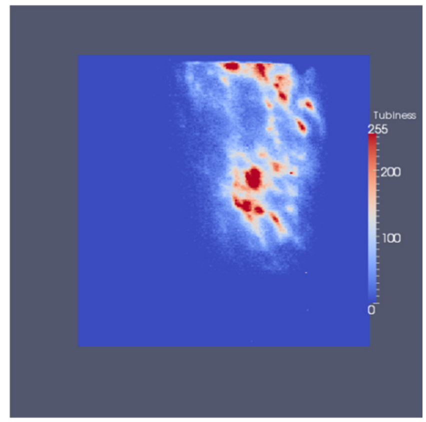
Visualization in the Sciences, Hands-On Workshop

Part 1: Implement various techniques in Paraview

Each of these will start by loading a data set that you are going to display. Copy the **McNeil_CNTs.vtk** data set from the homework ZIP file into C:\tmp. (This includes two RAMAN spectroscopy data sets of carbon nanotubes from Laurie McNeil's group.) Open it in Paraview using **File/Open**.

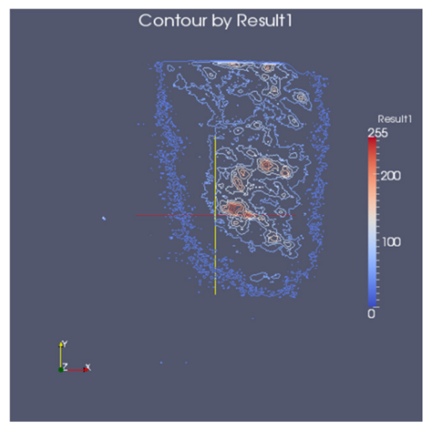
1a: Color map and legend



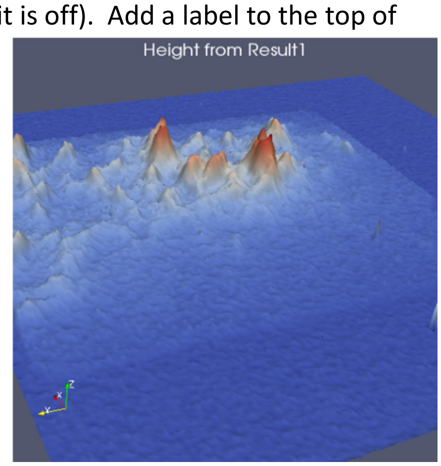
To get it to display the second data set using a color map, you select the **Display** tab in the **Object Inspector** window and then click the pull-down menu on **Color by** and select **Result2**.

To get a legend displayed for the color map, you click on the **Edit Color Map…** button to bring up the **Color Scale Editor** and then click the **Color Legend** tab and click the **Show Color Legend** checkbox. You can name the data set associated with the legend. When you're done, you should have an image like the one shown to the right.

1b: Contour and Label



After restarting Paraview and loading the data set, to get a contour map of the first data set in this file, select **Filters/Alphabetical/Contour** and make sure that **Contour By** is set to **Result1**. To make both data sets (Result1 and Result2) show up on the resulting contour, check the **Compute Scalars** check-box. Add multiple values into the **Value Range** box by pressing **Delete All** and then **New Range**. Click **OK** to use the default values and then press **Apply**. To have the contour colors match the data set you contoured, by, choose the **Display** tab and then use the **Color by** pull-down to select **Result1**.

