ParaView Improvements for the Mantid Project

The VATES (Visualisation and Analysis Toolkit Extensions) is a parallel stream of work we are currently conducting against the MantidProject http://www.mantidproject.org/Main_Page. One of the largest units of work is the addition of 3D visualization capabilities, for which, we have had good success thus far using Paraview. The Vates Simple Interface (VSI) is our home-grown Paraview GUI linking Mantid and Paraview functionality together.

There are some issues we've come across during the development of the plugins and the VSI for the VATES sub-project. Some of these issues deal with trying to present the same operational capability between ParaView and the VSI in order to maintain ease of transition between the two programs. Our goal is to make users comfortable with ParaView mechanics while keeping the toolbox offered by the VSI limited. That way, they can use ParaView to do more complex tasks without too much extra effort. Also, the scientists we serve have come to expect certain functionality and control from visualization packages that we want to have ParaView implement. The sections below are ordered in terms of the priority we would like to see the improvements done.

Non-Orthogonal Axes

Certain scientific datasets have patterns which have non-orthogonal axes associated with them. An example of this is shown in Figure 1. It shows a single crystal with an underlying hexagonal lattice structure in two dimensions. The standard plotting shows the coordinate axes aligned in the normal orthogonal manner. However, this dataset needs to have the red axis aligned at 60 degrees from the green axis. This can be extended into three dimensions for triclinic type crystals. By having the non-orthogonal axes, the associated lattice coordinates for the crystal can be read from the corresponding axis ticks. Some way of providing this information via the filter source plugins is required, ideally vectors could be provided to describe each axis.

Axis Labels and Units

The plugins we have created for the VATES project have access to axis labels (titles) and units for the dataset they are connected too. We would like to be able to set the labels and units via the

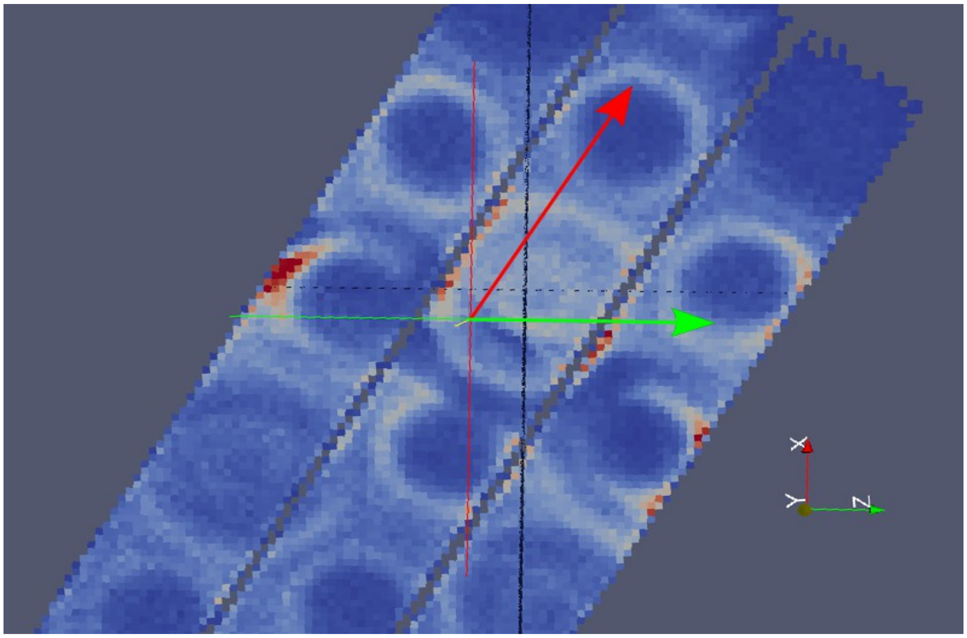


Figure 1: View of data requiring the use of non-orthogonal axes.

