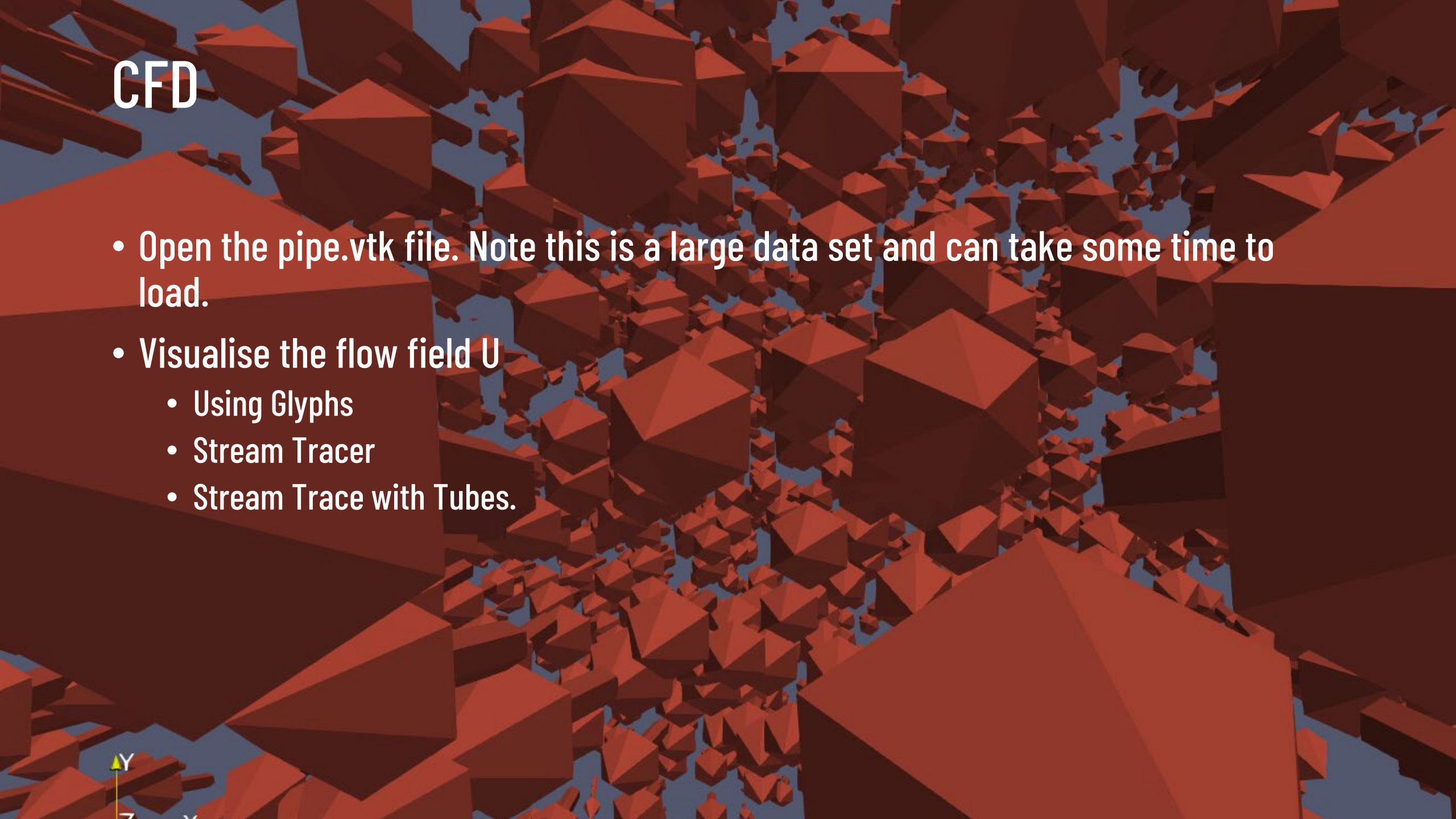
Cryo-EM



- 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