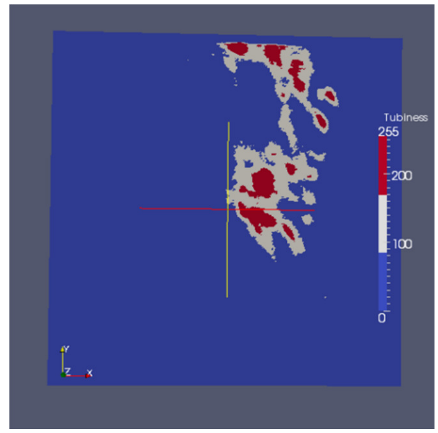


Leave the color legend showing (turn it on as described in Part 3a if it is off). Add a label to the top of the drawing by selecting **Sources/Text**, filling in the text with "Contour by Result1" and then **Apply**. If you want to move the text around, use the **Display** tab and either position by filling in the **Lower Left Corner** values or by using the **Use Window Location** checkbox and the presets below it. The image should look like the one to the right.

1c: Height field

After restarting Paraview and loading the data set, to get a contour map of the first data set in this file, select

Filters/Alphabetical/Warp By Scalar. Set the **Scale Factor** to 0.04 and press **Apply**. To smooth the appearance of the surface, use **Filters/Alphabetical/Generate Surface Normals** and the **Feature Angle** to 90 degrees. Use the left mouse button to set a viewpoint and then add a label to tell which field is mapped to height. If you used Result1 in Warp By Scalar, the image will look something like the one to the right.

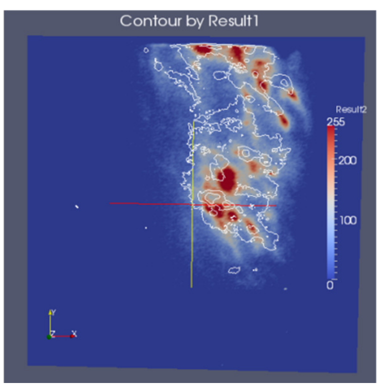


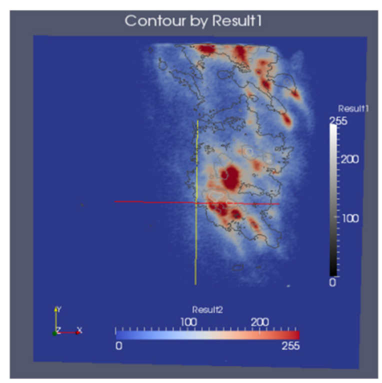
1d: Banded color map and legend

After restarting Paraview and loading the data set, to get a contour map of the first data set in this file, select the **Display** tab and then pick a data set to **Color by**. Click **Edit Color Map** and then set the **Resolution** to 3 (make sure that **Use Discrete Colors** is enabled). The resulting image should look like the one to the right if you color it by Result2.

1e: Color + contour

After restarting Paraview and loading the data set, you're ready to create a 2-data-set display where color comes from one data set and contour comes from another. Go to the **Display** tab and select **Result2** for **Color by**. Then generate a contour for this data set using **Filters/Alphabetical/Contour**. For the contour, **Contour By** the **Result1** data set and generate 5 contours and **Compute Scalars** as before. On the **Display** tab, be sure to **Color by** the **Result1** data set.





To make both data sets visible, click on the grayed-out eyeball icon next to the **Mcneil_CNTs.vtk** entry in the **Pipeline Browser** to make it visible again. To differentiate the contours from the color map, select the **Contour** entry from the **Pipeline Browser**, then the **Display** tab and then set **Color by** to **Solid Color** and pick a color using **Set Solid Color** to white. After adding a text label, the resulting image should look something like the one to the left.

Alternatively, you can **Color by** the **Result1** value and use **Edit Color Map…** to **Choose Preset** of the grayscale or another map and use it. Then you can go to the **Color Legend** tab and select **Show Color Legend for Result1**. Then you can click on the legend itself in the image and drag it to the bottom or left side of the image. This will produce two legends, one for each color map, like in the image on the right.

Part 2: Import custom ASCII file format with implicit coordinates

Sometimes you come across a file format for which there is not an available converter. In these cases, you have to do the conversion yourself. If the file is an ASCII-format file with a header and a body, you can sometimes replace the header and keep the body and end up with a file that VTK can load natively.

