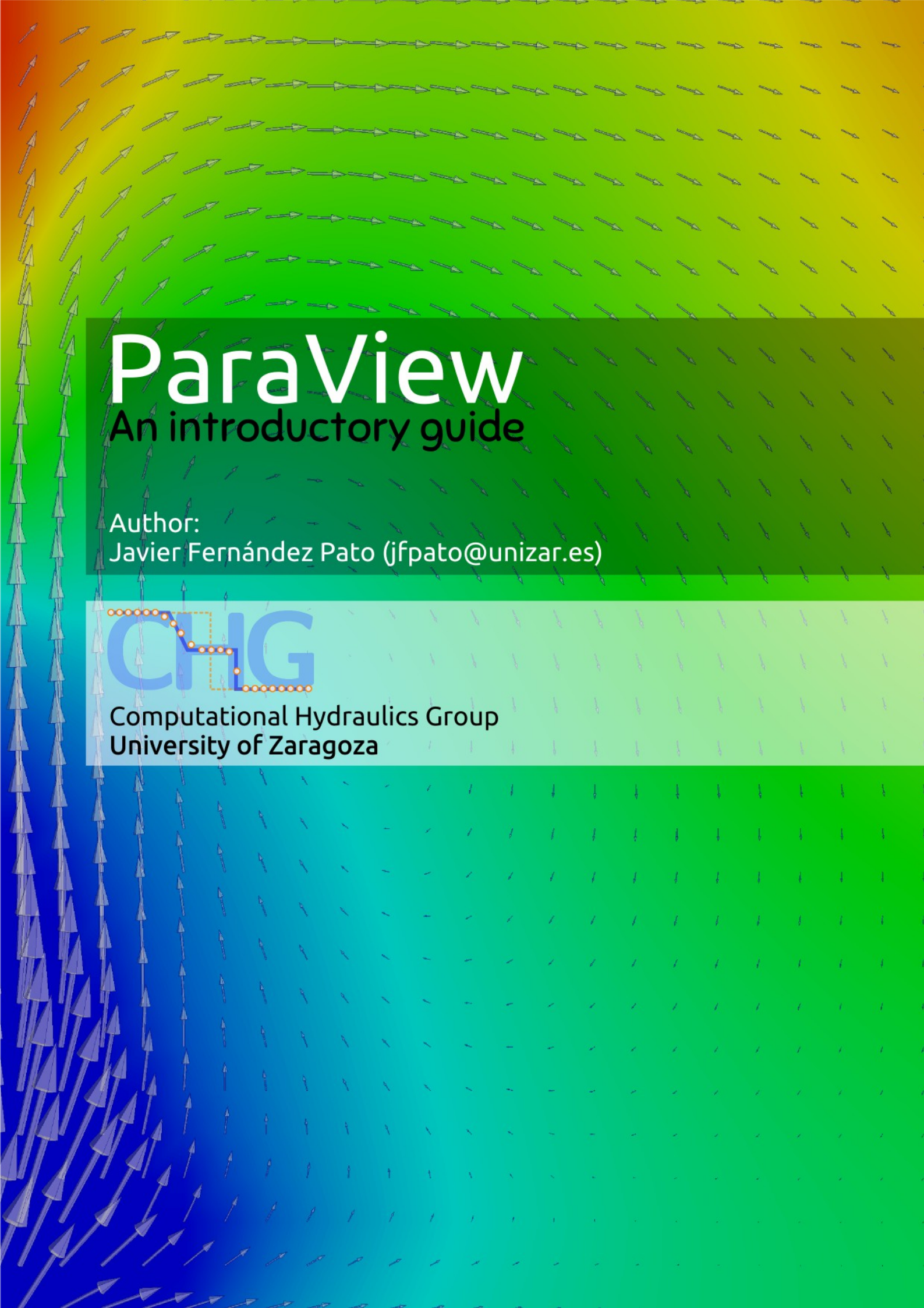


ParaView
An introductory guide

Author:
Javier Fernández Pato (jfpato@unizar.es)

Computational Hydraulics Group
University of Zaragoza

Contents

1. Loading data and customizing the color scale
    1.1. User interface
    1.2 Toolbar components
    1.3 Loading data
    1.4 Color scale and key features
    1.5 Adjusting data range
2. Creating velocity vector fields
3. Plotting 1D information

1 Loading data and customizing the color scale

Note: This tutorial is based off of ParaView version 5.2. The reader can download a copy from www.paraview.org.

1.1 User interface

The user interface shown in Figure 1.1 is the default graphic configuration loaded when ParaView is started. Every item of the default layout is configurable and can be moved around. The main components are listed below:

- **Menu bar:** Allows the access to the program features and commands.
- **Toolbars:** Provides a quick access to the most common features of ParaView.
- **Pipeline browser:** Allows to view and select pipeline objects.
- **Properties and information panels:** When a pipeline is selected in the pipeline browser, this panel allows to view and change its parameters.
- **Render view:** This is the layout where the data is visualized. It can be subdivided in several views in order to display diferent kind of graphic information: 2D and 3D views, plots over time, etc.

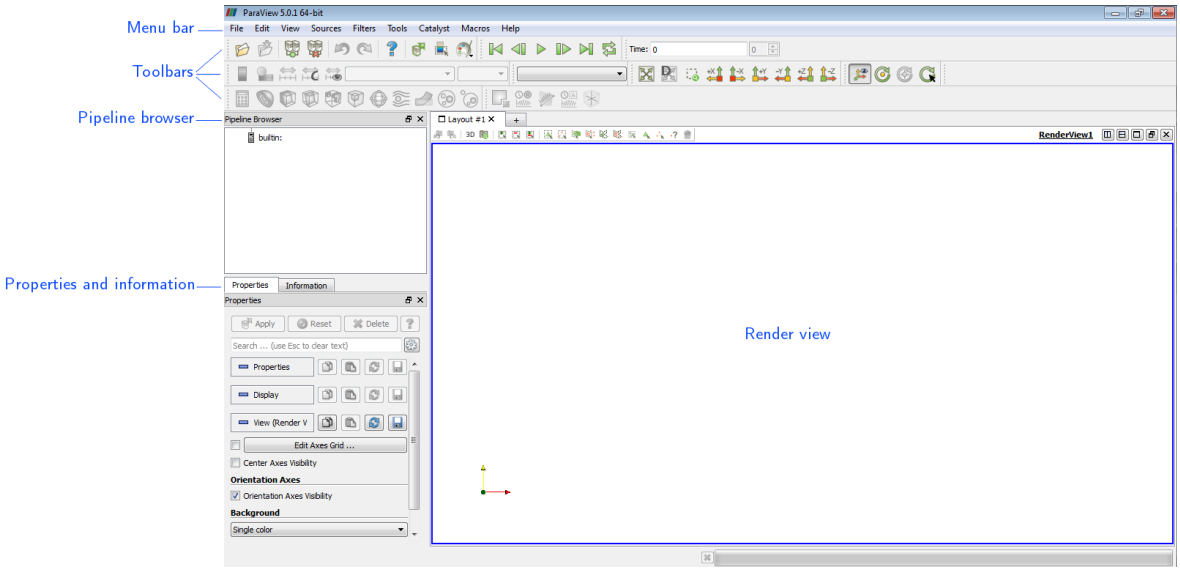


Figure 1.1 Default user interface.