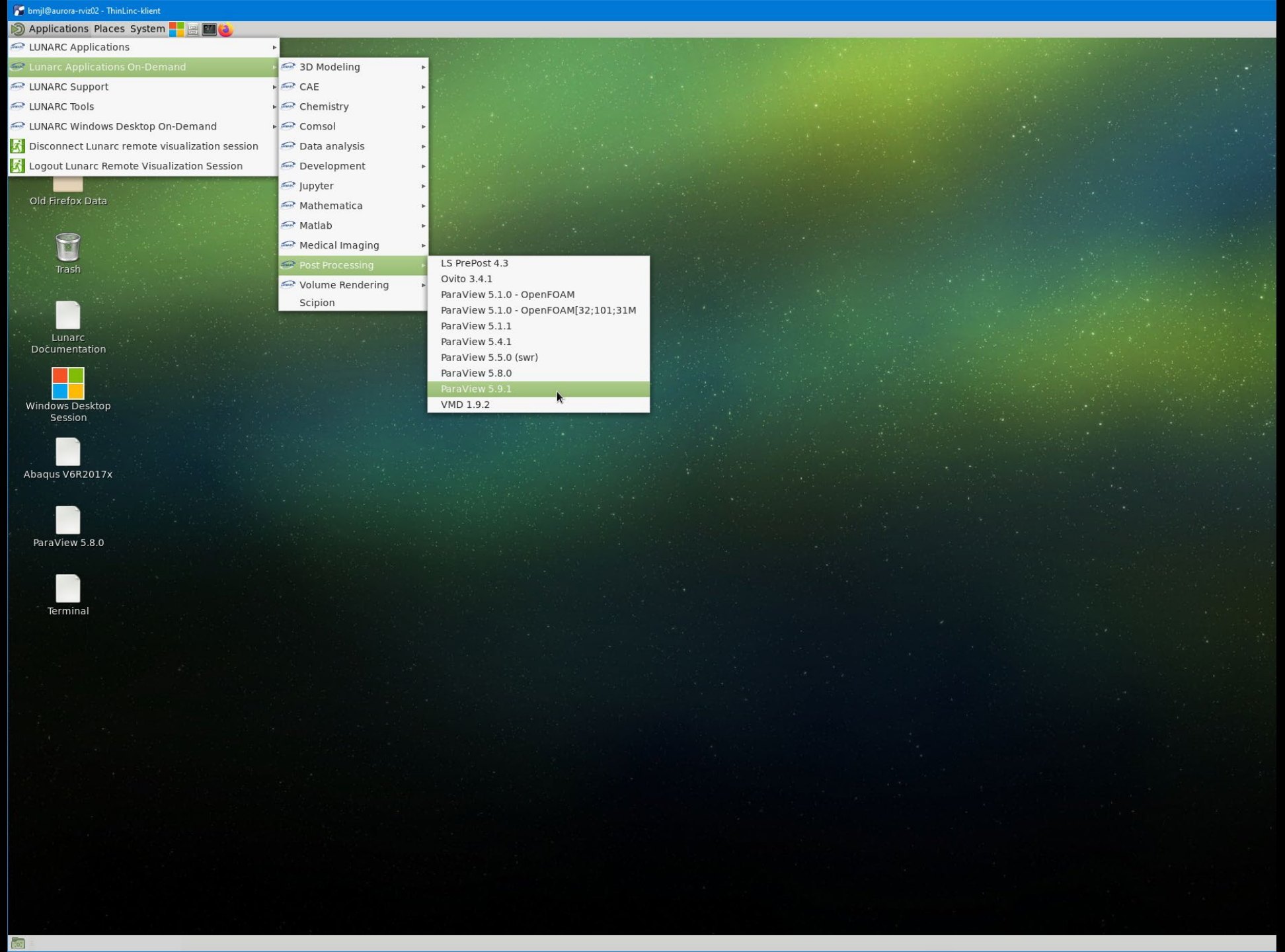
CFD

- 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.