This first example is such a case. We'll be loading ASCII files written by the scientist that have five data points per line separated by spaces with no header.

We're going to edit this file (make a copy first) to turn it into one of the *VTK* formats that Paraview can read natively. These file formats are described in the ***VTK-file-formats.pdf*** document that is included in the Zip file. The one we will be using is an ASCII structured-points file. The values in the file are derived from information the scientist told us in email. Add the following entries into the beginning of the file, before any of the data. Be very careful to include things in the order presented and to include every line with appropriate spacing (for example, there is a space between '#' and 'vtk' in the first line). Again, do not include the blue comment text in the header.

- # vtk DataFile Version 2.0 — File-format indicator string ("magic cookie")
- Converted Jonathan Lees file — This comment line can change, but must be present
- ASCII
- DATASET STRUCTURED_POINTS
- DIMENSIONS 7461 4096 1 — Tells the number of points in X, Y, and Z
- SPACING 1 1 0 — Computed based on number of points and X,Y extent
- ORIGIN 0 0 0 — Centered around (0,0) in X and Y
- POINT_DATA 30560256 — 7461x4096x1 data values total
- SCALARS AR float 1 — One scalar per value. Name based on file name (AR)
- LOOKUP_TABLE default — Use the default color map

2a: Edit and load the file

Edit the file ***AR.spec*** using a text editor according to the instructions above and save the resulting file as ***AR.vtk***. Another approach is to write a *header.txt* file with the information above and then run the following command in the DOS command prompt from the directory where the files are: *type header.txt AR.spec > AR.vtk*. Start VTK and open the file. This will read in the file as a 2D structured grid, which will basically be an image that is displayed in the 2D window.

2b: Save and subset the file

It will probably take a long time for ParaView to parse and load the ASCII file. To make future operations faster, you can save the data as a binary file that will be loaded later. To do this, select **File/Save Data** and then change the **Files of type:** dialog to *VTK ImageData Files (*.vti)*. Give the file the name AR.vti and click **OK**. Make sure **Compressor Type** is set to *None*. Now when you load ***AR.vti***, it will read in much faster.

To avoid crashing your computer by running out of memory in the following steps, you'll need to subset the data being used. Use **Filters/Alphabetical/Extract Subset** to do this. The **VOI** should be set from 0 to 3000 on the first axis, from 0 to 1000 on the second, and from 0 to 0 on the third. This will zoom in on one part of the data. To save this subset of data, ensure that *ExtractSubset1* is selected in the **Pipeline Browser** window and then do **File/Save Data**. Still save this as VTI data, but this time with the name ***AR_small.vti***.

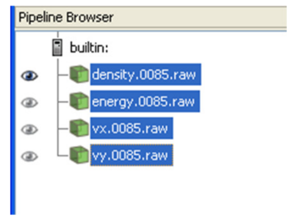
2c: Other files

Do the same operations above for the *MTM.FVMAT* and *MTM.spec* files, to produce *FV_small.vti* and *M_small.vti*. Call the data sets **FV** and **M** in the headers.

Part 3: Combine and compute data sets

3a: Combine

The next step is to combine all three data sets into one, so that you can use different data channels for different visualization features in the same display. To do this, you open all three small data sets and then select them in the **Pipeline Browser** by clicking on the top one and then control-clicking on each of the others (see figure to the right) and then selecting **Filters/Alphabetical/Append Attributes**.



This will create a new object that has all three data sets included in it, so that you can select which channel to use for each display quantity, as we did in Part 1.

To save this as a combined data set, select the **AppendAttributes** entry in the **Pipeline Browser** and then select **File/Save Data**. Save this as a VTK-format file so that you can load it again later and not have to go through the selection and combination steps for Part 4. Include this VTI file in the ZIP file that you turn in for the project, naming it **3a.vtk**. Save it as a binary file so that it will load faster.

3b: Compute

Because the **AR** field has such a wide range of values, we may want to take the logarithm for purposes of display (when this is done, it is critical to let the scientist know and show in the legend that this has been done). We do this using the **Filters/Alphabetical/Calculator**. This brings up a form where we fill in the **Result Array Name** with **logAR** and then fill in the input box above the calculator with the formula to convert: **ln(AR)**. You can either fill this in by typing it or by using the keys and the pull-down **Scalars** box to select the fields that you want.

To make this 2D image-based data set into geometry with attached scalar fields, select **Filters/Alphabetical/Extract Surface**. This will cause a 3D display window to appear alongside the 2D display from before.

When this is done, select the **ExtractSurface1** entry in the **Pipeline Browser** and then **File/Dave Data** to save this as a VTK file format, naming it **3b.vtk**.

Part 4: Displaying multiple data sets

Restart ParaView and open the **3b.vtk** file you saved above. Apply a **Warp Scalar** filter like in part 1 to create a height field, based on the **logAR** field. Then color the resulting height field based on the **FV** data set.

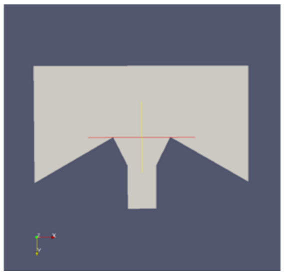
Part 5: Combine and compute data sets

5a: Combine and compute

Re-open Paraview and load the two VTI files named *russ_SP1_140.vti* and *russ_SP2_140.vti*. Use the **Append Attributes** filter like in homework 1 to combine these two data sets into one. Use the **Calculator** filter to produce a field named ***Difference*** that is computed as *Pressure1-Pressure2*. Save the resulting file as ***combined.vtk***.

5b: Select a subset based on data values

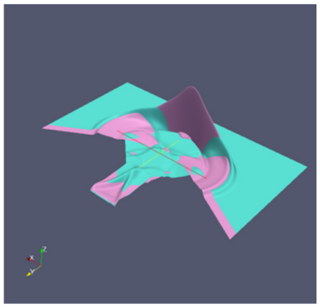
Re-open Paraview and load ***combined.vtk***. Use **Filters/Alphabetical/Threshold** to select a subset of the data based on data value. In the *Properties* tab, select *Scalars* to be *Pressure1* and set the *Lower Threshold* to be at the value -***40000***. If the 3D view window shows only the outline of the data set, you can use the *Display* tab and set *Representation* to ***Surface***. The result should look like the image on the right.



Part 6: Implement various techniques in Paraview

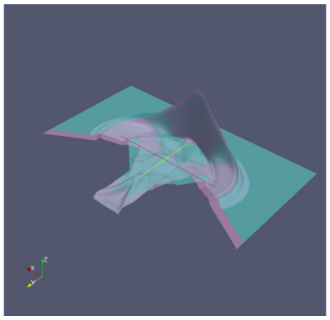
You've now generated two data sets that have two different pressure fields, named **Pressure1** and **Pressure2** that include 2D data from two different simulations with slightly different parameters. We'll take a look at a couple of techniques that might be used to display the differences in pressure between the two data sets here.

6a: Two opaque surfaces



Restart Paraview and re-open both VTI files). Use the **Threshold** filter on each to get only the portions that are above -40000 (**Important:** be sure to click on the **SP2_140.vti** object before adding the second threshold, so that it will add the filter to the appropriate location.) Then select each **Threshold** filter and do a **Warp by Scalar** filter with a *Scale Factor* of ***0.002*** to produce two height fields. Run **Filters/Alphabetical/Extract Surface** and then **Filters/Alphabetical/Generate Surface Normals** with the **Feature Angle** set to 180 to produce a smooth surface for each of them. Select the **Display** tab and set **Color by** to **Solid Color** and click on **Set Solid Color** and select light pink for the surface associated with SP1 and unsaturated light blue for the surface associated with SP2.

When you're done, you should have an image like the one shown to the right, which has the red surface higher when the first slice had higher pressure and the blue surface higher when the second pressure was greater. Save the state file and image.

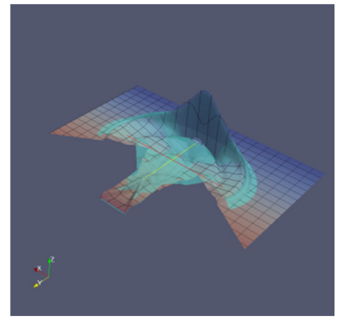


6b: Two translucent surfaces

Select the first **GenerateSurfaceNormals** data set in the **Pipeline Browser** and set the **Opacity** on the **Display** tab to 0.5. Select the second **GenerateSurfaceNormals** and do the same. Save the state file and image.

6c: Two translucent surfaces with contours on one of them

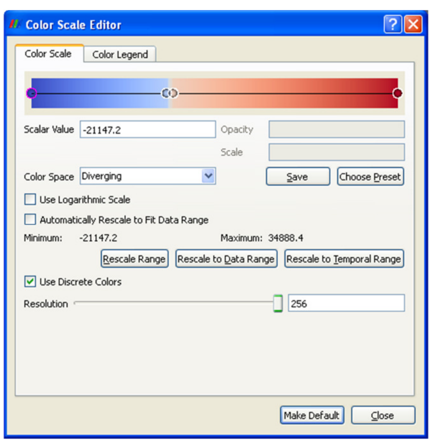
Select the first **GenerateSurfaceNormals** data set in the **Pipeline Browser** and the use the **Calculator** filter to generate a new data set named **X** based on the **coordsX** scalar field; this will basically be a data set that records the X value at every location. Then use the **Contour** filter to generate 20 contours across the data set (use the **Delete All** button and then the **New Range** button) based on **X**.

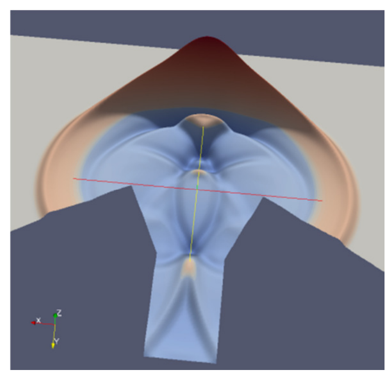


Select the first **GenerateSurfaceNormals** data set again in the **Pipeline Browser** and the use the **Calculator** filter to generate a new data set named **Y** based on the **coordsY** scalar field; this will basically be a data set that records the Z value at every location. Then use the **Contour** filter to generate 20 contours across the data set. Turn off the **Calculator** surface if it is still visible and turn on the first **GenerateSurfaceNormals** using the eyeball buttons in the **Pipeline Browser**. The resulting image should look something like the image to the right (you may have to set the color of the contours to black manually).

6d: Adjusting the color map and range

Restart Paraview and load *russ_SP1_140.vti* data set. **Threshold** and **Warp by Scalar** to produce a 3D display. We want to make a sudden transition to clearly show what is above and below zero. We start by selecting ***Pressure1*** for *Color by* on the *Display* tab and then changing the color map on the *Display* tab using the **Rescaleto Data Range** and then **Edit Color Map** buttons. On the dialog box, we press **Choose Preset** and then select **Cool to Warm** to get a medium-contrast map.





To focus our attention on the changes in the middle part of the image, we move the elements in the color map by clicking to add control points just above and below white and then click on the circles and drag them in the map. We also click on the line near the white dot to produce a new color-map control point and drag it to the left as well. The dialog box looks like the image to the left. To precisely move the control points on the color map, you can un-check *Automatically Rescale to Data Range* and then adjust the *Scale Value* on each point after you have clicked it; you can set the one to the left of white to -500 and the one to the right of white to 500. The image itself now looks like the image to the right.