1.2 Toolbar components

The main toolbar of ParaView represents an easy way to access the most commonly used features (see Figure 1.2). In this section, a brief description of some of its components (the trivial ones are omitted) is presented.

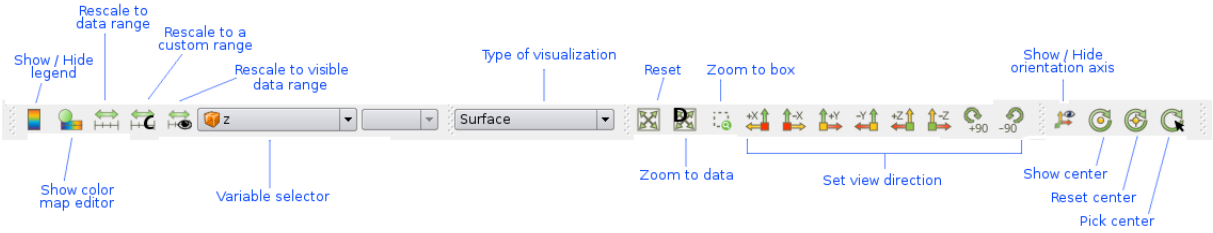


Figure 1.2 Toolbar components.

- **Rescale to data range:** Rescales the color using the data range (at the current time step) from the data source selected in the Pipeline browser i.e. the active source.
- **Rescale to custom range:** Rescales the color using a range provided by the user.
- **Rescale to visible data range:** Rescales the color using the data range visible in the layout, depending on the zoom level.
- **Variable selector:** All the variables presented in the VTK les are listed in this pull-down menu. The selected variable changes together with the active selection in the pipeline browser.
- **Type of visualization:** Changes among diferent visualization types. The most common are surface and surface with edges. Figure 1.3 shows the diference between this two types of visualizations.

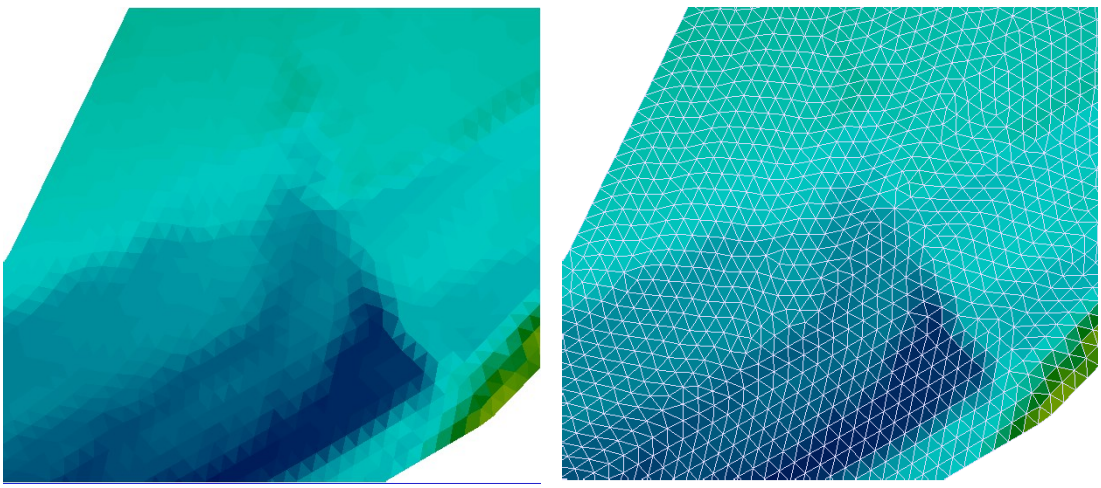
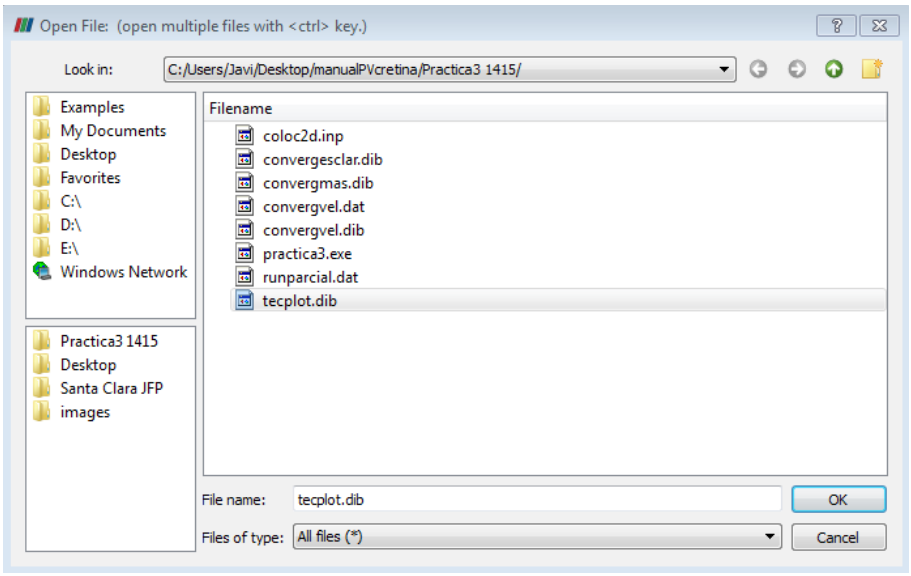


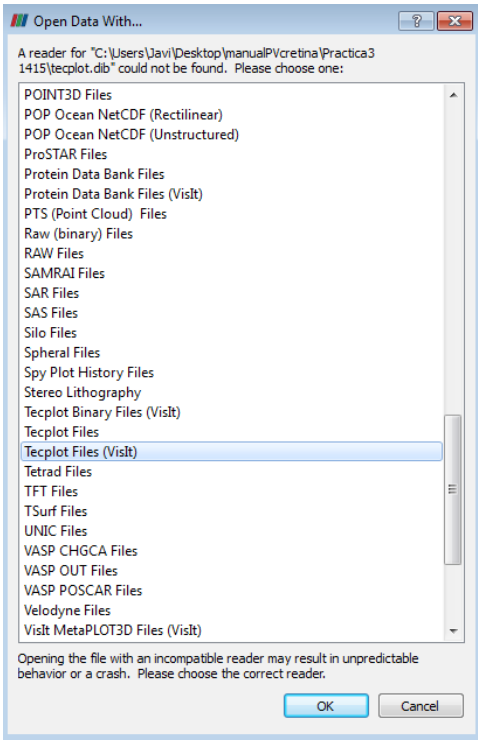
Figure 1.3 Surface (left) and surface with edges (right) visualization types.