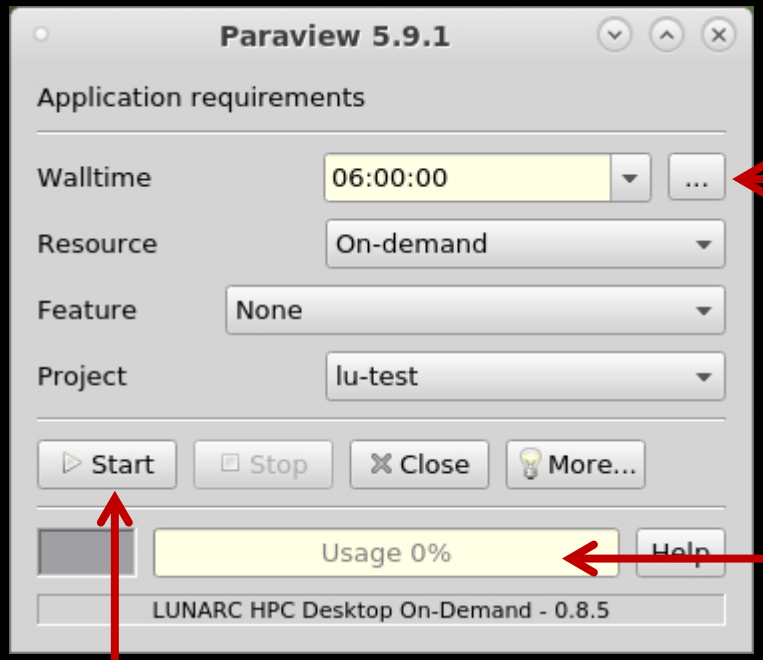


Creating Visualisation Apps with Trame

Starting ParaView on COSMOS


bmjl@aurora-rviz02 - ThinLinc-klient
Applications Places System
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
3D Modeling
CAE
Chemistry
Comsol
Data analysis
Development
Jupyter
Mathematica
Matlab
Medical Imaging
Post Processing
Volume Rendering
Scipion
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
VMD 1.9.2
Old Firefox Data
Trash
Lunarc Documentation
Windows Desktop Session
Abaqus V6R2017x
ParaView 5.8.0
Terminal