ParaView C++ API and have them rendered when the axes are displayed for the given dataset. Currently, the adding of labels to the axes is only available via the ParaView GUI. Some way of providing this information via the filter source plugins is required.

Time Range Updates

We have a rebinner filter developed for the VATES project that allows a user to reorient the data. This includes the ability to change the fourth or time dimension of the data. The filter allows for changing the range extents as well as the number of steps inside that range. We would like for the ParaView GUI animation controls (VCR and Time Information) to reflect these updates. The VSI currently has to check after every rebinner filter invocation and manually update the animation controls.

Labeling the Time Information

The neutron datasets we work with that are four dimensional rarely have time as the actual fourth dimension. We would like for a capability to change the label on the Time Information widget to reflect the correct label and units for the current dataset's “time” axis. This could be done in conjunction with the setting of the other axis labels and then automatically updated once the dataset is rendered.

Axis and Grid Layout

Currently, the axes are restricted to being drawn on the bounding box for the rendered dataset. This restriction causes the biggest annoyance to the scientists. Namely, the corner intersection for the axis triad is not at the (0,0,0) point. They have also requested the ability to adjust that intersection to any arbitrary point. Also, grid lines are drawn on both surfaces of the bounding box for a given axis. We would like to have an option to specify the faces on which the grid lines are drawn. All of these options should be available from the Edit Cube Axes Properties dialog.

A Multislice Viewer for ParaView

The VATES sub-project has created a multislice data view based on the functionality of the Sliceomatic (http://www.mathworks.com/matlabcentral/fileexchange/764) MATLAB add-in. A representative snapshot of the view is shown below.

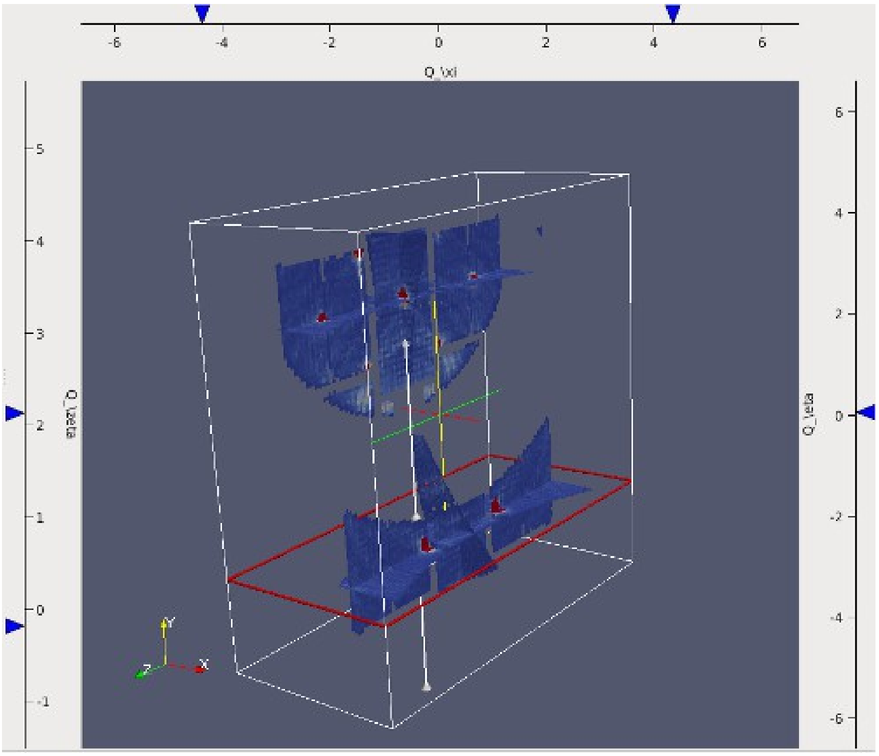


Figure 2: Multislice view from VATES simple interface.

This view is currently available via the MantidPlot program and as a standalone program associated with the Mantid code. This view is part of the VATES Simple Interface (VSI) which uses ParaView based plugins and libraries to provide visualization and manipulation of multidimensional datasets. The goal of the VSI is to provide a targeted set of functionality for handling the multidimensional datasets. If users wish to obtain more capability, our ParaView plugins can be used directly from ParaView itself. We would like to be able to provide the multislice view within ParaView plugin mechanism.

The view consists of a standard ParaView pqRenderView surrounded by three axis controls. The axis controls know the extent (min, max) of the 3 dimensional data. The use of the view is as follows: After loading the data, a slice may be added to the view by clicking on the scale of one of the axis controls. A slice is created orthogonal to the axis control's direction and at the axis coordinate specified when clicking the scale. More slices can be added at any time during the view's life-cycle. An individual slice may be repositioned by grabbing one of the axis indicators (blue triangle) and sliding along the axis control scale. On grab, the slice is immediately selected in the view, a highlight is placed on the selected indicator and the slice is rendered as the indicator is moved about the scale. Each axis indicator also contains a right-click context menu that currently allows the user to hide/show the given slice or to delete that slice. One of the last features to be applied is to enable the capability of hiding slice widget view representation which consists of the data volume outline, plane outline, and directional arrow.

A Threeslice Viewer for ParaView

The VATES sub-project has created a four panel view called the Threeslice view. This consists of a 3-dimensional view having three orthogonal slices added to the dataset. The other three panels are 2-dimensional views of each of the orthogonal slices. A representative snapshot of the view is shown below.

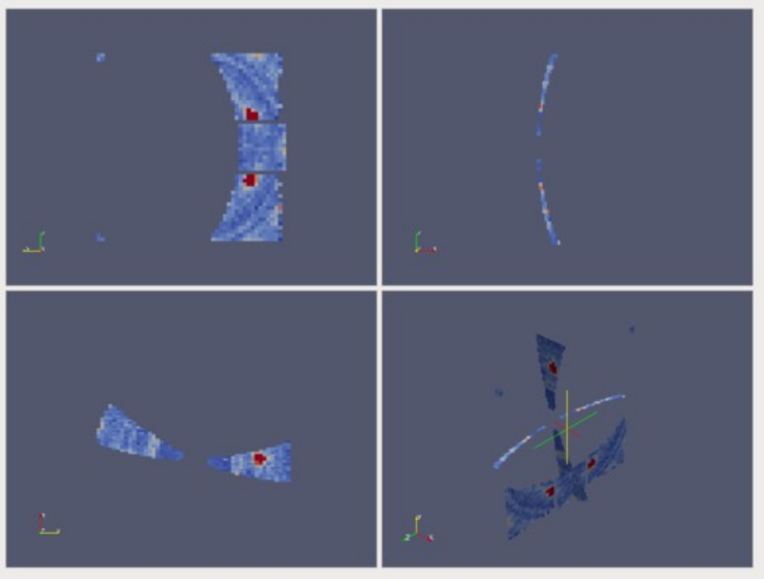


Figure 3: Threeslice view from VATES simple interface.

During the creation of the view, the three orthogonal slices are added by hand to the 3-dimensional dataset and the representative orthogonal slices are plotted in their own pqRenderViews with the 3-dimensional capabilities (trackball, etc.) removed. There is much more functionality we wish to build into this view. It's capabilities should be similar to this example (http://github.enthought.com/mayavi/mayavi/auto/example_volume_slicer.html#example-volume-slicer) using the Mayavi toolkit.

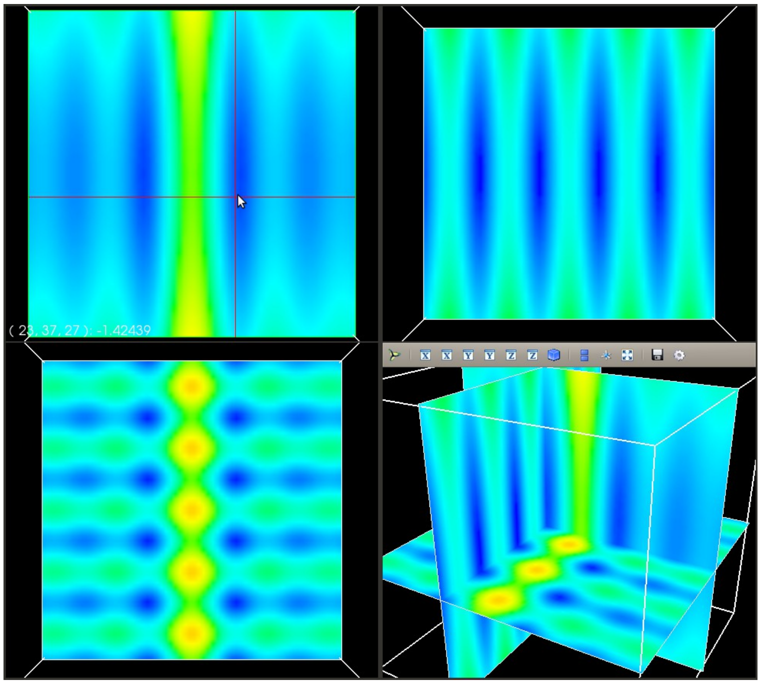
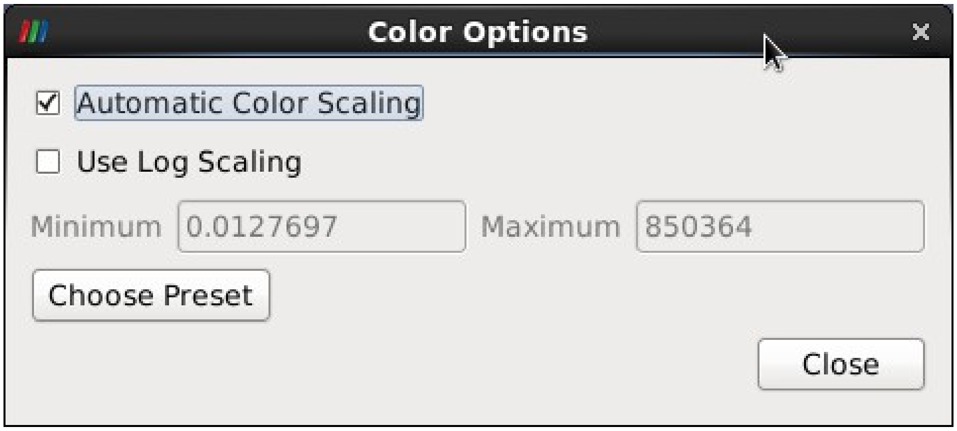


Figure 4: Volume slicer example from the Mayavi toolkit.

Again, there is a 3-dimensional view and three 2-dimensional views of the orthogonal planes. The 3-dimensional view is created using the vtkImagePlaneWidget. Having this widget alone available from the ParaView interface would be extremely useful. The other capabilities for the view center around the 2-dimensional views. When one of those views is clicked, a cross-hair appears on the view and the coordinates of the cross-hair intersection are displayed on the view as text. The cross-hair can be dragged anywhere inside the selected two dimensional view and the other views respond by changing the viewed data for the 2-dimensional views and the corresponding planes on the 3-dimensional view change as well.

Color Mapping Selection

A few of our initial users have complained about the complexity of the color selection dialog box in ParaView. In the VSI, we have attempted to simplify it with the following.



Color Options

☑ Automatic Color Scaling

□ Use Log Scaling

Minimum 0.0127697 Maximum 850364

Figure 5: Simplified color options dialog from VATES simple interface.

We would like to see something similar done for ParaView. Now, we understand that the ParaView user base is used to the current dialog and would not be happy if that changed. Therefore, we would like a configuration option that would allow a user to choose to have the simplified version come up by default while having the original (advanced) still available via a button from the simplified dialog. In this way, the default behavior of ParaView is unchanged.