1.3 Loading data

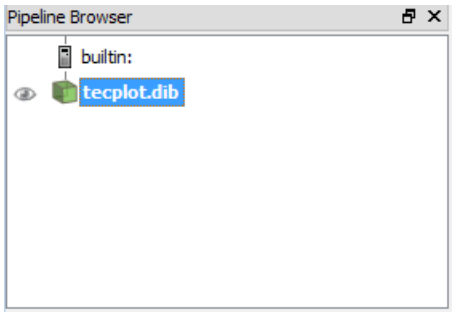
The most common file type associated with ParaView is the .vtk (Visualization ToolKit). ParaView can load a single .vtk file or a set of .vtk files named using certain patterns (for instance: data001, data002, ...). Usually, each of these files corresponds to diferent times of the simulation. The files can be loaded by selecting *Open* in the *File* menu. A file selector appears in which the whole set of .vtk files can be selected. Nevertheless, ParaView can load many other file types. For this example, the file named “tecplot.dib” is selected, as follows:



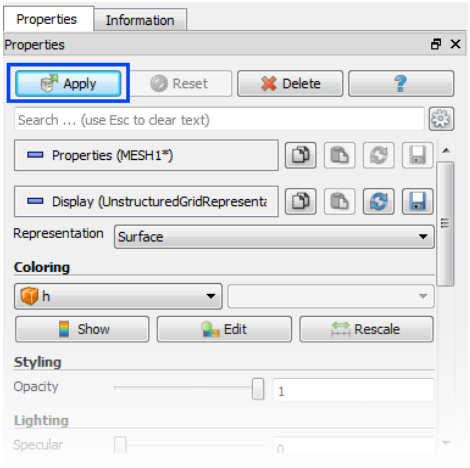
After clicking OK, the reader for these types of files should be chosen:



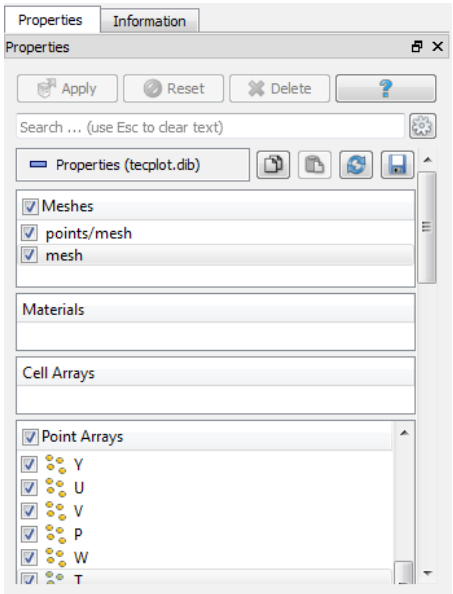
Once loaded, an item appears in the pipeline browser:



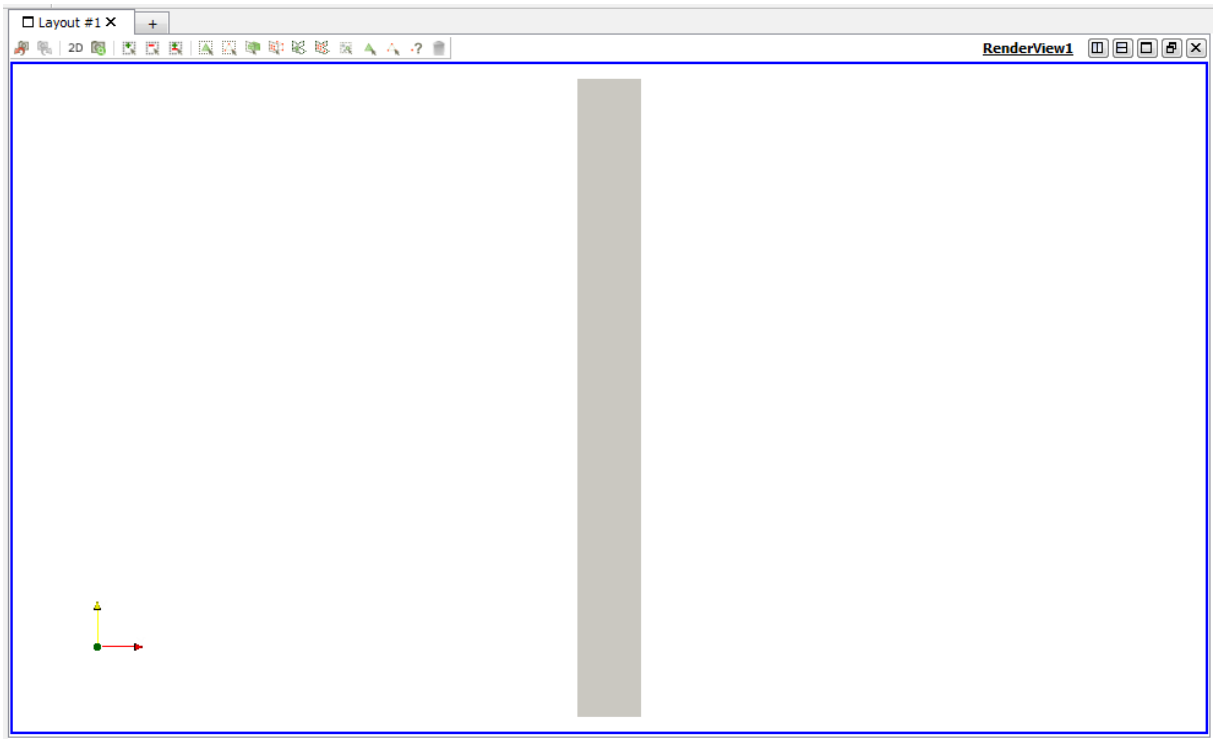
After loading the files, the user should click in the *Apply* button of the *Properties panel*:



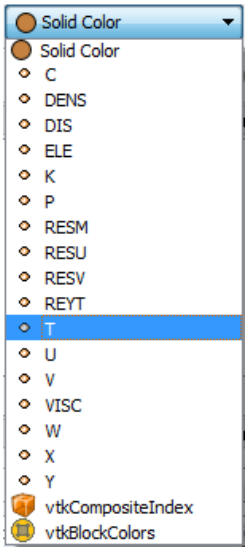
Now the user should choice the variables to load. For this example, the information is loaded over the mesh (click *mesh* box) and the variables of interest (check all the variables by clicking *Point Arrays*):



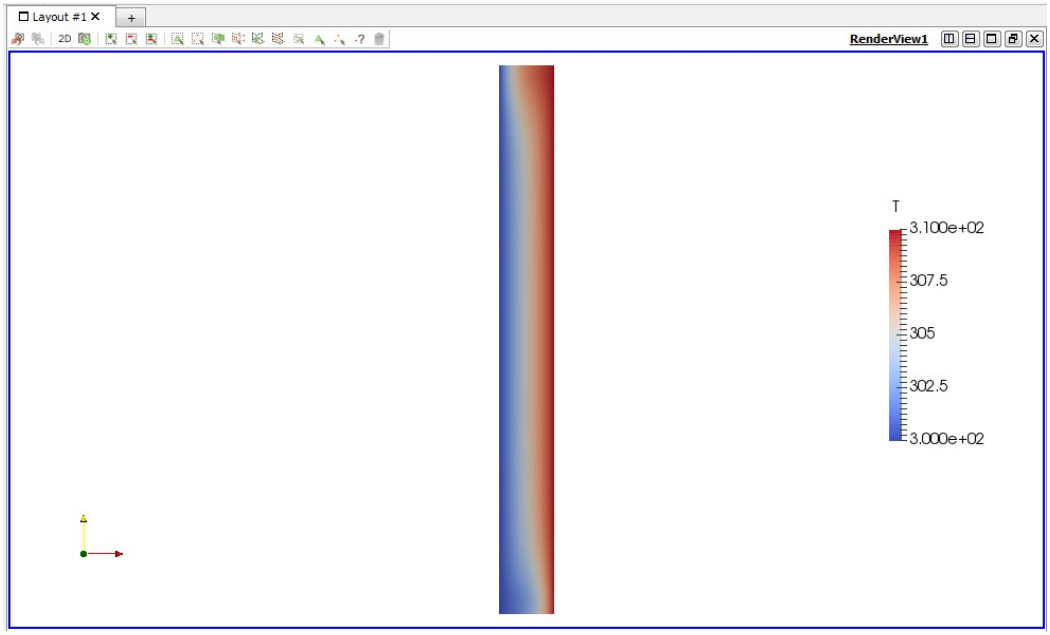
Now the geometric domain should appear in the *Render View* layout, as follows:



In order to visualize the temperature color map, this variable (T) should be chosen in the variable selector:



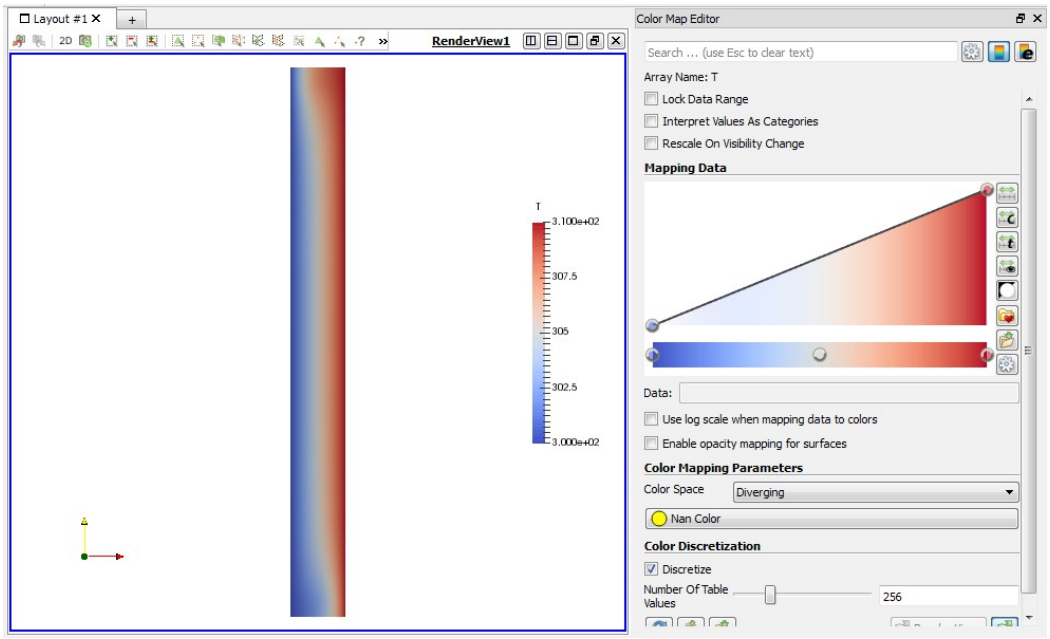
The color map should look as follows:



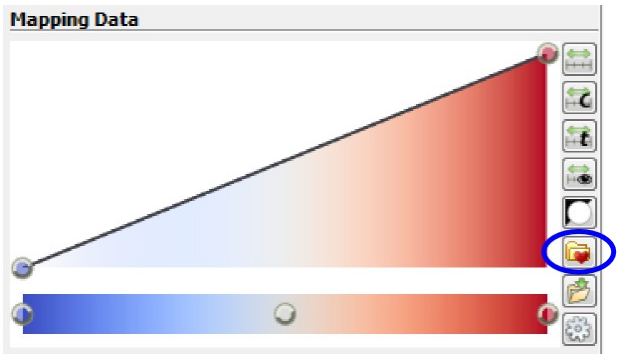
1.4 Color scale and key features

One of the most basic feature of ParaView is the choice of a color scale for an adequate visualization. The default color scale loaded in the beginning is a basic red-white-blue linear scale. The color scale can be fully customized in the *color map editor*:

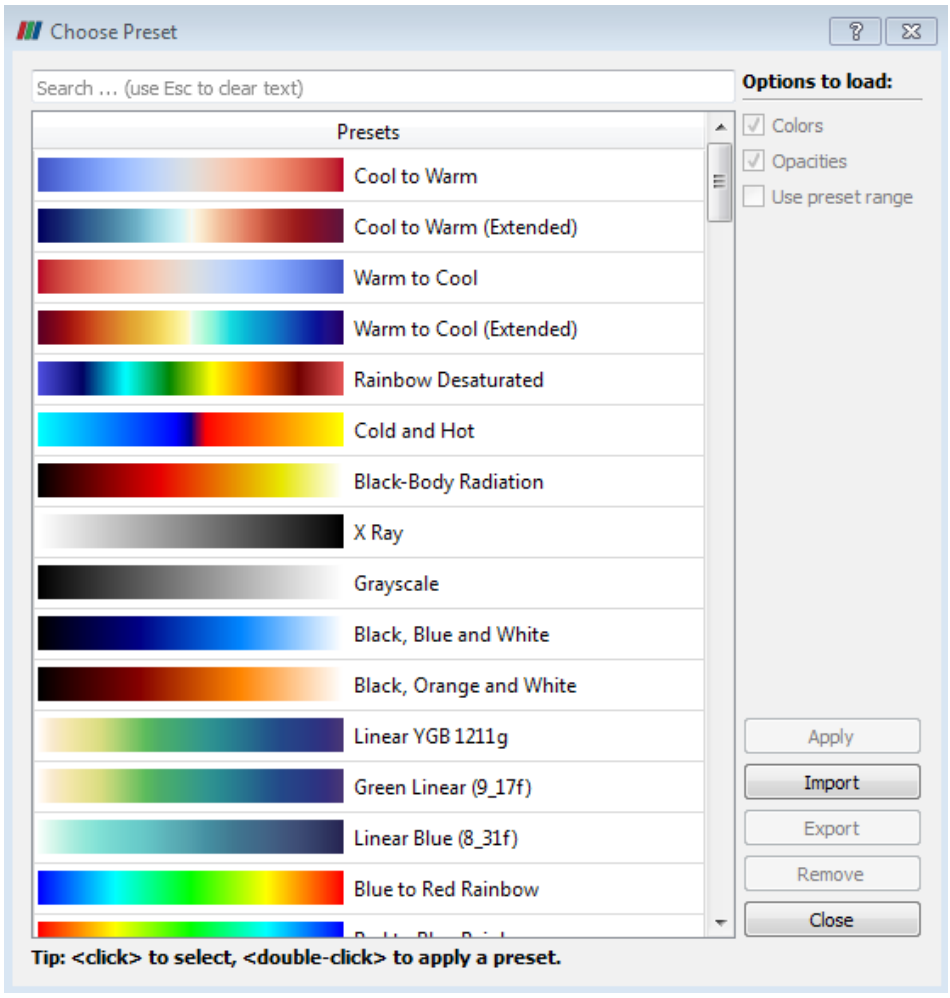
The *color map editor* should appear in the right part of the screen:



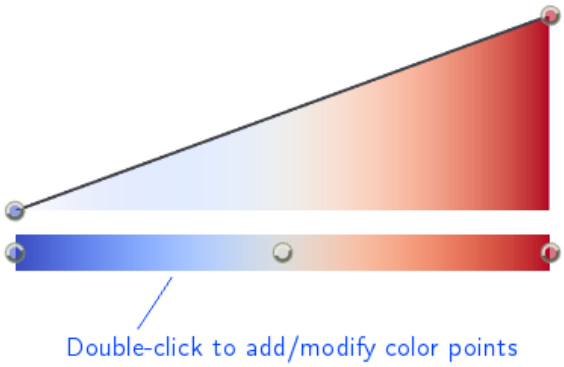
There are a lot of predefined color maps that can be used by picking the button in the color map editor:



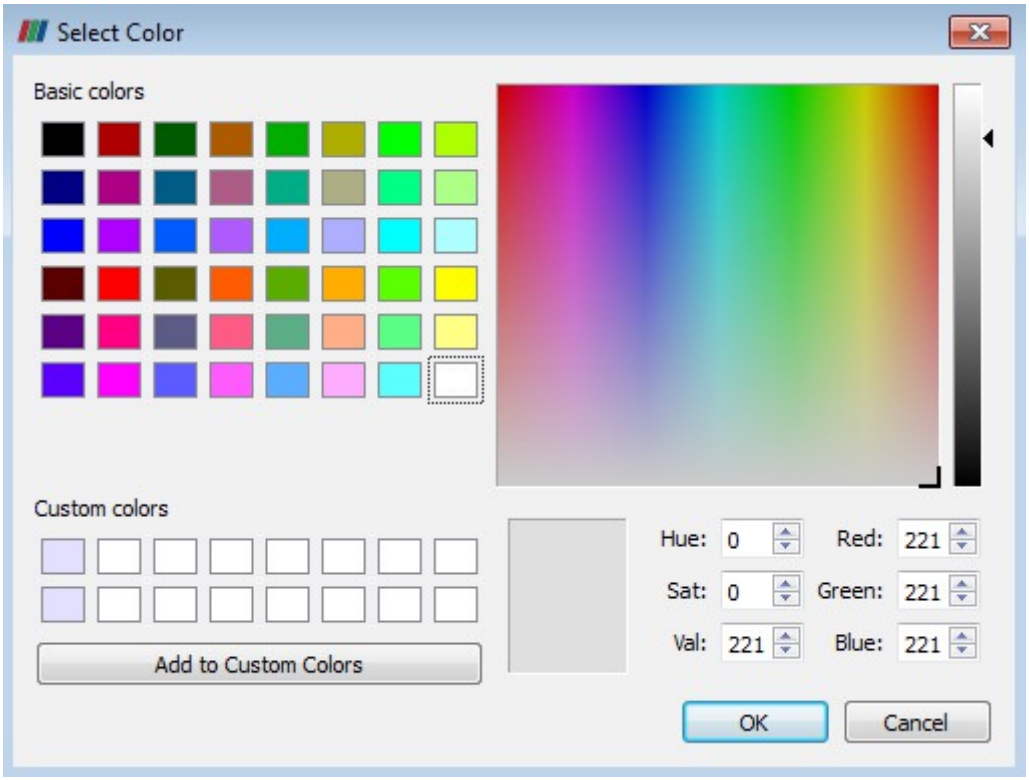
A new window should appear containing a selection of presets:



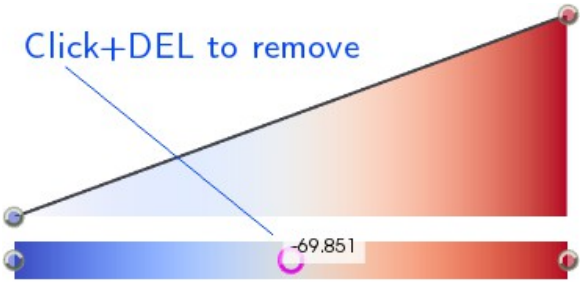
Another possibility consists in editing manally the color map, adding and modifying as many colors as needed. This can be done by double-clicking at any point of the color bar in the *Mapping data* panel inside the *Color map editor*:



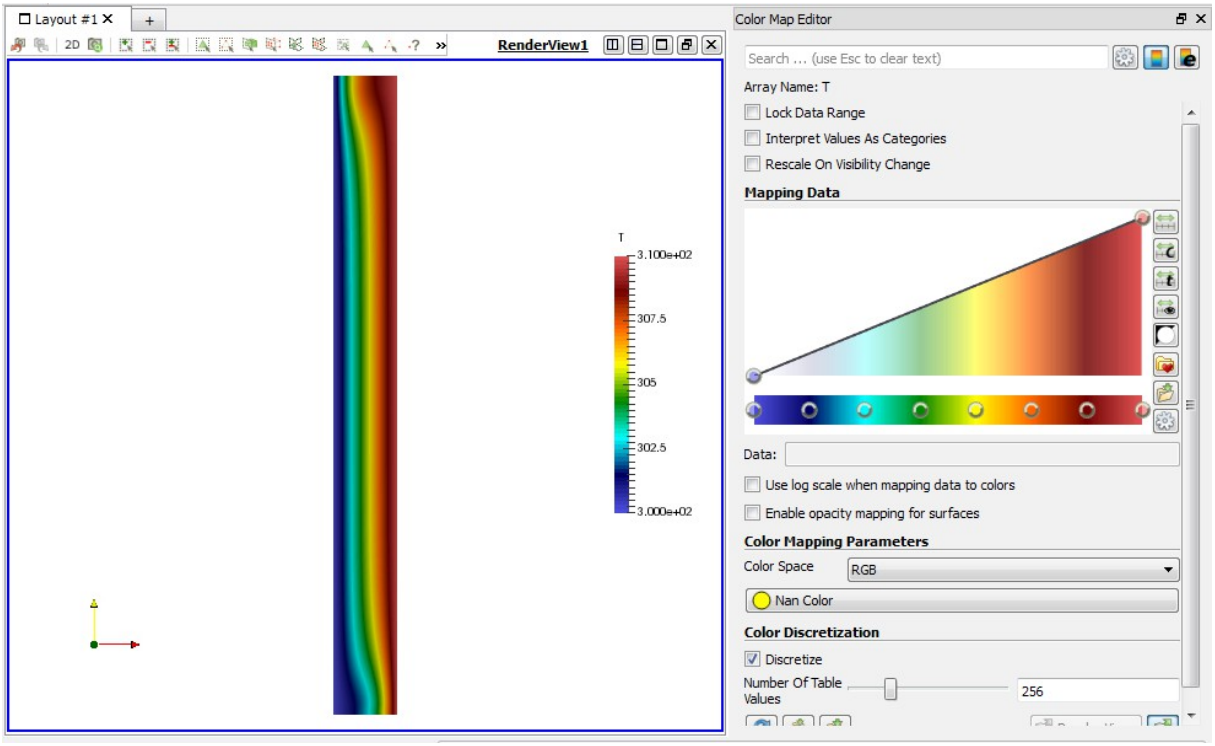
A full color selector will appear, as seen on the next figure:



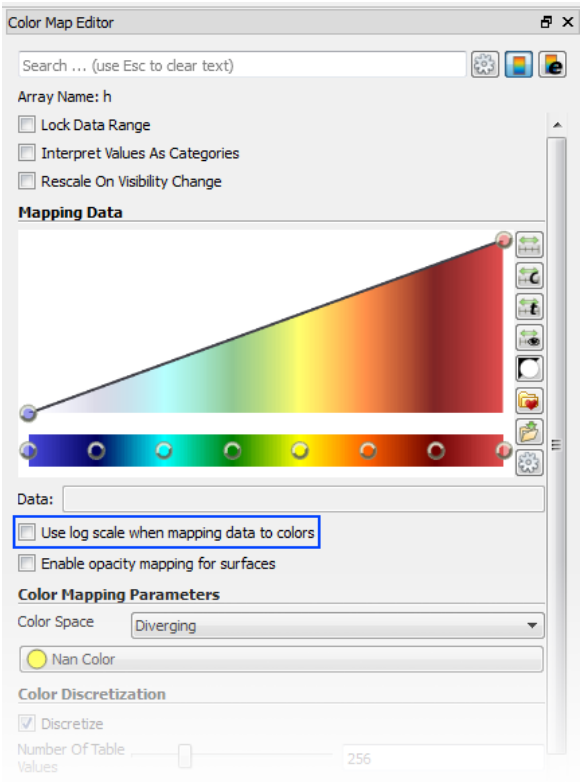
A color can be removed from the color map by clicking in it and pressing the *Delete* key:



For the present example, the Rainbow desaturated color map is chosen:

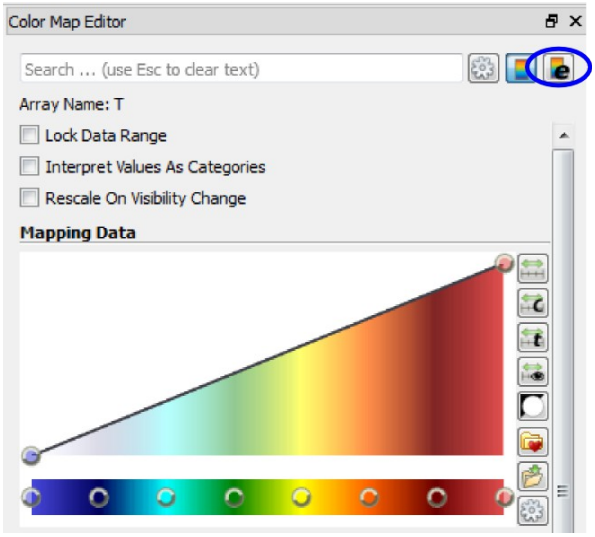


An interesting feature of the color map editor is the possibility of choosing a logarithmic color scale:

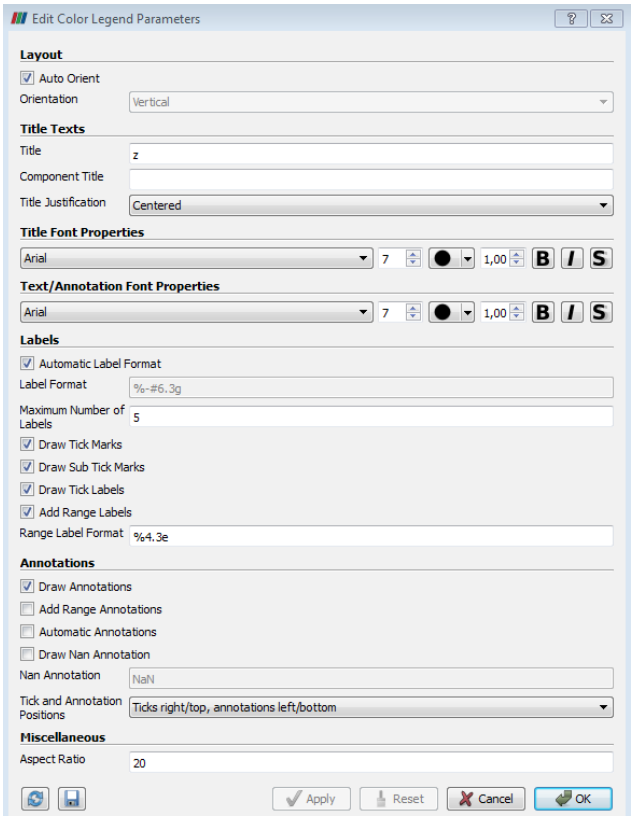


This is specially useful when the dierences in the variables along the domain are small. In this case the logarithmic scale emphasizes these diferences and allows a better visualization. On the other hand, it should be taken into account that negative or null values can not be represented in logarithmic scale.

The legend of the color scale is also configured in the color map editor by:

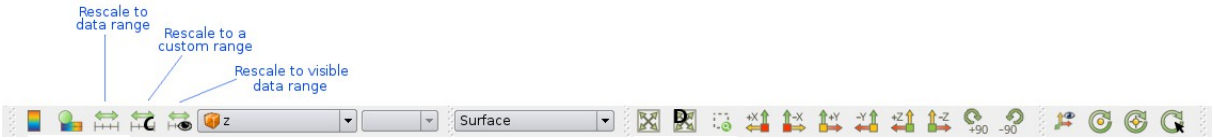


A new window appears in where all the characteristics of the color legend are configured, such as title content and font, labels format, tick marks, etc.



1.5 Adjusting data range

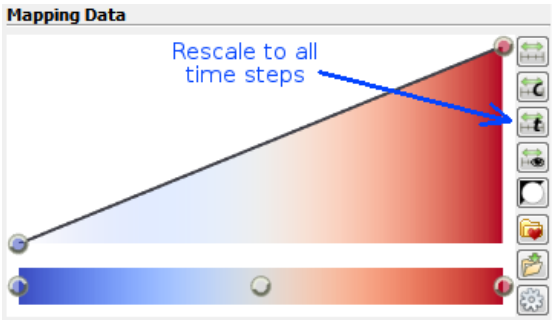
In order to get a proper data visualization and a correct interpretation of the color information, it is important to adjust the color scale to our necessities. ParaView has four different adjustment modes. Three of them are found in the toolbar:



- **Rescale to data range:** Rescales the color using the data range (at the current time step) from the data source selected in the PIpeline browser i.e. the active source.
- **Rescale to custom range:** Rescales the color using a range provided by the user.
- **Rescale to visible data range:** rescales the color using the data range visible in the layout, depending on the zoom level.

and there is one more in the *Color map editor*:

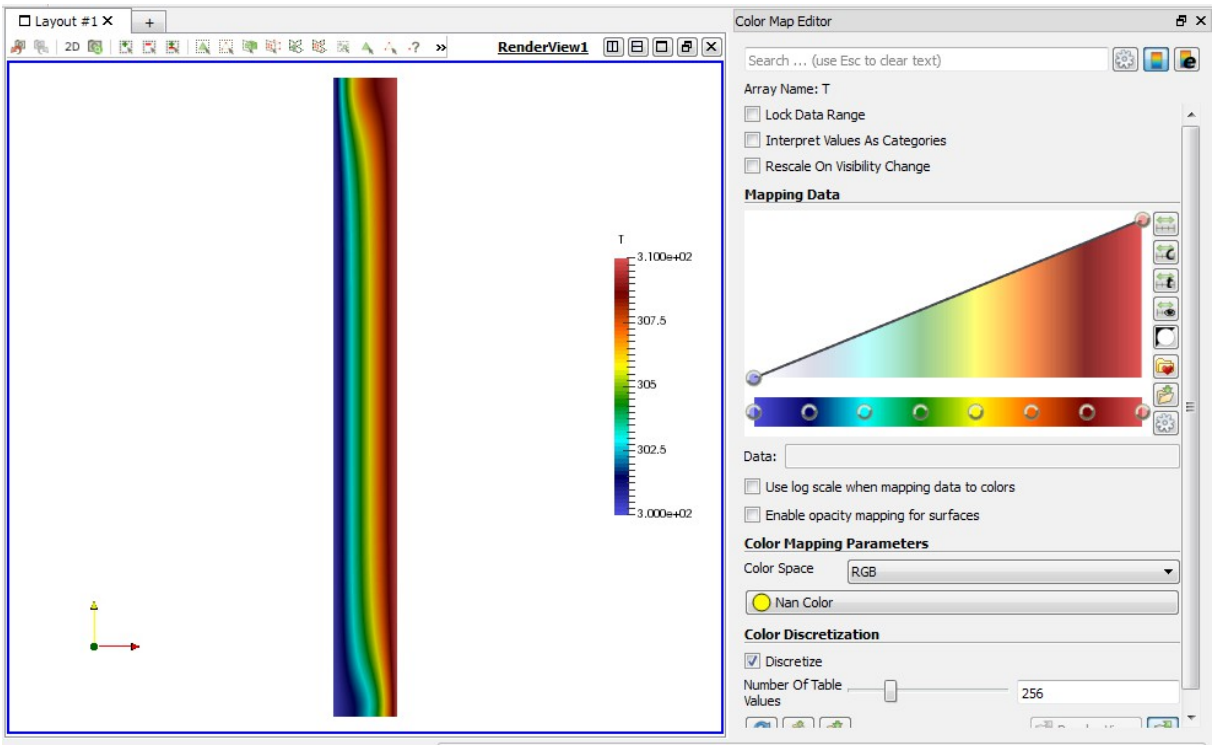
- **Rescale to data range over all time steps:**



It is important to choose carefully the data range. If some of the data values are out of the selected range for the color scale, they are represented by the color corresponding to the upper (or lower) limits of the scale, leading to a misleading color representation of the data.

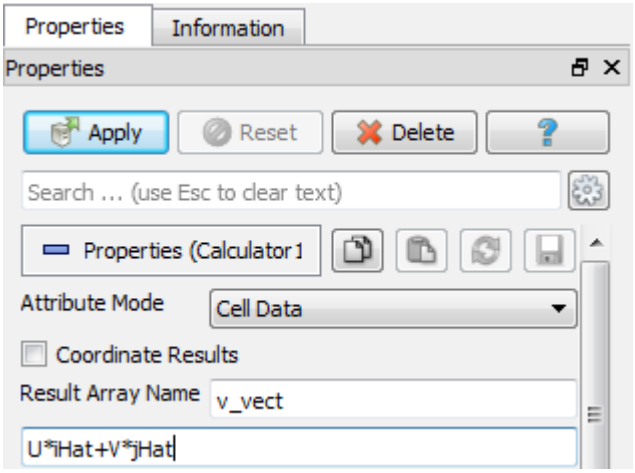
2 Creating velocity vector fields

ParaView has many features and lters for a full visualization of 1D, 2D and 3D data in many dierent ways. In this chapter, the creation of velocity vector fields is explained. This feature provides a very intuitive visualization of the flow direction and magnitude. We will start with the temperature (T) representation in a rainbow color scale, for the sake of clarity. The layout should look as follows:



In order to create a 2D velocity vector map for this example, the velocity vector and modulus variables should be created. This is done by means of the *Calculator* filter (*Common | Calculator*) in the *Filters* menu. Then, follow the next steps:

1. Select *Cell data* as *attribute mode* in the *Properties* panel.
2. Give a name for the new variable (*v_vect*, for instance) in the *Results array name* box.
3. Give the corresponding expression for velocity vector variable: *U*iHat+V*jHat*.
4. Press *Apply*.