ParaView from LUNARC On-Demand

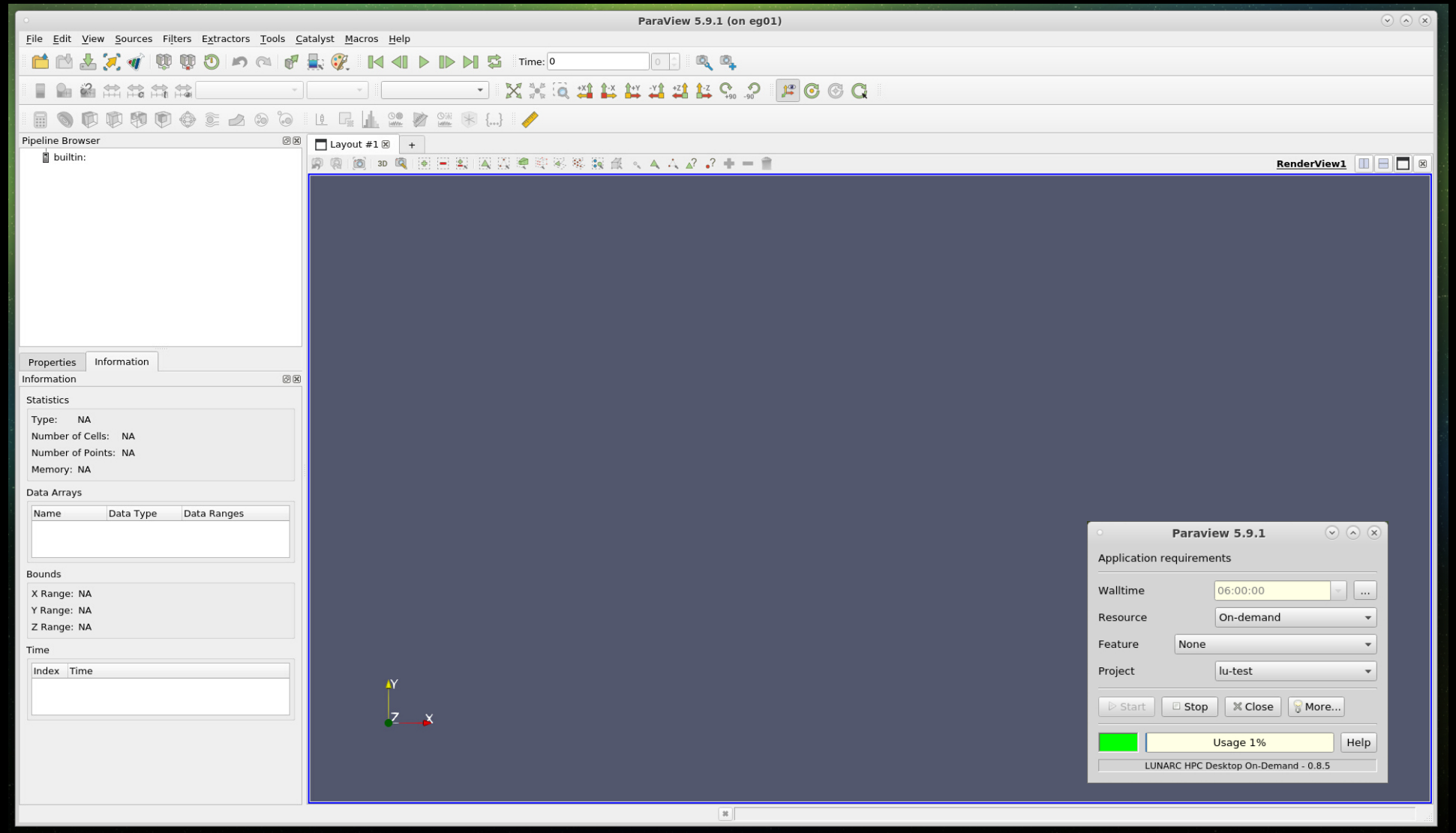


Give an estimate of how long you will need ParaView

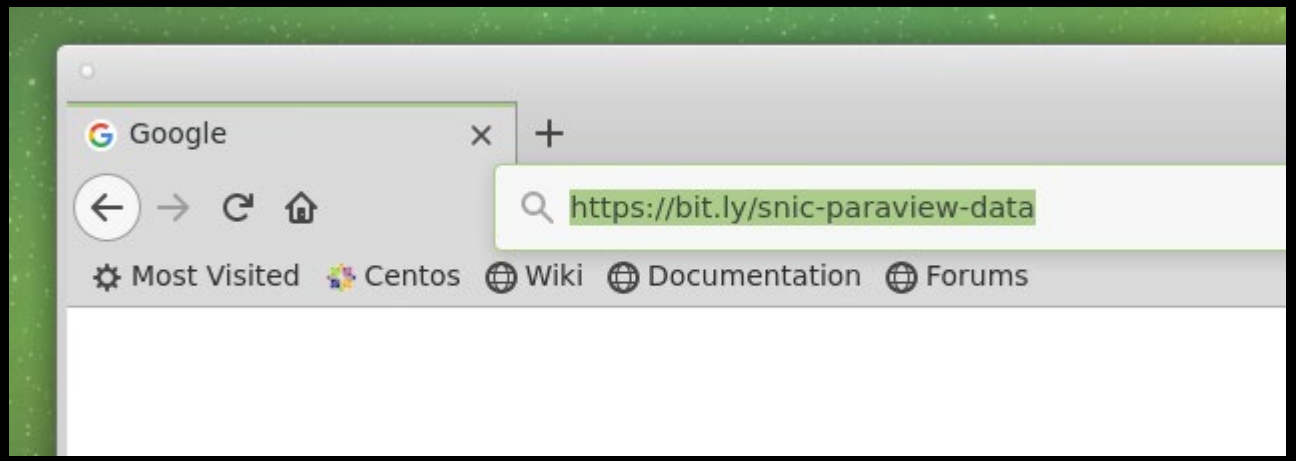
Shows how much of the allocation have been used.

Start ParaView here

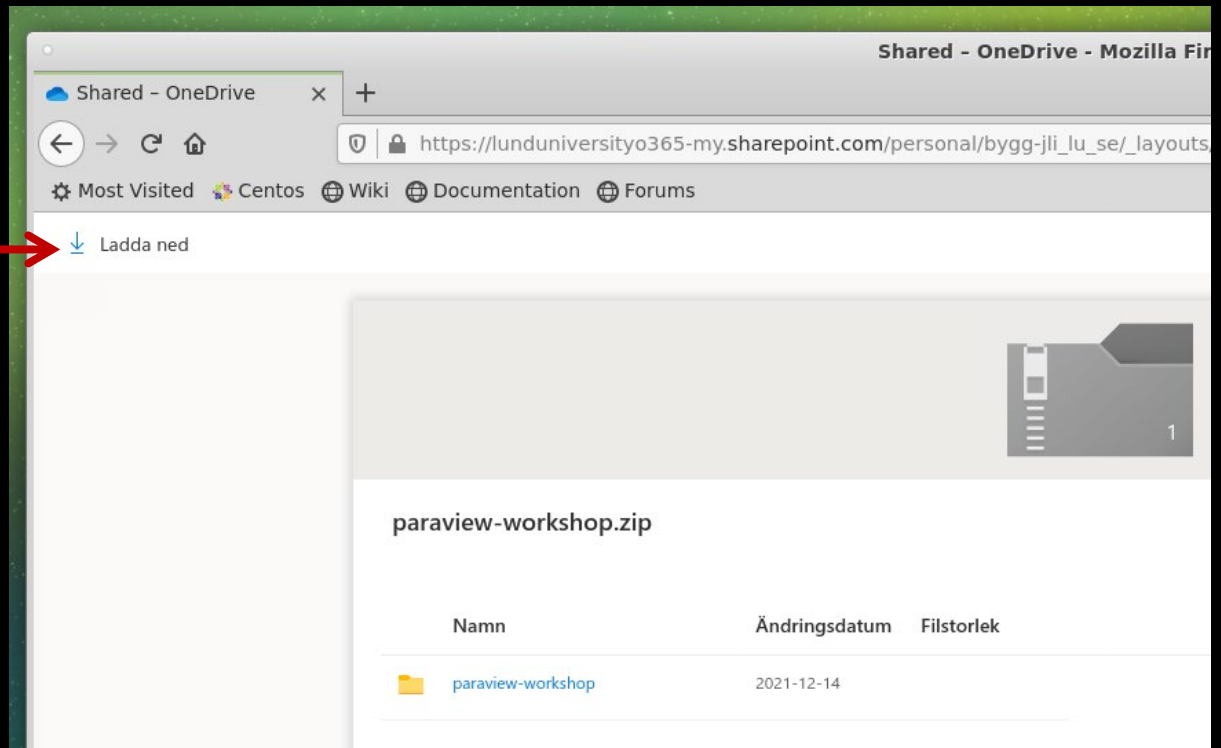
ParaView running on Aurora



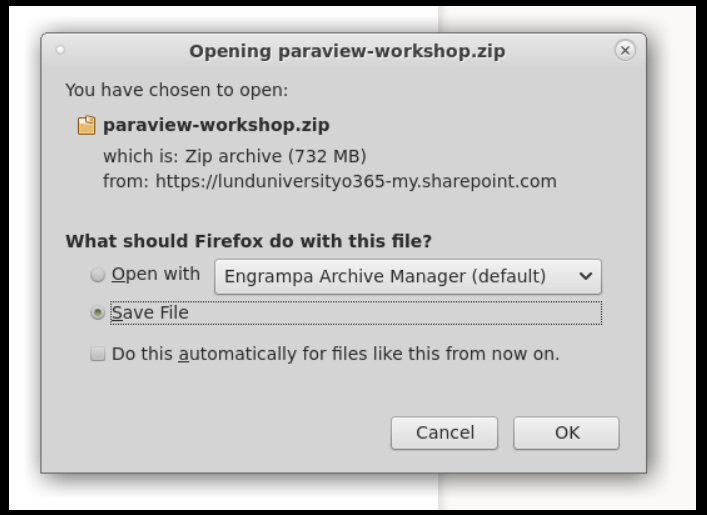
Downloading data to Aurora

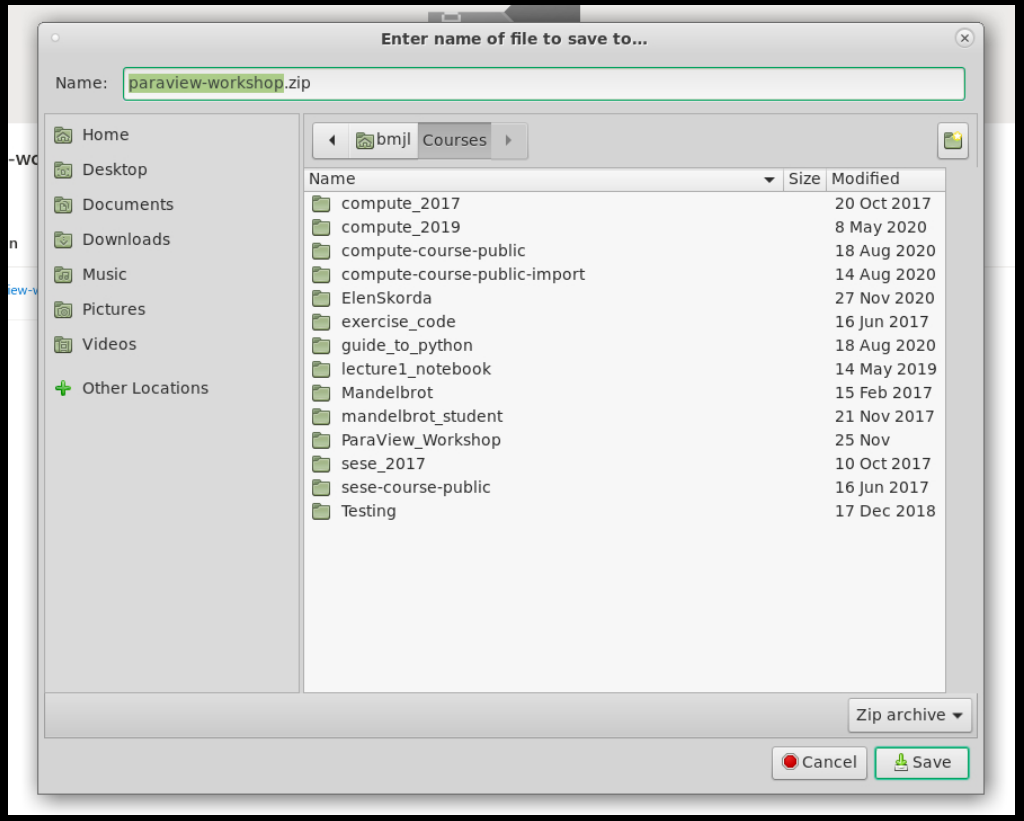


Download here



Downloading data to Aurora





Extracting data set