**A Brief Introduction to ParaView**

**Description:**

ParaView is an open-source platform that can visualize 2D, 3D, and 4D (time varying) datasets. ParaView can process multiple very large data sets in parallel then later collect the results to yield a responsive graphics environment with which a user can interact. The better the processor and graphics hardware the machine or machines hosting the software, the faster and better ParaView will run, however it can run quite well on a laptop such as a MacBook Pro.

Since ParaView is an open source application, anyone can download the program and its source code for modifications. Go to <https://www.paraview.org/download/> and choose the Binary Installers option for your operating system.

For further help, I have attached two PDF documents that are from Kitware. One is a tutorial of the ParaView software and shows the user how to create sources, apply filters, and more. The other is a guide on how to do scripting, macros, and more intense use of the application.

**Installation:**

Open the downloaded binary installer and follow the prompts then drag the application into your applications folder.

**Tour around software:**

Take a look at Section 2.1 of the ParaView Tutorial PDF for details of the application’s GUI environment. The Chapter 2 tutorial as a whole does an excellent job touring the software and its workflow for those unfamiliar with the software and its general capabilities.

**Notes:**

* The best (easiest) way to view data is as some value(s) with an associated point in 3D space. Such that you might have a delimited file that looks like “x, y, z, v1, v2”. This format is incredibly robust and can be processed in the pipeline to become volumes and be filtered for certain thresholds, etc.
* Enable Auto Apply in the preferences. Then you can ignore where I state to click Apply in the walkthrough and it allows you to make changes in the 3D viewer.
* One convenient feature is to save the state of the ParaView environment. This saves all the options you selected on all the filters you applied to visualize some data. Select File->Save State…

**Walkthrough:**

The product of this walkthrough is saved in a state file “faults.pvsm”. When loading be sure to choose to load the files under a specified directory and not the file paths in the state file. Try to follow this walkthrough before looking at the saved state file.

Let’s look at the file “PetrelFault2\_Bradys.dat”. For simplicity, we should change the file extension from “\*.dat” to “\*.txt” as the default “\*.txt” reader is incredibly robust and will handle any delimited file. Notice that the first line of this file is “-316.1086 -1173.0156 133.2091” where our delimiter is a space “ “ and there is no header line. This is quite okay. Open ParaView, select File->Open… and choose “PetrelFault2\_Bradys.txt”. In the Properties window, you will notice some new options appear to tell the application how to read the file. Uncheck the “Have Headers” option as this file does not have headers and change the delimiter from a comma “,” to a space “ “ then click “Apply” at the top of the properties window. A table should appear, with each column corresponding to columns in the file. If the table doesn’t appear that’s okay. Essentially this file reader made a VTK data structure (vtkTable) containing the file's contents. Do this same process for all of the faults files. Once all of the files are loaded, select all to them by using Command-click in the properties window and apply the Group Datasets filter from Filters->Common->Group Datasets which allows us to do batch processing to all of them since they all follow the same format and data type.

Now we need to give meaning to the combined data structures by applying filters. Use the “Table to Points” filter from Filters->Alphabetical->Table To Points. Be sure to change which fields are the x, y, and z fields. Let’s also check the “Keep All Data Arrays” option in the properties window because this data set is just points without an associated value. Click “Apply”. Now we have a dataset that can be visualized in ParaView! Let’s also apply a triangulation to the data set to visualize in a better light. Select Filters->Alphabetical->Delaunay 2D. All options to set up the Filter should be okay to use by default so just click “Apply”. Once the data set is processed, you will notice some new options appear in the Properties window. Change the coloring from “Solid Color” to one of the fields such as “Field 2” to color the surface by elevation. Scroll down in the Properties window and check the “Axis Grid” option to show the axis.

Now you can apply a slice filter or a clip filter. Let’s look at the slice filter which will show how the surfaces which represent the faults intersect some cross-sectional plane. Apply the slice filter, Filters->Common->Slice, to the grouped data sets filter and use the Plane Parameters in the Properties window to manipulate the position of the cross-sectional plane. You can also drag the plane in the Render View, just be sure to have auto apply enabled in your preferences. If you’d like to still see data and use the cross-sectional plane as a means of clipping of the data, apply the Clip filter instead of the Slice filter.