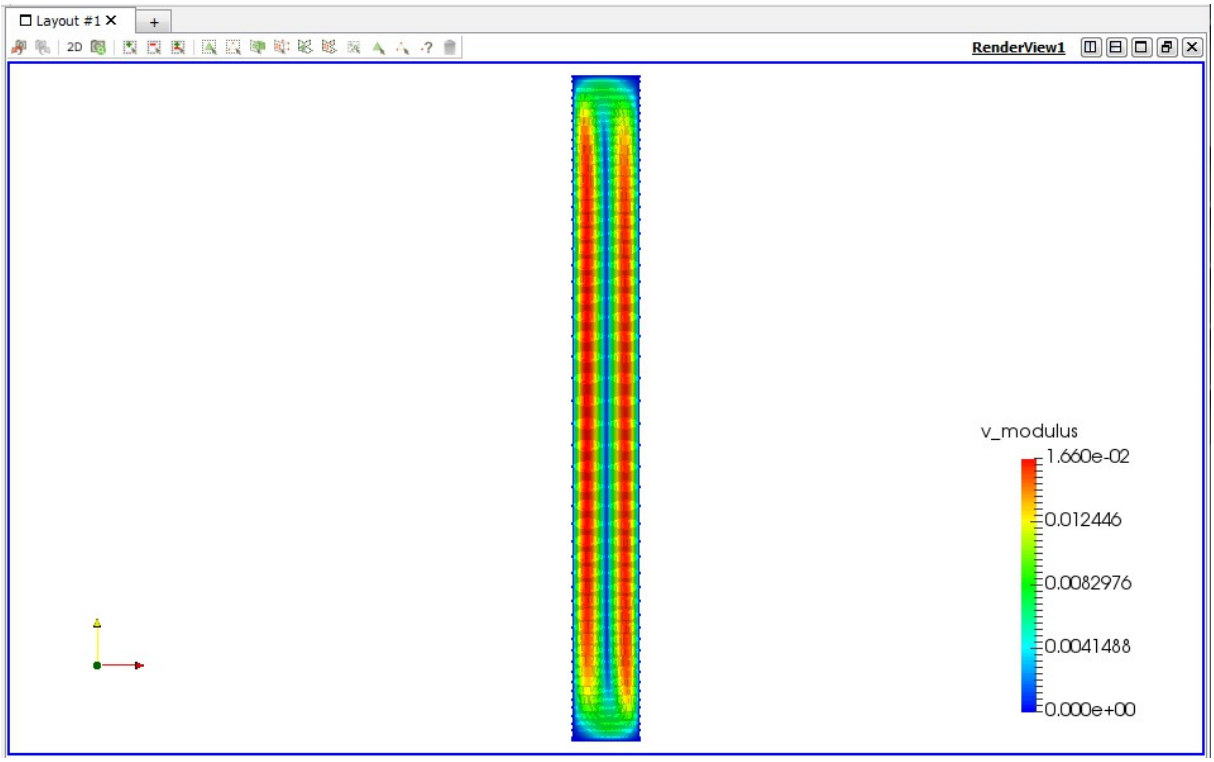


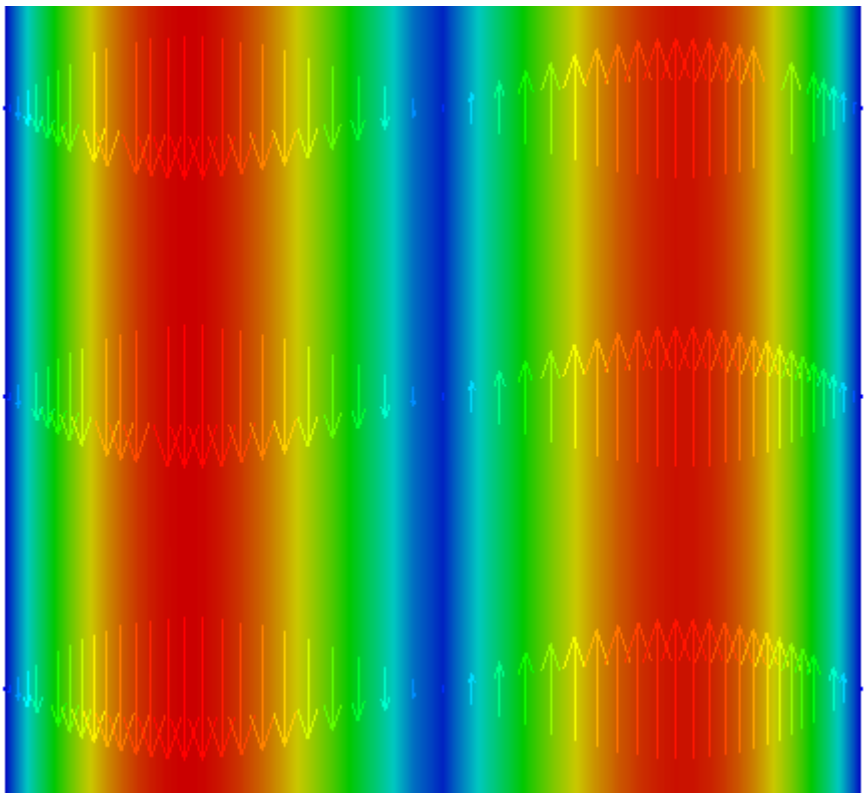
5. Select again the *Calculator* filter in the *Filters* menu.
6. Give a name for the new variable (*v_modulus*, for instance) in the *Result array name* box.
7. Give the corresponding expression for velocity vector variable: *sqrt(U*U+V*V)*.
8. Choose the *Blue to red rainbow* color scale for this variable.
9. Press *Apply*.

Now these variables have been included in the variable selector. Now the velocity vector field can be created by following the next steps:

1. Select the *Calculator2* item in the *Pipeline Browser*.
2. In the *Filters* menu, select *Commom | Glyph*.
3. Now the vector field characteristics are configured in the *Properties* panel:
   - **Glyph source:** Select *2D Glyph* together with *Arrow* as *Glyph type*.
   - **Active attibutes:** Choose velocity modulus (*v_modulus*) and *v_vect* as scalar and vector active attributes, respectively.
   - **Orientation:** Orient box checked.
   - **Scaling:** Set the scale mode on *scalar* with scale factor of 0.1.
   - **Masking:** In order to prevent a saturation of arrows in the visualization, the glyph mode is set as *Every Nth Point* with *Stride=2*.
   - Press *Apply* button (in order to visualize immediately the filter setup, the user can press this button after each of the previous steps).

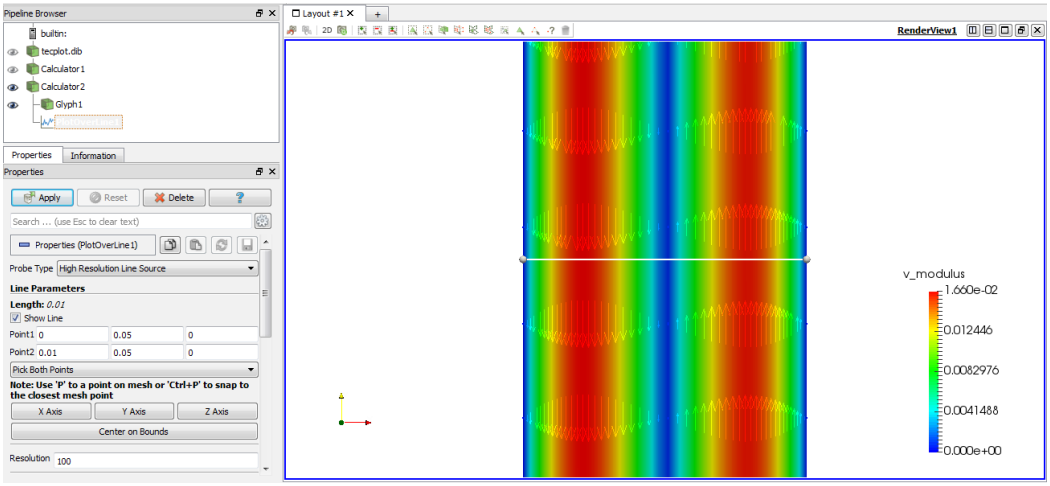
After these steps, the result should look as in the next figures:


Layout #1
RenderView1
v_modulus
1.660e-02
0.012446
0.0082976
0.0041488
0.000e+00

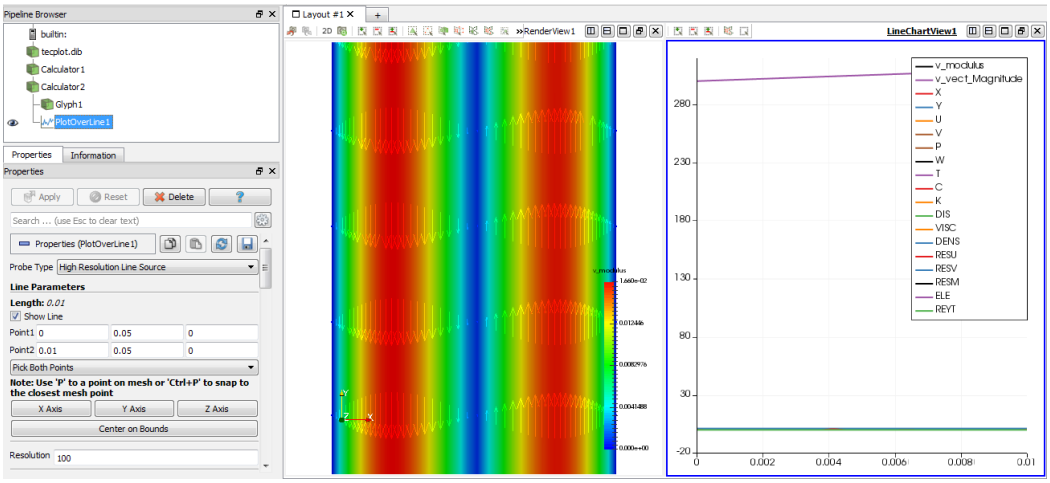


3 Plotting 1D information

Sometimes it is useful to extract 1D information from a 2D domain, as the one used as example in this tutorial. For steady states, the most common way to do it is to plot a variable over a line. In ParaView this is done by means of the *Plot Over Line* filter. Choose the *Calculator2* object in the pipeline browser and select the *Plot Over Line* filter in *Filter* | *Data Analysis* | *Plot Over Line*. The initial and final point should be given by the user by pulling the dots manually in the layout or by specifying the coordinates of both points in the *Properties* panel:



Once the setup is completed, press the *Apply* button:



In order to get an useful visualization, some of the variables should be hidden from the plot. This is done in the *Properties* panel: