LLNL-SM-484703

Lawrence Livermore National Laboratory

Visualization with VisIt Class Exercises

3rd Edition

B. J. Whitlock

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Introduction

This document contains the exercises for the *Visualization with VisIt* class. The exercises are broken down into sets that can be completed as the class progresses and you become more familiar with using VisIt. The VisIt class allows time for the first few exercises in each exercise group, making the remaining exercises optional.

Conventions

This document will usually give the exact steps needed to complete the exercises but exercises will get more difficult and sometimes you will only be given the name of a database and an image of a plot, leaving you to figure out how to recreate the image using your acquired knowledge of VisIt's features. Some exercises within an exercise group require the successful completion of the previous exercise.

Text formatting conventions

- Names of controls or windows in VisIt's user interface will appear in **bold** print but words like: window, button, list, and menu will not usually be printed in bold print.
- When you are asked to type text, the text will be *italicized.*
- Filenames will be *italicized.*

Data directory convention

- This class uses many data files in a directory called *VisItClassData*.
- The *VisItClassData* directory may be installed in various locations such as C:\ or on the Desktop. It depends on how the class materials have been installed for your training session.
- The exercises herein will refer to the actual installed location of the *VisItClassData* directory as VISITCLASSDATA and you will need to substitute the appropriate directory when you need to navigate to the data directory.
- The instructor will tell you where data files were installed.
- The instructions may refer to subdirectories under the VISITCLASSDATA directory and will do so by appending a subdirectory name to VISITCLASSDATA (e.g. VISITCLASSDATA\shapefile).

Organization

The organization of this exercise guide roughly follows the topics discussed in the Visualization with VisIt class, though sometimes new concepts may be introduced before they have been discussed in the class for the sake of providing more interesting exercises.

Tips before you begin

- It might be helpful if you close the compute engine every once in a while after making it do a lot of computations to free up memory and resources. Closing the compute engine is especially important if you notice that your Windows desktop computer becomes sluggish.

Exercise group 1: Working with files

The exercises in this section will give you experience for browsing the file system and selecting files using VisIt's **File open window**.

Exercise 1a) Filtering out unwanted files and opening a file

This exercise teaches you how to open the **File open** window and use it to select files that you can plot in VisIt.

1. Open the **File open window** by clicking on **Open** button in the **Main** window's **Source** controls.
2. Type VISITCLASSDATA into the **Path** text field and press the Enter key. Many data files should appear in the **Files** pane of the **File open** window.
3. Type *multi*.silo* into the **Filter** text field and press the Enter key. This will filter out the files that do not begin with "multi" from the list of available files.
4. Click on *multi_rect3d.silo* in the **Files** list to highlight the file.
5. Click the **OK** button to open *multi_rect3d.silo*.

Exercise 1b) Viewing file information

Now that you have opened a file, look at the **File information** window to see what the file contains.

1. Open the **File information** window by clicking on **File information …** in the **Main** window's **File** menu. The **File information** window displays information about the currently open file such as the names of its variables, how many time steps it contains, etc.
2. Click the **File information** window's **Dismiss** button.
3. Notice that the **Plot** and **Operator** buttons in the **Plot list area** becomes active since there is now an open database.
4. Click on the **Plots** button to see that most plot types have variables that they can plot. When a plot type has variables that it can use to create a new plot, it is enabled in the menu.

Exercise 1c) Making a plot

Now that you know how to open a file, and you have opened a file, you can make a plot with that file. We will not cover making a plot in great detail at this point since that will be done later but you will at least learn the basics of creating a plot here.

1. Click the **Plots** button in the **Plot list area**.
2. Move the mouse down the list to the **Pseudocolor** option. This should bring up a small pull-right menu (a **variable** menu) containing the variables from the open file that a Pseudocolor plot can use.

3. Select the **d** option from the Pseudocolor plot's **variable** menu to create a Pseudocolor plot of the variable *d*. Notice that a Pseudocolor plot entry appears in the **plot list** in the center of the **Main** window.
4. Click the **Draw** button above the **Plot list** in the **Main** window to make VisIt draw the plot and put it in the visualization window.
5. Click the **Delete** button to delete the plot from the **plot list**.

Exercise 1d) Opening .visit files

Some database file formats are capable of storing multiple time steps in a single file but most formats only support storing a single time step in a file. VisIt's animation controls are only enabled for databases with multiple time steps. If you have a set of files where each file is a time step in a larger time-varying database, you need to tell VisIt how to recognize all of your files as being related in a larger time-varying database. VisIt supports a .visit file, which is a text file with the .visit extension that contains all of the names for each time state in your time-varying database.

1. Click on the Windows **Start** menu button in the lower left corner of the screen.
2. Click on the **Run …** option in the Windows **Start** menu.
3. Type: *wordpad VISITCLASSDATA\wave.visit* to open *wave.visit* in wordpad so you can examine the contents of the *wave.visit* text file. Be sure to substitute the right path for VISITCLASSDATA.
4. Close wordpad.
5. Open VisIt's **File open** window by clicking on the **Open** button in the **Main** window's **Sources** controls.
6. Type: **.visit* into the **Filter** text field so only files that end in *.visit* will be shown in the **Files** list.
7. Click on *wave.visit* in the list of files.
8. Click the **OK** button to open *wave.visit*.
9. Notice that the **Time** controls in the **Main** window will become active

Exercise 1e) Automatic file grouping

Using .visit files is required in some cases, but often you can try and have VisIt guess how sets of files are related instead of using a .visit file. VisIt provides a feature called *Automatic file grouping*, which causes the database server to scan the list of filenames that it reads from the file system and return them to other VisIt components as a single database entity that we call a virtual database.

A virtual database is a group of related files that have been combined into a database that has no corresponding file on disk. VisIt can sometimes guess file relations incorrectly if the filenames for unrelated files match closely enough to the files that are part of a virtual database. Sometimes, you will need to use the filter in the **File open** window to make sure a virtual database contains only the files that you wanted to be in the virtual database.

Automatic file grouping is enabled by default but it can be turned off so related files are no longer grouped into virtual databases. If automatic file grouping is off and if you have no .visit file to group files, the **Time** controls will not work when you open one of the individual time step files because they only have a single time step.

1. Open the **File open** window.
2. Type: *w*.silo* into the **Filter** text field and press the Enter key to cause the new filter to take effect and populate the **Files** list with only files that begin with *w* and end in *.silo*. The only entry in the **Files** list should be a virtual database called: *wave*.silo database* and it will have a few dozen file names under it but they are part of the virtual database.
3. Select the *Off* option from the F**ile grouping** combo box to turn off automatic file grouping. Note that the *wave*.silo database* disappears and is replaced by the individual filenames.
4. We usually always want automatic file grouping to be enabled so turn it back on by selecting *Smart* or *On* from the **File grouping** combo box.
5. Click on *wave*.silo database* to highlight it.
6. Click the **Ok** button to open *wave*.silo database*.

Exercise 1g) Reopening a database

VisIt allows you to reopen a database. Here are some common reasons for wanting to reopen a database:

- The file changed on disk
- You have a running simulation and more time states were added to the database
- You have switched to a later time state that you know has a new variable and you want that variable to appear in the variable lists.

When you open a database using VisIt, the currently open database is set to the database that you opened. When you click the **Reopen** button, VisIt discards any cached information about the database and rereads that information from the database server so variable lists, etc can be updated using the new database information. Also, any plots that used the database are regenerated by the compute engine after it reopens the database.

1. Open the **File open** window.
2. Type: **.visit* into the Filter text field and press the Enter key to apply the new filter to the list of files. This should cause a new set of files to show up in the **Files** list. All of the new files should have the *.visit* extension.
3. Open *wave.visit*.
4. Click on the **Add** button in the **Plots** controls and add a Pseudocolor plot of *pressure* by moving the mouse down over the **Pseudocolor** option and clicking on **pressure** in the **variable** menu.
5. Click the **Draw** button to make VisIt draw the new plot.
6. Use the **time slider** in the **Main** window to set the active time state to be *0170*. Note that as you move the **Animation slider**, the **Time** text field next to it will

display the cycles for the database. Stop moving the **animation slider** once the **Time** text field contains *0170*.

7. Click the **Reopen** button to make VisIt reopen the *wave.visit* database at the current time state.
8. Click on the **Delete** button above the plot list to delete the plot of *pressure*.
9. Add a Pseudocolor plot of *transient*. Note that this variable was not present in earlier time steps and by reopening the file at this later time step, we obtained new variables to plot.
10. Click the **Draw** button to make VisIt draw the new plot.
11. Click the **Delete** button to make VisIt delete the plot.

Exercise 1h) Replacing a file

VisIt provides support for database replacement, which replaces the databases in the plot list with a new database. If the new database has the same variables as the old database, the plot is regenerated using the new database.

1. Add a Pseudocolor plot of *u*.
2. Click the **Draw** button to make VisIt draw the new plot.
3. Open the **File open** window.
4. Type: **.silo *.visit* into the **Filter** text field and press the Enter key to make VisIt use the new file filter.
5. Click on the *globe.silo* file and open it.
6. Click the **Replace** button to make VisIt replace the current plot from *wave.visit* with a plot from *globe.silo*. This will work because both databases have a variable called *u*. If the database we're using to replace did not have a variable called *u*, VisIt would issue an error message and not regenerate the plot. Note that the **Animation slider** is no longer enabled since the new database has only one time state.
7. Click the **Delete** button to make VisIt delete the plot.

Exercise group 2: Remote visualization

Most of the time, data is generated in parallel on large supercomputers either on site or at other locales. Few users run simulations on their desktop computers so in order to visualize data that was generated on remote computers, the data either has to be moved locally to the desktop, if the data is small enough, or moved to a powerful visualization server where it can be processed. There are two problems with this because the data either has to be moved or performance suffers from having to display the visualization code back to the desktop computer through secure shell. To remove both of these obstacles to visualization, VisIt allows you to do distributed computing, where you can run VisIt locally on your desktop computer but have all of the data processing done in parallel on the same machine that generated the data.

In order to run in distributed mode, VisIt needs to know how to run on the remote computer. You can furnish information about where to look for the VisIt executable or how many processors to use when running in parallel using host profiles. A host profile contains all of the information that VisIt needs to run its compute engine and database server on the remote computer.

These exercises have been removed.

Exercise group 3: Working with plots

Exercise 3a) Hiding a plot

1. Open *multi_curv2d.silo*.
2. Create a Mesh plot of *mesh1* and click the **Draw** button.
3. Create a Pseudocolor plot of *d* and click the **Draw** button. The visualization window should contain plots that together look like a rainbow with mesh lines on it.
4. Create a FilledBoundary plot of mat1 and click the **Draw** button. Note that the FilledBoundary plot covers up the Pseudocolor plot.
5. If you want to see the Pseudocolor plot again, you need to hide the FilledBoundary plot. Click the **Hide/Show** button to hide it.
6. Click the **Hide/Show** button a few times until at last, the FilledBoundary plot is hidden again.
7. Select both the Pseudocolor plot and the FilledBoundary plot in the **plot list**.
8. Click **Hide/Show** a few more times. Note that the *(hidden)* indicator alternates between the two plots. Alternating hidden plots in this manner is more useful in 3D because plots can sometimes occupy the same screen space and cause depth buffer conflicts if you draw them both. This way you only draw one of the plots.
9. Click the **Delete** button until all of the plots have been deleted.

Exercise 3b) Changing a plot's variable

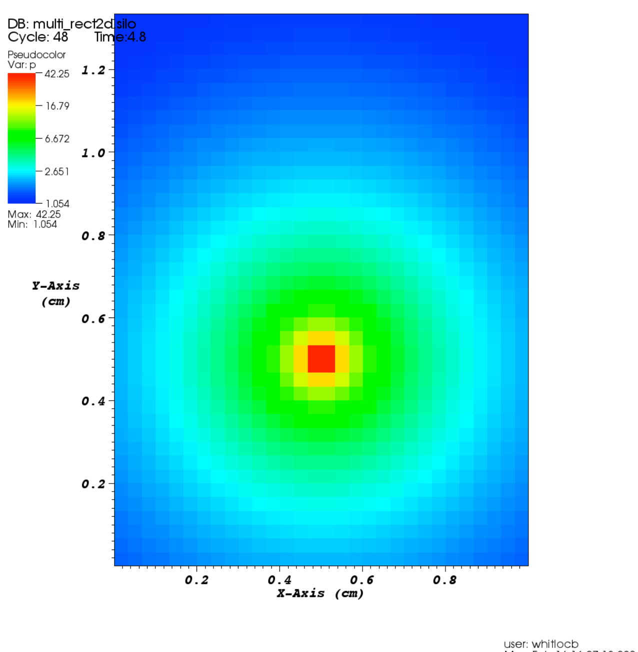
Sometimes it can be a lot of effort to set up a plot with all of its operators and plot attributes. If you want to look at related variables with the same plot and keep switching back and forth, VisIt allows you to change the variable for selected plots.

1. Create a Pseudocolor plot of *d* from *multi_curv2d.silo* and click the **Draw** button.
2. Change the variable to *p* by clicking on the *p* option in the **Variables** menu above the plot list.
3. Change the variable to *u*.
4. Change the variable to *v*.
5. Delete the plot by clicking the **Delete** button.

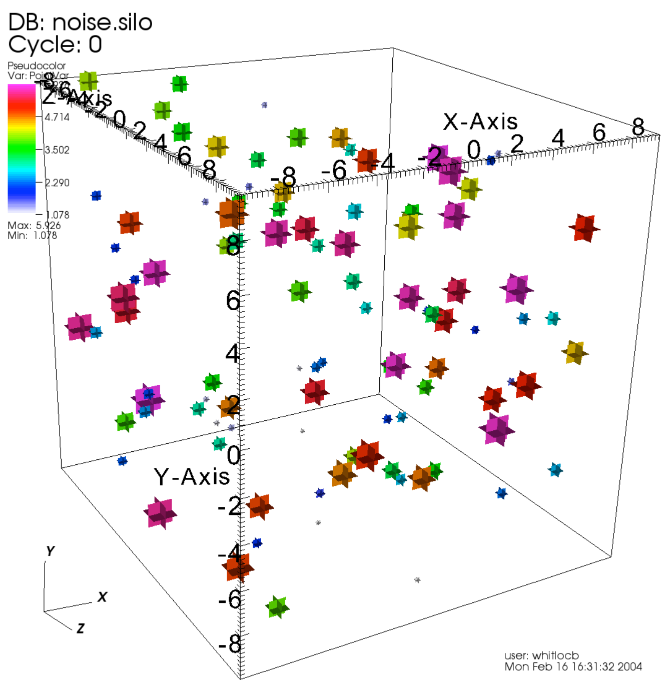
Exercise 3c) Pseudocolor plot

The Pseudocolor plot maps scalar values to colors so you can easily pick out regions of interest in the data. The Pseudocolor plot is one of the most important plots in VisIt.

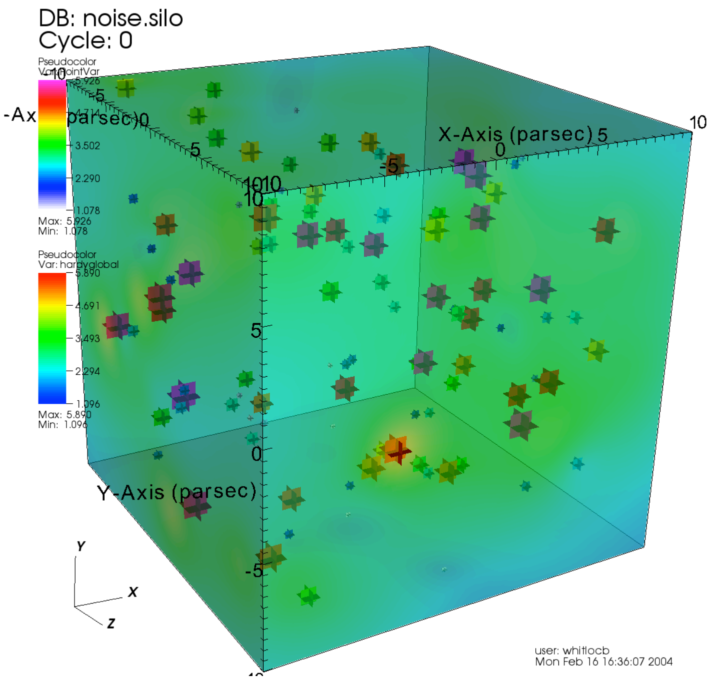
1. Open *multi_rect2d.silo*.
2. Create a Pseudocolor plot of *p*.
3. Click the **Draw** button.
4. Open the **Pseudocolor plot attributes** window. You can double click on the plot list entry to do this.
5. Change the centering of *p*, which is a zone centered variable, to node centered by clicking the **Nodal** radio button and clicking the **Apply** button. Notice that the plot gets smoother. Change back to **Natural** centering when you are done.
6. Click the **Maximum** check box to turn on the **Maximum** text field. Type *10* into the **Maximum** text field and click the **Apply** button. This makes the maximum value used in coloring the plot *10* instead of the actual maximum data value. Setting the limits can improve how colors are spread out when most values in the data cluster near the average data value.
7. Click the **Maximum** check box again to turn it off and click the **Apply** button.
8. Better color mapping can also be achieved by using different data scaling. Click the **Log** radio button to use logarithmic data scaling. Click the **Apply** button.
9. Click on the **Skew** radio button to enable skew scaling. Enter a skew factor of *0.005* into the **Skew factor** text field and click the **Apply** button.
10. Skew scaling can highlight data at the low end or the high end of the data range. Type *10* into the **Skew** factor text filed to highlight data with larger values. Click the **Apply** button.



11. Delete the plot.
12. Open *noise.silo*.
13. Add a Pseudocolor plot of PointVar.
14. Open the **Pseudocolor plot attributes** window.
15. Change the point type to **Axis** so points will be displayed as axis-aligned planes.
16. Click the **Scale point size by variable** check box so point scaling is enabled.
17. Type *0.2* into the **Point size** text field.
18. Click the **Color table** button and select *calewhite*.
19. Click the **Apply** button and then the **Draw** button to make VisIt generate the plot.



20. Add another Pseudocolor plot. This time, use the variable *hardyglobal*.
21. Change the opacity of the second plot to *50%* by clicking on the **Opacity** slider and decreasing the opacity until it reads *50%*.
22. Click the **Apply** button.
23. Click the **Draw** button in the **Main** window to make VisIt draw the second Pseudocolor plot. Once the plot has been generated, the points from the first Pseudocolor plot should be visible inside the second plot because it was made transparent.

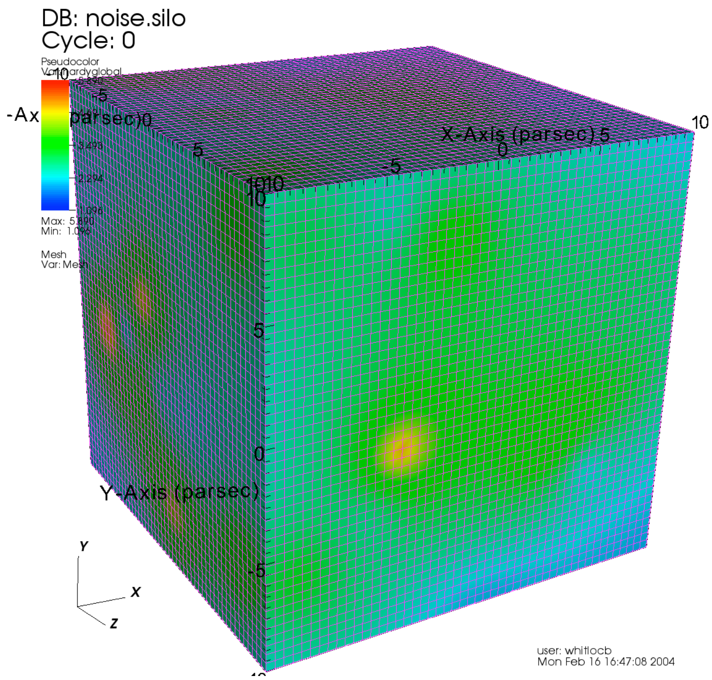


24. Delete both Pseudocolor plots by clicking the **Delete** button in the **Main** window until there are no more plot entries in the **Plot list**.

Exercise 3d) Mesh plot

The Mesh plot shows the mesh lines for a mesh variable.

1. Use *noise.silo*, which was opened in the previous exercise.
2. Create a Pseudcolor plot and click the **Draw** button.
3. Add a Mesh plot of *Mesh* and click the **Draw** button. Notice that the mesh lines get drawn over the Pseudocolor plot.
4. Open the **Mesh plot attributes** window.
5. Most of the time, the Mesh plot uses the vis window's foreground color. If you want to make it use another color, click the **Custom** radio button for mesh color. Once you've done that, click on the **Mesh color** color button and select a new color for the mesh lines.
6. Click the **Apply** button.
7. Delete both plots.

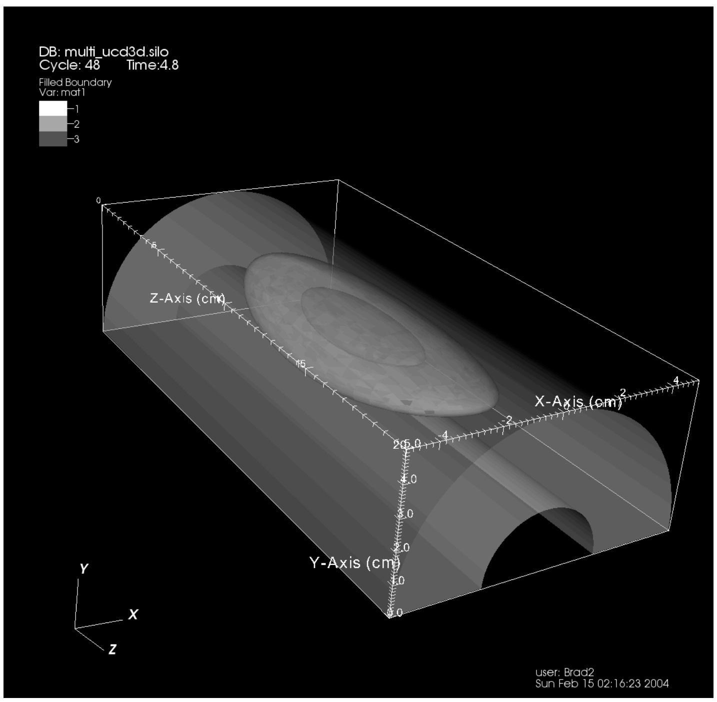


Exercise 3e) FilledBoundary plot

The FilledBoundary plot is for coloring a database's material subsets so it is obvious where they are located in the mesh. This exercise will familiarize you with some of the options for the FilledBoundary plot.

1. Open *multi_curv3d.silo*.
2. Created a FilledBoundary plot of *mat1* and click the **Draw** button.
3. Open the **FilledBoundary plot attributes** window.
4. Click on the **Wireframe** check box to turn on wireframe mode. Change the **Line style** to dashed and click the **Apply** button. The FilledBoundary plot will now only show the edges of the materials in the mesh as a set of wireframe lines.
5. Turn off wireframe mode and click the **Apply** button.
6. Click on the **Multiple** radio button to make sure that the plot's subsets are colored using user defined colors. Click on the color for material 1 in the **Boundaries** list and then click on the color button immediately above it. This will change the color for the highlighted subset once you click the **Apply** button. Set the colors for several material subsets and click the **Apply** button. Note that you may have to rotate the plot in the visualization window in order to see your changes. Your best bet is to change colors beginning at the bottom of the Boundaries list. You can also change the opacities independently for each subset to make some subsets more transparent than others.
7. Open *multi_ucd3d.silo*.
8. Click the Replace button to replace the FilledBoundary plot's database with *multi_ucd3d.silo*.

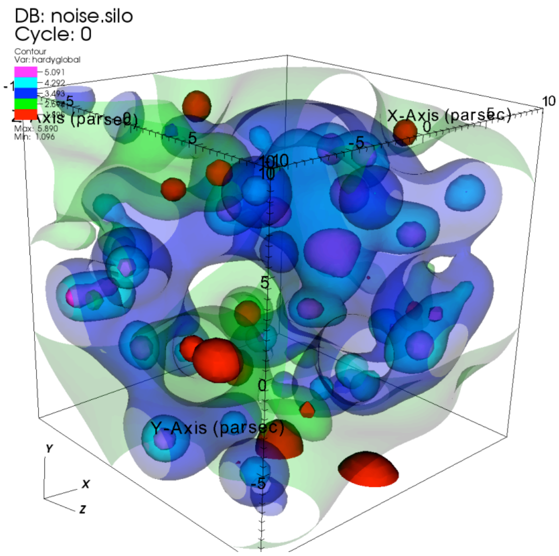
9. Since the variable is a material, make the first material in the **Boundaries** list be partially transparent. This should result in the whole plot being transparent and all the same color.
10. Click the **Draw internal surfaces** check box to show the material boundaries that are internal. Click the **Apply** button.
11. Finally, change opacity of the entire plot by dialing down the **Opacity slider** to about 37%.
12. Make the plot use the *xray* color table instead of using the user-defined colors. Click on **Color table** and then choose the xray color table. This should produce a plot that shows a couple internal layers of transparent materials, which is a useful feature for looking at nested parts. The plot looks reminiscent of a radiograph.

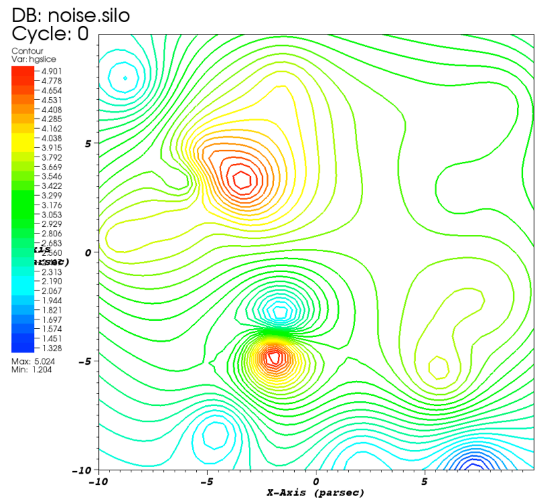


Exercise 3f) Contour plot (optional)

1. Open *noise.silo*.
2. Create a Contour plot of *hardyglobal*.
3. Open the **Contour plot attributes** window.
4. Type *5* for the new **N levels** value and press Enter. This should change the number of colors in the **multiple color list** to *5*.
5. Change the opacity for contour level two to *20%.*
6. Change the opacity for contour level three to *30%.*
7. Change the opacity for contour level four to *40%.*
8. Click the **Apply** button and then the **Draw** button to make VisIt generate the plot.
9. Change the opacities back to 100%.
10. Change the plot variable to *hgslice* using the **Variable**. This will change the plot to 2D.

11. Type *30* for a new **N levels** value and press Enter. This should change the number of colors in the **multiple color list** to *30*.
12. Click the **Color table** radio button to make the plot use a color table for its colors.
13. Click the color table button and select the *hot* color table.
14. Click **Apply** to make VisIt regenerate the plot.

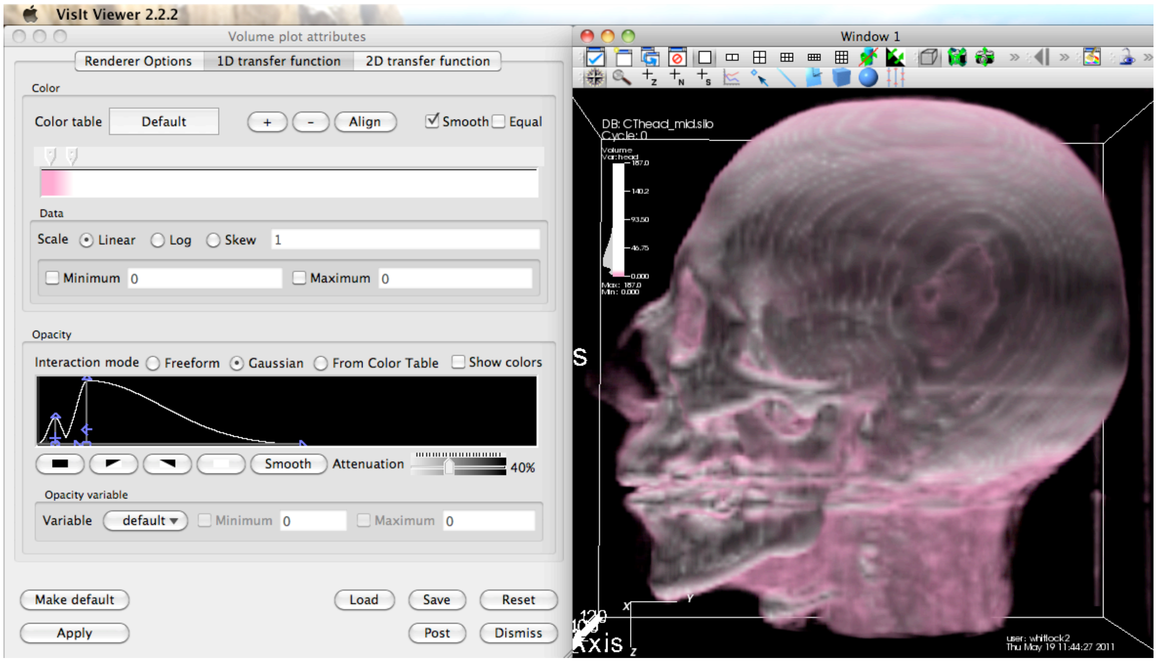




Exercise 3g) Volume plot (optional)

1. Open *CThead_mid.silo*.
2. Create a Volume plot of *head*.
3. Add a Box operator by selecting **Box** from the **Operators/Selection** sub menu.
4. Open the **Box operator attributes** window by selecting **OpAtts->Selection->Box** from the **Main** menu.
5. Type *13.* for **X-minimum**.
6. Type *95.* for **X-maximum**.
7. Type *0.* for **Y-minimum**.
8. Type *120.* for **Y-maximum**.
9. Type *0.* for **Z-minimum**.
10. Type *113.* for **Z-maximum**.
11. Click the **Apply** button.
12. Open the **Volume plot attributes** window.
13. Change the number of sample points to *1000000*.
14. Change the **Rendering Method** to *3D Texturing*.
15. Change the **Number of slices** to 800.
16. Click on the **1D transfer function** tab.
17. Click the **Gaussian** radio button to change to a Gaussian opacity curve.
18. Create some curves by clicking in the **opacity control**. Move the **control points** for the curves to change their shapes.
19. Change the opacity attenuation to 40%.

20. Remove some **color control points** by clicking the **"-"** button. Arrange the color control points and set their colors as shown in the picture below.
21. Click the **Apply** button and then the **Draw** button to make VisIt generate the Volume plot.
22. Experiment with different colors, opacity curves, and views.

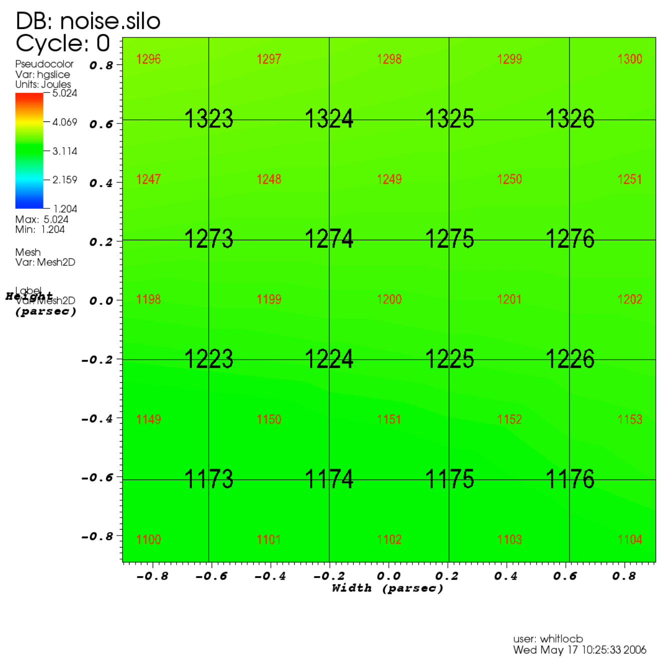


Exercise 3h) Label plot (optional)

VisIt provides the Label plot to make it easier for you to see the actual values used in your simulation's data by allowing you to plot text labels of the values directly on your plots.

1. Open *noise.silo*.
2. Create a Pseudocolor plot of *hgslice* and click the **Draw** button.
3. Create a Mesh plot of *Mesh2D* and click the **Draw** button.
4. Create a Label plot of *hgslice* and click the **Draw** button.
5. Zoom in on the plot and watch the labels appear as you zoom in.
6. Open the **Label plot attributes** window and click off the **Restrict number of labels to** check box and click the **Apply** button. Observe that there are a lot more labels that get displayed because VisIt no longer tries to adaptively fit labels onto the screen.
7. Turn on the **Restrict number of labels to** check box and click the **Apply** button.
8. Enter *300* into the text field next to the **Restrict number of labels to** check box in order to set the approximate number of labels that VisIt will display. Click the **Apply** button.

9. Go to the **Main** window and change the Label plot's variable, using the **Variable** menu, so it plots *Mesh2D* instead of *hgslice*.
10. Zoom in on the plots until the labels are easily readable.
11. Click the **Show nodes** button in the **Label plot attributes** window and click the **Apply** button. Note that the Label plot now displays the mesh's cell and node id's.
12. Click on the **Specify cell label color** check box and select a new color for cell labels. Click the **Apply** button.
13. Increase the **Node label height** to *4%* and click the **Apply** button.
14. Experiment with **Label display format** and watch the labels change between IJ logical indices and normal indices.
15. Delete all plots.

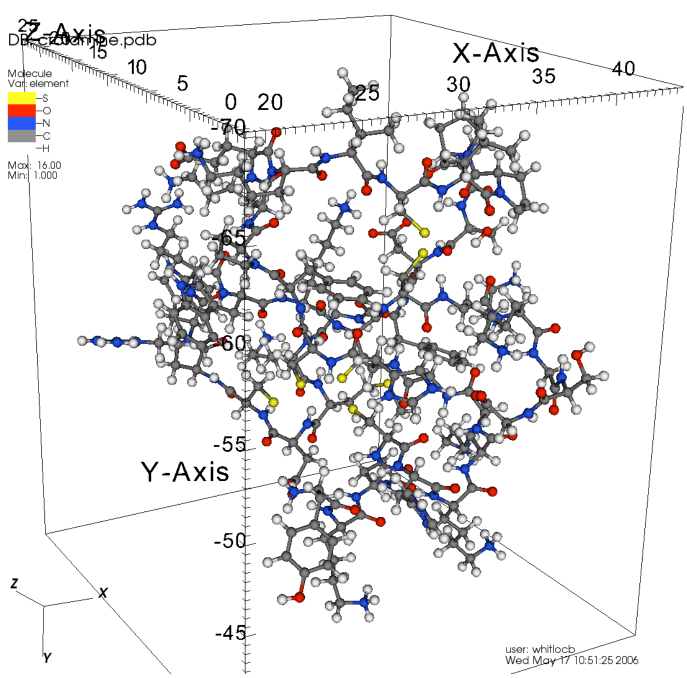


Exercise 3i) Molecule plot (optional)

VisIt's Molecule plot allows you to view Molecular structures out of Protein Databank files.

1. Open *crotamine.pdb*.
2. Add a Molecule plot of *element*.
3. Click the **Draw** button.
4. Change the Molecule plot's active variable to *backbone*.
5. Change the Molecule plot's active variable to *resseq*.
6. Change the Molecule plot's active variable to *restype*.
7. Change the Molecule plot's active variable back to *element*.

8. Open the **Molecule plot attributes** window.
9. Click on the **Atoms** tab and enter *0.4* into the **Fixed atom radius** text field. Click the **Apply** button. Note that the atoms get larger.
10. Zoom into the plot until you see a few dozen atoms. At this point, you can see that they are composed of coarse polygonal spheres.
11. Select *Super* from the **Atom sphere quality** combo box and click the **Apply** button.
12. Reset the view by right-clicking in the visualization window and choosing **Reset view** from the **popup** menu.
13. Try and rotate the plot a little. It may be somewhat sluggish.
14. Select *Sphere imposters* from the **Draw atoms as** combo box and click the **Apply** button.
15. Rotate the plot now. The plot should spin freely. Zoom in on the plot again until you see only a few dozen atoms. Note that this method of drawing atoms uses 1 textured polygon per atom and can thus draw them quickly since much less geometry is involved.
16. Change the **Draw atoms as** method back to *Sphere* and set the **Atom sphere quality** to *Medium* and click the **Apply** button.
17. Change the **Radius based on** value to *Covalent radius* and click the **Apply** button.
18. Turn off the atoms by selecting *None* from the **Draw atoms as** combo box and click the **Apply** button.
19. Click the Bonds tab.
20. Select *Lines* from the **Draw bonds as** combo box and click the **Apply** button.
21. Choose a thicker line value from the **Bond line width** combo box and click the **Apply** button.
22. Click the **Reset** button to restore the defaults for the Molecule plot.
23. Delete the plot.



Exercise group 4: Visualization windows

Exercise 4a) Using multiple windows

It is commonplace to compare databases using multiple VisIt windows. Using multiple windows allows side by side comparison of databases so VisIt provides special features for cloning windows, copying plots, and locking windows together in time and locking views.

1. Open *wave*.silo database*.
2. Add a Pseudocolor plot of pressure.
3. Click the **Draw** button.
4. Click **View** option in the **Lock** submenu in the **Main** window's **Windows** menu. This will make the vis window's view be sent to other windows that have locked views when you change the view.
5. Click **Time** option in the **Lock** submenu in the **Main** window's **Windows** menu. This will make the vis window's time be sent to other windows that have locked time when you change the active time state.
6. Click on the **Clone window** option in the **Main** window's **Windows** menu. The **Clone window** option makes a copy of the active window, including all of its plots and view, etc. Cloning the window will make the new window be the active window. That is fine because we need to click the **Draw** button to get vis window 2's plots to generate.
7. Make vis window 1 be the active window by clicking on the **Make window active** button in the vis window's toolbar. Slowly move the mouse over the buttons in the upper left corner and tool tips will appear for the button under the mouse cursor.
8. Click the **1x2** option in the **Layout** submenu in the **Main** window's **Windows** menu. This will cause VisIt to change the sizes of both windows so they can fit side by side.
9. Rotate the plot in the first vis window. Watch the plot in vis window 2 redraw with the same view as the plot in vis window 1.
10. Change the **Animation slider** in the **Main** window to cycle 0350. Watch the plot in vis window 2 also change to cycle 0350.
11. Make vis window 2 be the active window by selecting 2 from the **Active window** combo box in the **Main** window. This will tell the viewer to make all changes to plots from vis window 2.
12. Change the **Animation slider** to cycle 0500. Watch the plot in vis window 1 also change to cycle 0500.
13. Rotate the view in vis window 2 and zoom in a little bit. Watch the plot in vis window 1 redraw with the same view as the plot in vis window 2.

14. Delete window 2 by dismissing it using the Windows window decorations in the frame around the vis window. VisIt will make vis window 1 be the active window again.

Exercise 4b) View manipulation

VisIt a feature called *Save view*, which allows you to save the current view to a toolbar button that you can press later. You can save up to 16 views and switch between them using a single button click. These views can even be saved to your VisIt configuration settings for use in later VisIt runs. Another view feature that you will use in this exercise is *Undo view*. *Undo view* lets you return to your last view if you make a mistake. You can undo up to 16 view changes.

1. Use the plots from the previous exercise.
2. Locate the **Save view** toolbar button in the **View** toolbar and click it. You can also click **Save view** under the **View** menu in the **popup** menu, which you can activate by right-clicking in the vis window.
3. Once you click the **Save view** button, a new icon with a 1 should appear in the toolbar and the popup menu. If you click on that button, VisIt uses the view that you currently have set.
4. Rotate the plot in the vis window.
5. Click **Save view** again. This should create a new button with a 2 in it.
6. Zoom in on the Pseudocolor plot.
7. Click **Save view** again. This should create a new button with a 3 in it.
8. Click the saved view 1 button.
9. Click the saved view 2 button.
10. Click the saved view 3 button.
11. Zoom in on the plot some more. Let's pretend that you zoomed in accidentally. If you want to undo the zoom, click the **Undo view** button in the toolbar or the popup menu.
12. Delete the Pseudocolor plot.

Exercise 4c) Special 2D view manipulation

Vis windows have several modes. The default mode is navigation mode, which allows you to rotate and translate plots to get a nice view of them. Another important mode is zoom mode. When a vis window is in zoom mode, you can click in the window and sweep out rectangles to zoom in on.

1. Open *RainierElevation.plt*.
2. Create a Pseudocolor plot of *E*.
3. Create a Mesh plot of *zone*.
4. Click the **Draw** button.

5. Switch the vis window into zoom mode by clicking on the magnifying glass button in the vis window's toolbar.
6. Click on the plot and sweep out a zoom rectangle that you would like to see in more detail. If you don't like the rectangle that you zoomed in on, remember that you can click the **Undo view** toolbar button.
7. Zoom in a few times by drawing zoom rectangles.
8. Click the **Reset view** toolbar button to switch to the original view for the plots.
9. Hold down the Shift key and sweep out a zoom rectangle. Note that the rectangle is constrained to be square when you hold down the Shift key.
10. The plots from the current database do not make much use of the vis window's area because they are relatively long and skinny. For plots with this kind of aspect ratio, VisIt provides a Full frame mode, that stretches the plots so they take up more of the vis window. The scaling is not uniform in the X and Y dimensions but it does get the plots to take up more space. Put the vis window in Full frame mode by clicking on **Fullframe** in the popup menu's **View** menu.
11. When you are done, turn off Full frame mode since it is not required for future exercises.
12. Switch the vis window back into navigate mode.
13. Delete the plots.

Exercise group 5: Working with operators

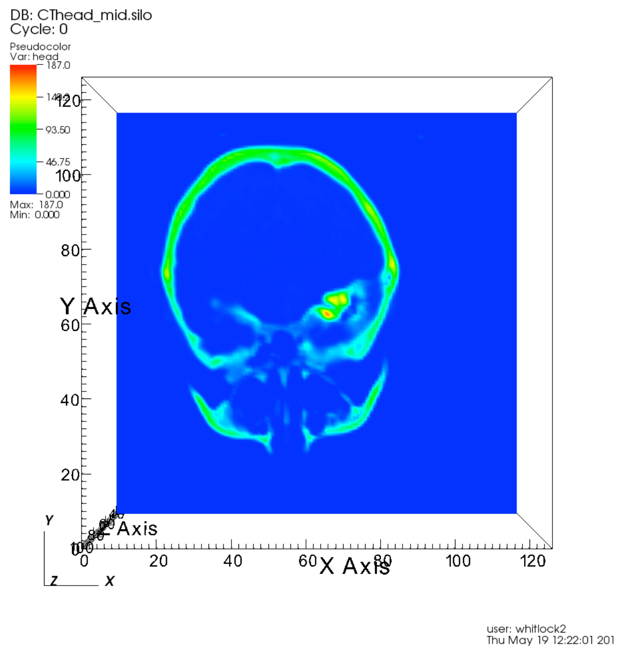
Exercise 5a) Slice operator

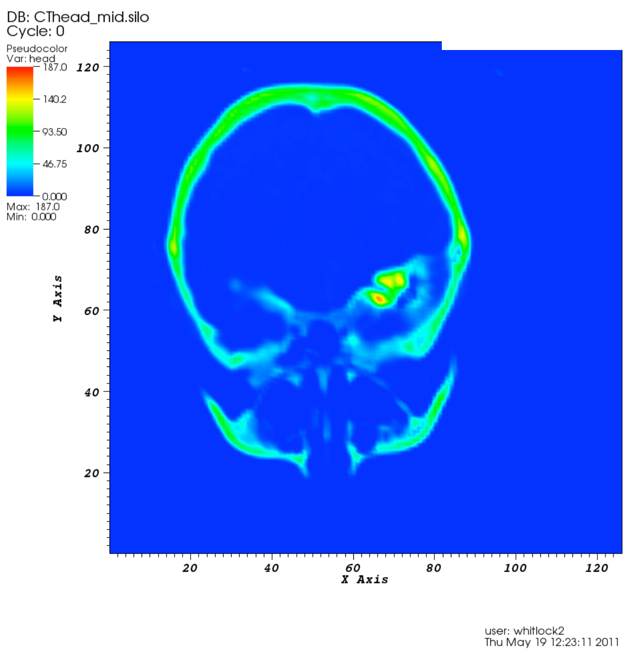
This exercise teaches you how to add an operator and change some of its attributes and make those attributes be the default attributes for the operator.

1. Open *CThead_mid.silo*.
2. Create a Pseudocolor plot of *head* and click the **Draw** button.
3. Add a Slice operator.
4. Open the **Slice operator attributes** window.
5. Turn off the **Project to 2D** check box.
6. Click the **Orthogonal Z axis** radio button to make the slice plane's normal be along the Z axis.
7. Click on the **Percent** radio button in the **Origin** group and set the **Percent** to be *50*. This will create a slice plane that is *50%* of the way through the data along the Z axis.
8. Click the **Apply** button. VisIt may ask whether you want to apply a Slice operator. Click the **Yes** button.
9. Click the **Make default** button to make these slice attributes be the default slice attributes for all future slices. If VisIt asks you if you want to really want to make these slice attributes be the default slice attributes, click the **Ok** button.
10. Delete the Pseudocolor plot
11. Add a Pseudocolor plot of *head* again and click the **Draw** button.
12. Add a Slice operator.

The new Pseudocolor plot should have the same slice plane as your original Pseudocolor plot. If you have not modified your 3D view settings, the picture should look like this:

13. Project the slice to 2D by clicking on the **Project to 2D** check box.
14. Click the **Apply** button. The plot should now be 2D but the data in the slice will look the same as before.
15. Delete the plot.

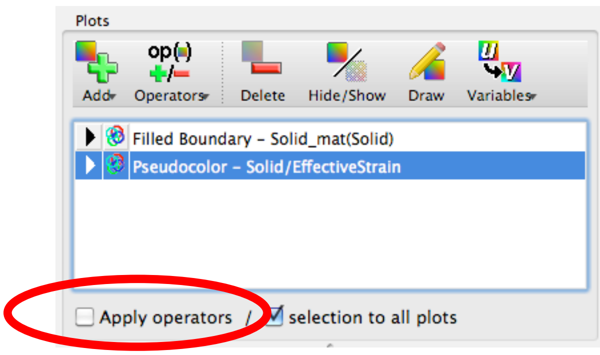




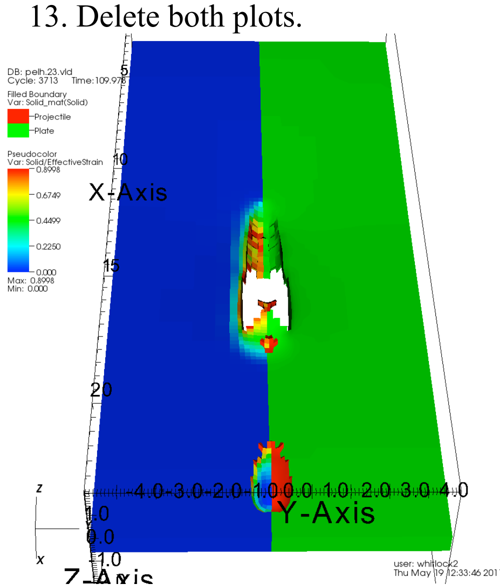
Exercise 5b) Reflect operator

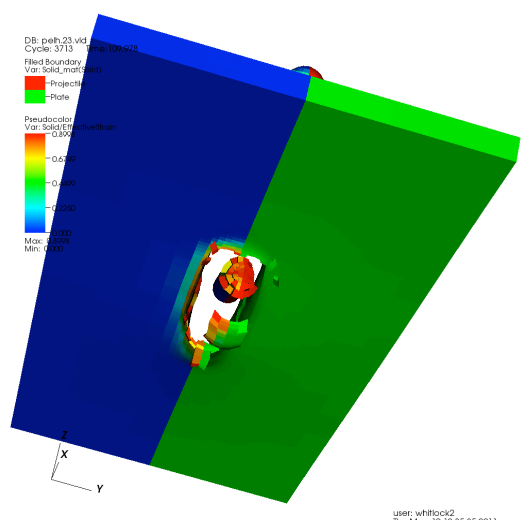
The Reflect operator is useful for displaying the full geometry of a simulation if only half or a quarter of it was simulated due to problem symmetry. The Reflect operator can also be used in this manner to display plots of multiple types when recreating the full geometry through reflection.

1. Open *VISITCLASSDATA\velodyne\pelh.*.vld database*.
2. Create a FilledBoundary plot of *Solid_mat(Solid)*.
3. Create a Pseudocolor plot of *Solid/EffectiveStrain*.
4. Click the **Draw** button. Only the Pseudocolor plot will be visible.
5. Turn off the **Apply operators** check box under the **Plot list** in the **Main** window because we are going to apply the Reflect operator to only the Pseudocolor plot.



6. Open the **Reflect operator attributes** window.
7. Click on the magenta ball to turn off the data in that quadrant. In this case, we only want the data from quadrant 1, which is where the data is located, to be reflected into quadrant 4. This will allow us to have the FilledBoundary plot on top and have the Pseudocolor plot on bottom.
8. Click the **Apply** button and click the **Yes** button when VisIt asks if you want to apply the Reflect operator.
9. Click the **Draw** button.
10. Advance time to the last time step.
11. Try turning on some other quadrants to see what happens.
12. Turn on the **Apply operators** check box in the **Main** window.
13. Delete both plots.

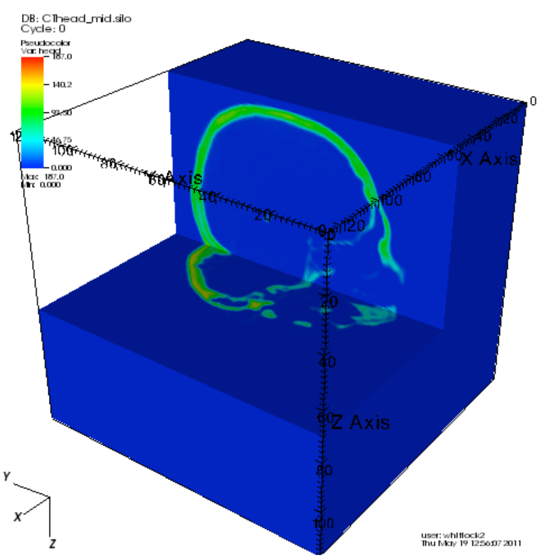


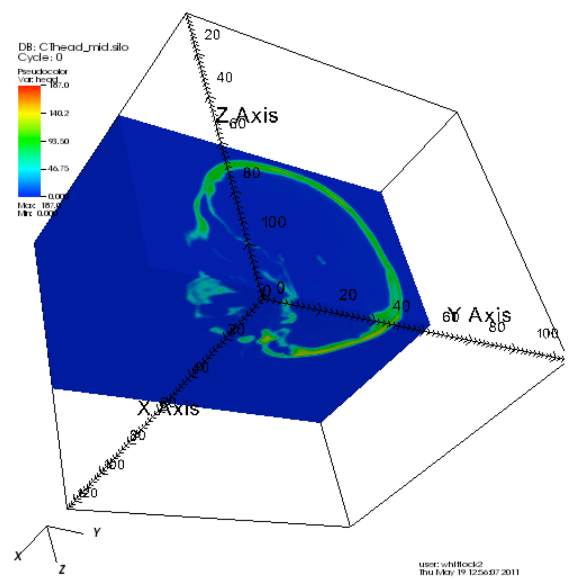


Exercise 5c) Clip operator

The Clip operator clips out regions of a plot using either planes or a sphere as the clipping surface. The resulting plot retains its original dimension. The clip operator is good for being able to see into the interior of a 3D plot. The Clip operator is useful in animations where you remove a wedge of a 3D plot and then move it out to see the inside of the plot.

1. Open CThead_mid.*silo*.
2. Create a Pseudocolor plot of *head*.
3. Add a Clip operator.
4. Open the **Clip operator attributes** window.
5. Turn on clipping plane 1 by clicking on the **On** check box in the **Plane 1** settings.
6. Set the **Origin** for plane 1 to *50 0 0*.
7. Turn on clipping plane 2 by clicking on the **On** check box in the **Plane 2** settings.
8. Set the **Origin** for plane 2 to *0 0 70*.
9. Set the **Normal** for plane 2 to *0 0 -1*.
10. Click the **Apply** button in the **Clip operator attributes** window.
11. Click the **Draw** button.
12. Rotate the view so you can see inside the head.
13. Click the **Inverse** button to get the piece we extracted.
14. Delete both plots.





Exercise 5d) Transform operator (optional)

The Transform operator is commonly used to change the scale or orientation of one database so it can be added to the same vis window as another database for comparison purposes. In this exercise, we will create a Contour plot of a skull database and use the

Transform and Clip operators to position a brain from an unrelated database inside the skull!

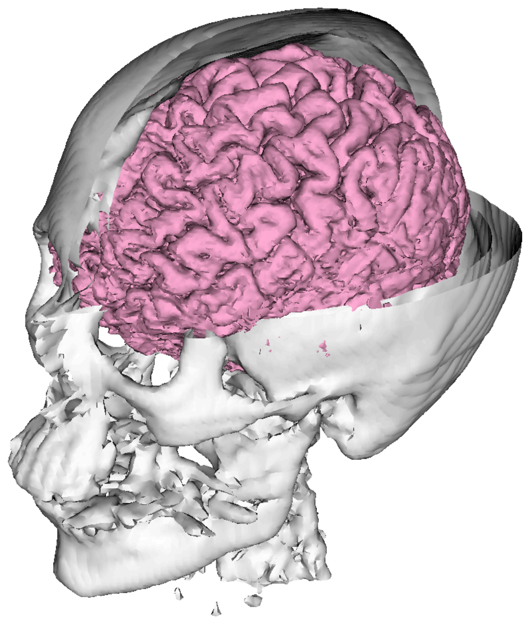
1. Open *CThead_mid.silo*.
2. Add a Contour plot of the *head* variable.
3. Open the **Contour plot attributes** window and change the **Select by** to be **Value(s)**. Type the value: *25* into the text field next to **Select by** and press the Enter key. This will create a single contour for the value *25* in the data range and the multiple color list will update to have a single color. Set the color to white.
4. Click the **Apply** button in the **Contour plot attributes** window.
5. Add a Clip operator to the Contour plot.
6. Open the **Clip operator attributes** window
7. Set the attributes for plane 1. Type: *53 0 0* into the **Origin** text field. Type: *-1 0 0* into the **Normal** text field. Turn plane 1 on.
8. Set the attributes for plane 3. Type: *53 0 47.6* into the **Origin** text field. Type *0 0 –1* into the **Normal** text field. Turn plane 3 on.
9. Click the **Apply** button in the **Clip operator attributes** window.

Here are the instructions for the brain transplant using the Transform operator.

10. Open *s01_anatomy_stripped.img.*
11. Turn off the **Apply operators** check box in the **Main** window. We have to do this because we are going to add multiple Transform operators and if that flag is on then both transforms will get the operator attributes that we set later.
12. Add a Contour plot of *Variable*.
13. Open the **Contour plot attributes** window and change the **Select by** to be **Value(s)**. Type the value: *38* into the text field next to **Select by** and press the Enter key. This will create a single contour for the value *38* in the data range and the multiple color list will update to have a single color. Set the color to be rose.
14. Expand the new Contour plot entry in the plot list.
15. Add a Transform operator using the **Operators** menu.
16. Open the **Transform operator attributes** window and turn on rotation by clicking the **Rotation** check box. Type: *0 1 0* into the **Axis** text field. Type: *90* into the **Amount** text field and make sure that the **Deg** radio button is set so that we are entering a rotation of 90 degrees about the Y-axis.
17. Add another Transform operator using the **Operators** menu.
18. Turn on rotation by clicking the **Rotation** check box. Type: *1 0 0* in the **Axis** text field. Type: *90* into the **Amount** text field and make sure that the **Deg** radio button is set so that we are entering a rotation of 90 degrees about the X-axis.
19. Turn on scaling by clicking the **Scale** check box. Type *0.2646* for the **X** scaling text field. Type *0.2457* for the **Y** scaling text field. Type *0.2457* for the **Z** scaling text field.
20. Turn on translation by clicking the **Translate** check box. Type *51.66* for the **X** translation text field. Type *66.78* for the **Y** translation text field. Type *42.24* for the **Z** translation text field.

21. Click the **Apply** button to apply the new Translate operator attributes to the second Translate operator.
22. Click the **Draw** button.

The resulting plots should look something like this:



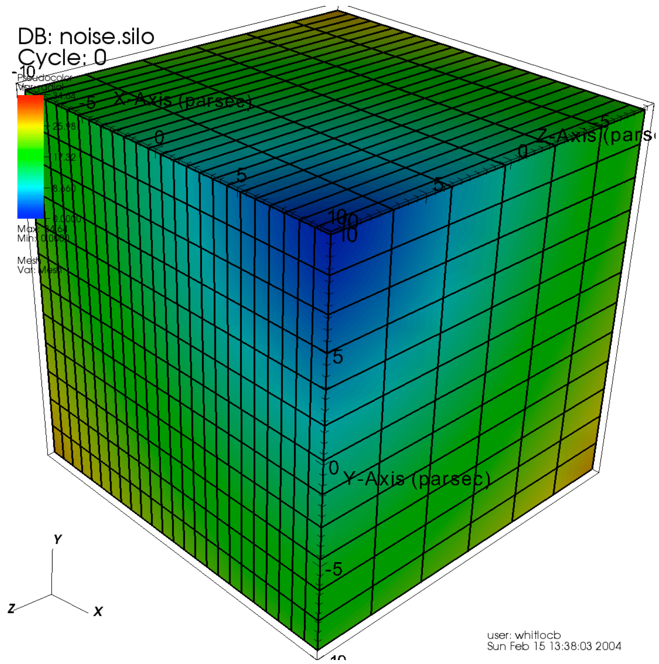
Exercise 5e) Index select operator (optional)

The index select operator allows you to resample your data down to fewer cells. The Index select operator also you select out a rectangular brick of cells.

Resampling

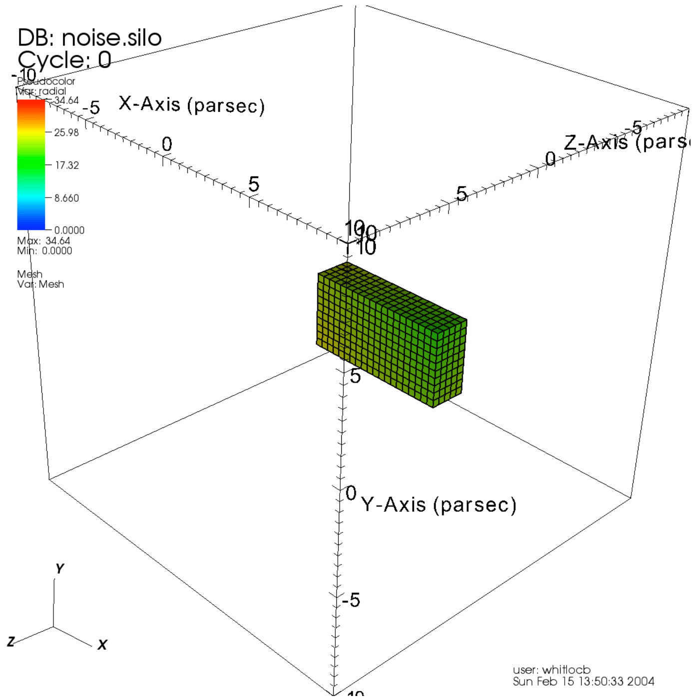
1. Open *noise.silo*.
2. Create a Pseudocolor plot of *radial*.
3. Create a Mesh plot of *Mesh*.
4. Turn on the **Apply operators** check box in the **Main** window if it is not turned on since we want the Index select operator to apply to both plots.
5. Open the **Index select operator attributes** window.
6. Click the **3D** dimension radio button because the plots to which we're applying the operator are 3D.
7. Type *0:max:2* into the top **Min,max,incr** text fields, which are for selecting the cell indices for the X dimension. We're telling the operator to take every other cell.
8. Type *0:max:4* into the middle **Min,max,incr** text fields, which are for selecting the cell indices for the Y dimension. We're telling the operator to take every 4^{th} cell.
9. Type *0:max:8* into the bottom **Min,max,incr** text field, which are for selecting the cell indices for the Z dimension. We're telling the operator to take every 8^{th} cell.

10. Click the **Apply** button in the **Index select operator** window and click the **Yes** button when VisIt asks whether you want to apply the Index select operator. When VisIt is done regenerating the plots, the X dimension should have ½ the cells, the Y dimension should have ¼ the cells, and the Z dimension should have 1/8 the cells that it originally had.
11. Click the **Draw** button.



Extracting a brick of cells

1. Use the current plots and Index select operator.
2. Type *10:30:1* into the top **Min,max,incr** text fields.
3. Type *10:20:1* into the middle **Min,max,incr** text fields.
4. Type *10:15:1* into the bottom **Min,max,incr** text fields.
5. Click the **Apply** button.
6. Delete both plots.



Exercise 5f) Threshold operator (optional)

The Threshold operator removes cells that do not have values that fall within the desired range. This exercise will show you how to remove cells by thresholding using the plotted variable and by thresholding using another database variable.

Threshold using the plotted variable

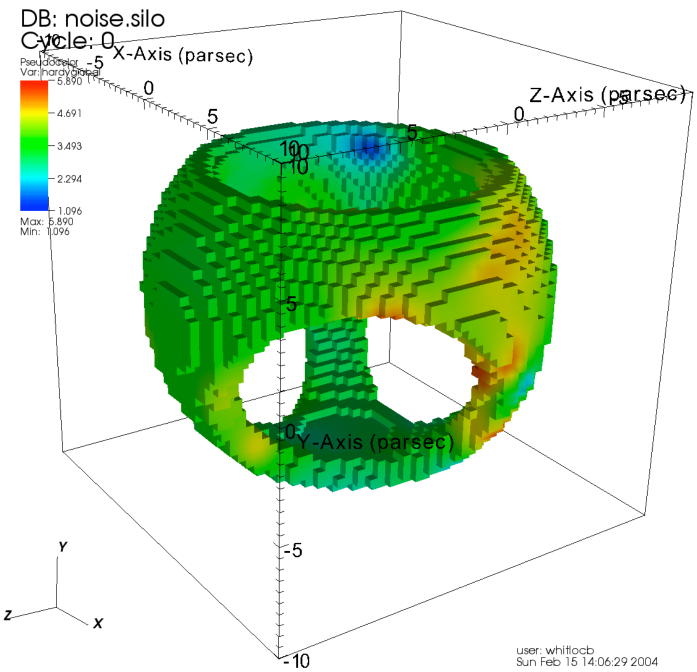
1. Open *noise.silo*.
2. Create a Pseudocolor plot of *hardyglobal*.
3. Click the **Draw** button.
4. Open the **Threshold operator attributes** window.
5. We want to find the cells in the plot that have high values for the plotted variable so type *5.5* into the **Lower bound** text field. This will cause VisIt to throw out all cells that don't have nodes that have values higher than *5.5*.
6. Click the **Apply** button in the **Threshold operator attributes** window.

Threshold using another variable

Sometimes it use useful to see the values for the plotted variable but select cells based on the value of another variable. The Threshold operator usually uses default, which evaluates to the plotted variable but if you want to threshold using another variable, you can do that. Let's show the plotted variable but only for cells whose chrome material volume fraction is at least *0.75*.

1. Click Delete selected variable since we want to threshold by a different variable.
2. Select *chromeVf* from the **Add variable** menu. The *chromeVf* variable is another scalar variable in the *noise.silo* database that contains the chrome material's volume fraction in each cell.
3. Type *0.75* into the **Lower bound** text field.

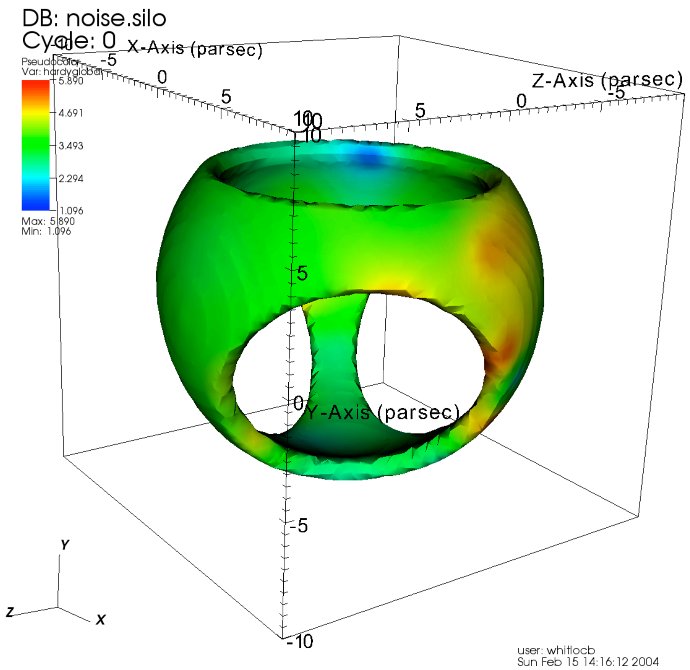
4. Click the **Apply** button.
5. Click the **Draw** button.



Exercise 5g) Isosurface operator (optional)

The Isosurface operator lets you create contours of one variable while coloring the plot by another variable. In this example, we will display a variable while contouring by a material volume fraction.

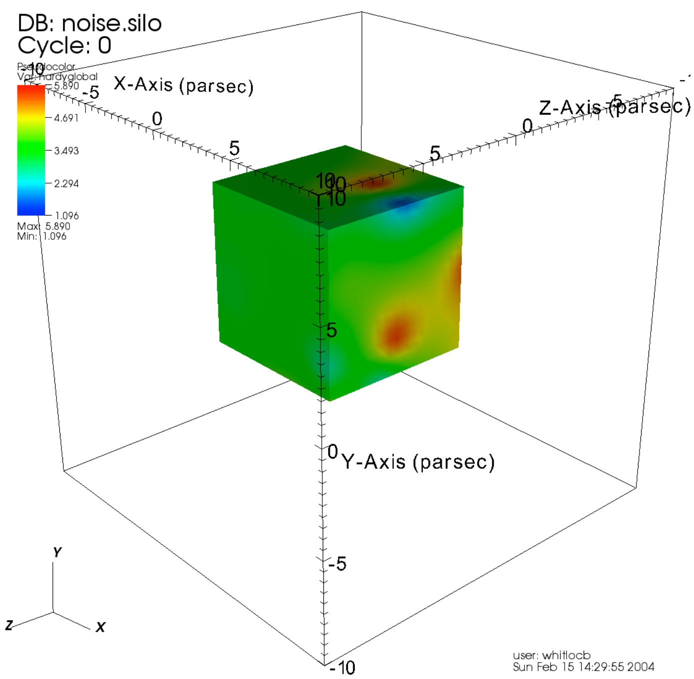
1. Open *noise.silo*.
2. Create a Pseudocolor plot of *hardyglobal*.
3. Click the **Draw** button.
4. Open the **Isosurface operator attributes** window.
5. Change **Select by** to be **Value(s)** and type in a value of *0.7*.
6. Select *chromeVf* from the **variable** menu. We are going to create a single isosurface with a value of *0.7* in the *chromeVf* variable. This has the effect of contouring by a material volume fraction since *chromeVf* is the volume fraction of the chrome material in each cell.
7. Click the **Apply** button.
8. Delete the plot.



Exercise 5h) Box operator (optional)

The Box operator selects cells that lie within an axis-aligned box and removes all other cells from plots to which the Box operator is applied.

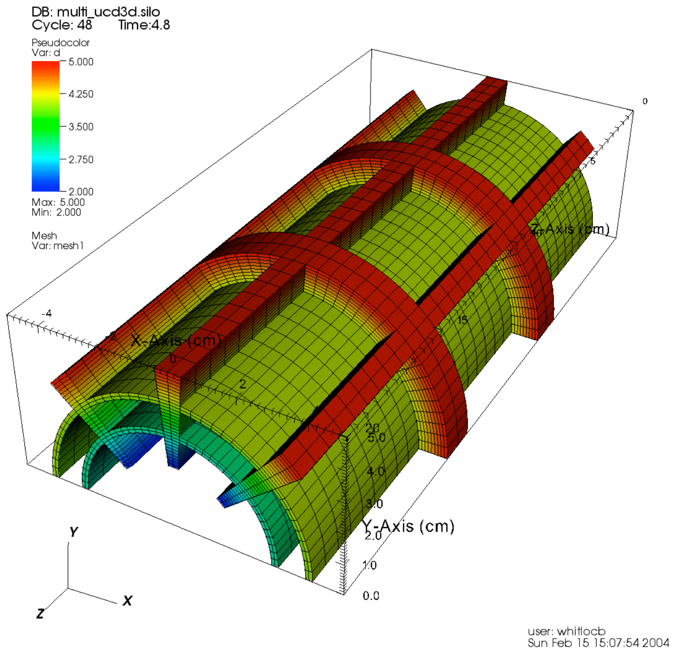
1. Open *noise.silo*.
2. Create a Pseudocolor plot of *hardyglobal*.
3. Open the **Box operator attributes** window.
4. Let's extract only those cells that are in a box whose origin is at *(0,0,0)* and measures *7* on each side.
5. Type *7* into the **X-Maximum** text field.
6. Type *7* into the **Y-Maximum** text field.
7. Type *7* into the **Z-Maximum** text field.
8. Click the **Apply** button.
9. Click the **Draw** button.
10. Delete the plot.



Exercise 5i) Inverse ghost zone operator (optional)

The inverse ghost zone operator is useful for finding out if your database has ghost zones, which are zones at domain boundaries that are normally not displayed and used internally for such purposes as producing smooth Contour plots.

1. Open *multi_ucd3d.silo*.
2. Create a Pseudocolor plot of *d*.
3. Create a Mesh plot of *mesh1*.
4. Click the **Draw** button.
5. Add an Inverse ghost zone operator.
6. Delete the plot when you are done.

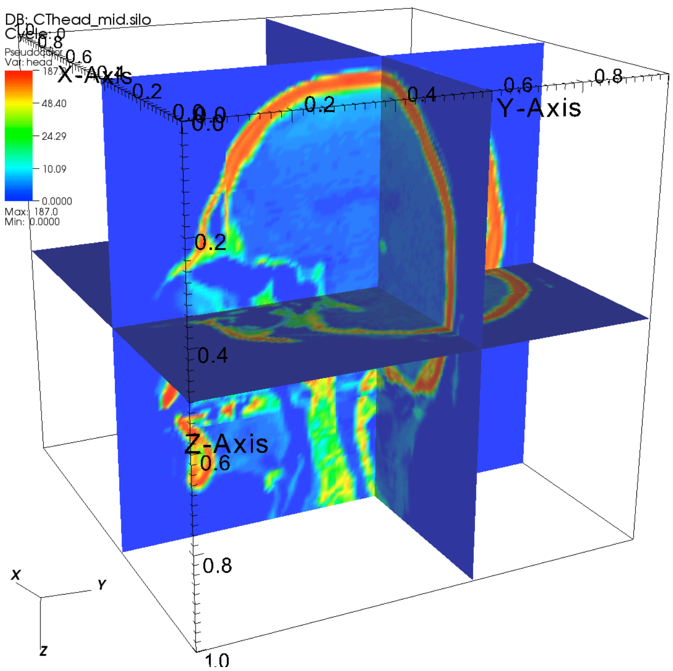


7.

Exercise 5j) ThreeSlice operator (optional)

The ThreeSlice operator computes three axis-aligned slice planes that converge at a point. The ThreeSlice operator is a nice, quick way to see inside 3D plots. In this exercise, we will examine the contents of a CT scan of a head using the ThreeSlice operator.

1. Open *CThead_mid.silo*.
2. Create a Pseudocolor plot of *head*.
3. Open the **Pseudocolor plot attributes** window and make the plot use **Skew** scaling with a **Skew factor** of *0.005*. Be sure to click the **Apply** button in the **Pseudocolor plot attributes** window for the new scaling to take effect.
4. Add a ThreeSlice operator to the Pseudocolor plot.
5. Open the **ThreeSlice operator attributes** window.
6. Type *53* into the **X** text field.
7. Type *75.6* into the **Y** text field.
8. Type *59.5* into the **Z** text field.
9. Click the **Apply** button in the **ThreeSlice operator attributes** window.
10. Click the **Draw** button.
11. Do not delete the plot. Will use it in the first exercise in the next exercise group.



Exercise group 6: Interactive tools

Exercise 6a) Point tool

The point tool lets you move a point around the vis window. Some plots and operators use the point tool's information to set their attributes. In the exercise for the ThreeSlice operator in the last exercise group, we created a plot that gets sliced with the ThreeSlice operator. The ThreeSlice operator's attributes can be set using the Point tool so we are going to turn on the Point tool in this exercise and move the origin of the ThreeSlice operator from the last exercise. If you have not done exercise 5j, do it now.

1. Turn on the point tool by clicking on the **Point tool** button in the **Tools** toolbar. This should make the point tool's hot point visible.
2. Move your mouse into the point tool's hot point, click with the left mouse button, and drag the hot point to the left and release the mouse button. Once you release, VisIt sets the origin for the ThreeSlice operator and regenerates the plot.
3. Repeat step 2 but first hold down the Shift key before you click inside the point tool's hot point. Move the mouse up and down to move the point tool in the direction of the axis that most faces the camera. Moving up moves backwards and moving the mouse down moves the point tool forwards.
4. Rotate the plot using the mouse. Rotation is done the same as it usually is if you keep the mouse outside of the point tool's hot point.
5. Move the point tool some more and get comfortable with moving it around.
6. You can move the point tool in the plane that most faces the camera if you hold down other modifier keys. Holding down the Crtl key moves the point tool up and

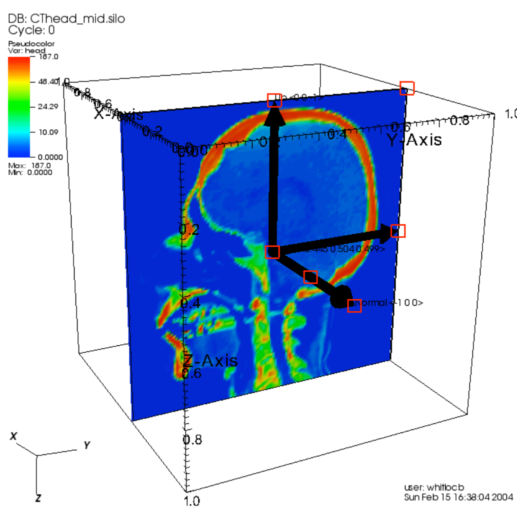
down as you move the mouse up and down. Holding down both Shift and Ctrl keys moves the point tool left and right as you move the mouse left and right.

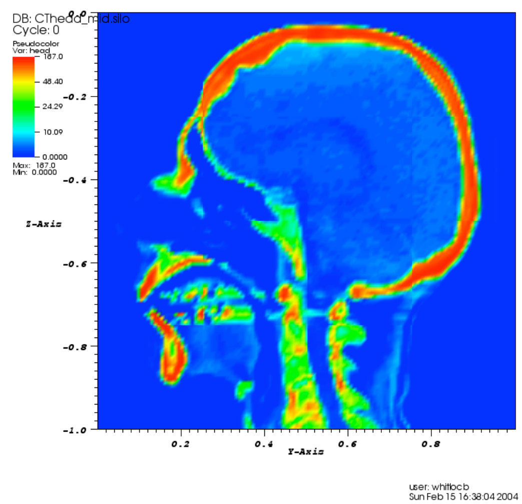
Exercise 6b) Plane tool with multiple windows

The plane tool allows you to interactively position a slice plane or plane source for the Streamline plot. In this example, we will use the Pseudocolor plot from the previous exercise but we will apply a Slice operator to it instead of a ThreeSlice operator. Then we will use the plane tool to set the slice plane used for the Slice operator.

1. In previous exercises, you created a Pseudocolor plot from *CThead_mid.silo* and applied a ThreeSlice operator to it. We will use that same plot for this exercise.
2. Remove the ThreeSlice operator from the Pseudocolor plot by either expanding the plot and clicking the operator delete "X" button or by clicking the **Remove last operator** option in the Operators menu.
3. Open the **Slice operator attributes** window
4. Click on **Orthogonal X Axis** radio button since we want to initially slice along the X axis.
5. Click on the **Point** radio button since we want to slice at a point. Next, type 53 *63 59* into the **Point** text field.
6. Type *0 0 -1* into the **Up Axis Direction** text field. If the **Up Axis Direction** text field is not enabled, skip this step.
7. Turn off the **Project to 2D** axis check box.
8. Click the **Apply** button and answer **Yes** when VisIt asks you to apply the Slice operator.
9. Click the **Draw** button.
10. Turn on the plane tool by clicking on the **Plane tool** button in the **Tools** toolbar.
11. Now try locking tools. When you lock tools, changing a tool in one window sends the new tool attributes to other windows that have also locked their tools. This will allow us to interactively move a slice plane in the first window and have it affect other windows. To lock tools for the window, click on the **Main** window's **Windows** menu option. Then click on the **Lock** submenu under the **Windows** menu. Finally, click on the **Tools** option in the **Lock** submenu.
12. Clone window 1.
13. Switch to the 1x2 window layout by clicking on the **Main** window's **Windows** menu option, and clicking on the **1x2** option under the **Layouts** submenu. This will cause VisIt to create a copy of the first visualization window and then resize both windows to they can be placed side by side. Note that the new window gets the plots and lock setting from the first window.
14. Make vis window 2 be the active vis window.
15. Open the **Slice operator attributes** window if you dismissed it.
16. Turn on the **Project to 2D** check box for the Slice operator. This will only apply to the Slice operator in vis window 2.
17. Click the **Draw** button.

18. Move the plane tool in vis window 1 along its normal axis by clicking in its **Normal translation** hot point and moving the mouse up or down. When you move the slice plane now, both windows should update because their tools are locked.
19. Experiment with the plane tool by moving various hot points and watch both visualization windows update.
20. When you are done, delete the second vis window and switch to a 1x1 window layout so vis window 1 is again full size.



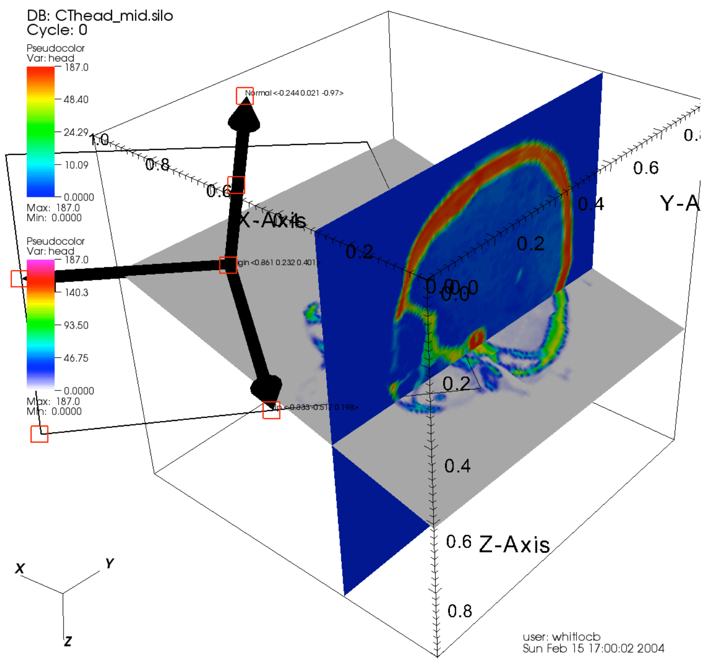


Exercise 6c) Plane tool with multiple plots (optional)

Interactive tools can be used to set the attributes for one plot of many plots in the plot list. In this example, you will learn how to set the slice plane attributes for one plot and not another using the plane tool.

1. This example uses the sliced Pseudocolor plot from the last exercise but it also requires you to create another plot. Add a new Pseudocolor plot of *head*.
2. Open the Pseudocolor plot attributes window and make the new Pseudocolor plot use the *calewhite* color table instead of the *hot* color table. This will make the plots easier to tell apart.
3. Turn off the **Apply operators** check box in the **Main** window since that toggle would interfere with the upcoming instructions.
4. Apply a Slice operator to the new Pseudocolor plot.
5. Open the **Slice operator attributes** window.
6. Turn off the **Project to 2D** check box and click **Apply**.
7. Click the **Draw** button.
8. Now there should be two sliced Pseudocolor plots in the vis window.
9. Move the plane tool. This should set the slice attributes for the second Pseudocolor plot.

10. Click on the first plot entry in the plot list so only the first Pseudocolor plot is selected. This should cause the plane tool to reset to that plot's slice plane.
11. Move the plane tool and watch the first Pseudocolor plot update.
12. Switch the plot selection a few times in the plot list and move the plane tool until you get the hang of how it works to set the slice attributes for individual plots.



Exercise group 7: Subsets

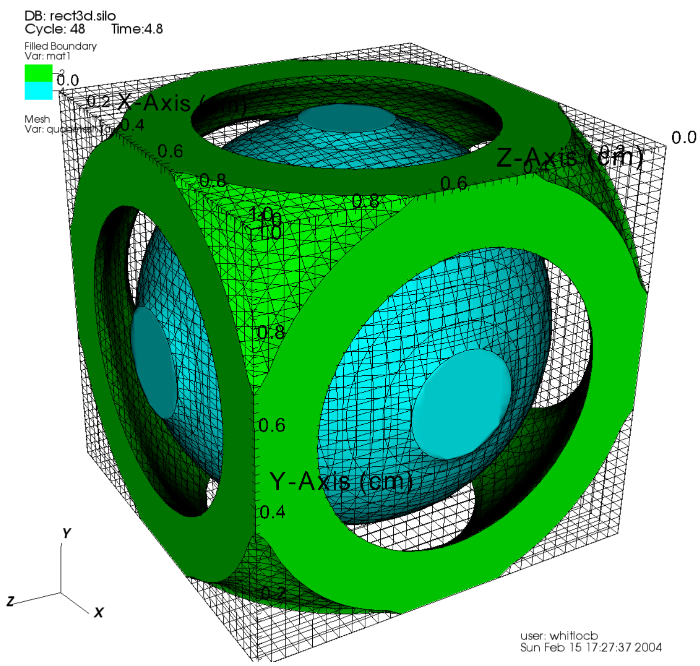
This exercise group focuses on the Subset window and using it to remove subsets from plots.

Exercise 7a) Material selection

This exercise focuses on material selection, which is when you remove materials from a plot.

1. Open *rect3d.silo*.
2. Create a FilledBoundary plot of *mat1*.
3. Click the **Draw** button.
4. Open the **Subset** window by clicking on the **Subset** icon in the plot entry or by clicking on **Subset …** in the **Main** window's **Controls** menu. The window will show the name of the mesh (*quadmesh3d*) on which the material variable *mat1* is defined.
5. Click on *mat1* to show the material subsets in the window's middle pane.

6. Turn off the material *2* subset by clicking on its check box and clicking the **Subset** window's **Apply** button.
7. Turn off the material *4* subset by clicking on its check box and clicking the **Subset** window's **Apply** button.
8. Now reverse the material selection by clicking on the middle pane's **All sets Reverse** button. The **All sets Reverse** button turns on subsets that are off and vice-versa for all sets.
9. Make sure that the **Apply selection to all plots** check box in the **Main** window is off.
10. Add a Mesh plot of *quadmesh3d* and click **Draw**. The new plot should have the same subsets selected as the first plot.
11. Click the **All sets Reverse** button in the middle pane of the **Subset** window to reverse the selection (SIL restriction) for all of the Mesh plot's subsets. Click the **Apply** button. The colored surfaces for the FilledBoundary plot should now never be obscured by any mesh lines since they have opposite SIL restrictions.

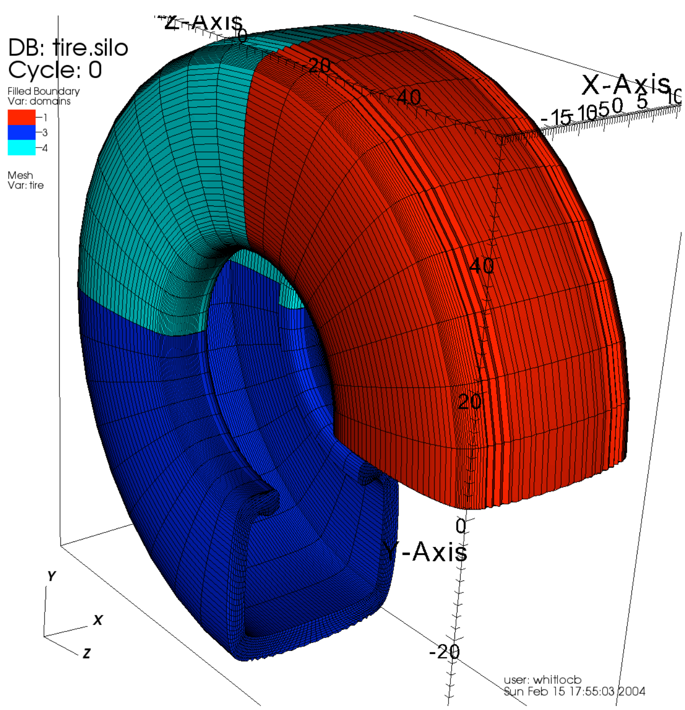


Exercise 7b) Domain selection

This exercise focuses on domain selection, which is when you remove domains from a plot so they are not processed or drawn in the visualization.

1. Open *tire.silo*.
2. Create a Subset plot of *domains*.
3. Create a Mesh plot of *tire* and click the **Draw** button.

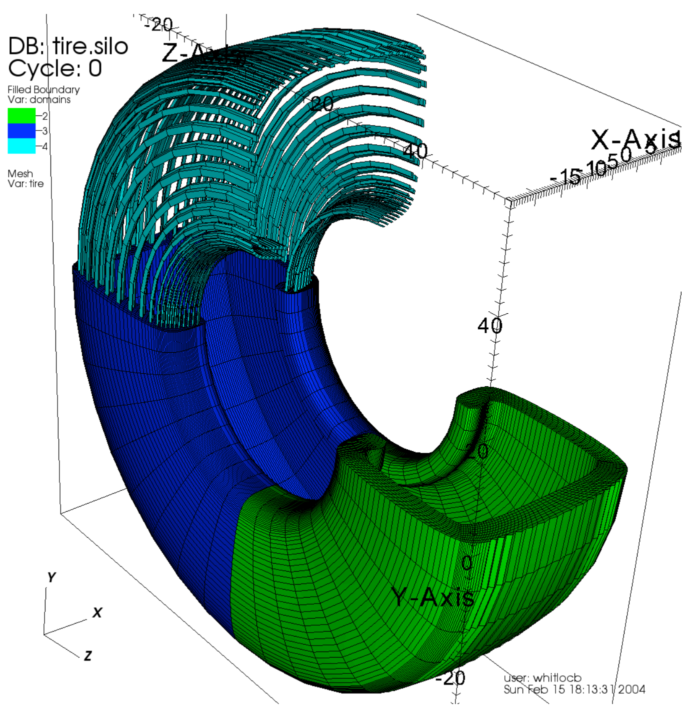
4. Open the **Subset** window. Since the *tire.silo* database has both domain and material subsets, there are two subset categories under *tire*, which is the mesh that is comprised of the domain and material subsets.
5. Make sure that the **Apply selection to all plots** check box in the **Main** window is still off.
6. Select both plot entries in the **plot list**.
7. Click on the *domains* category in the **Subset** window. This should cause the middle pane of the **Subset** window to be filled with a list of the database's domain subsets.
8. Turn off the *domain1* subset by clicking on its check box. Click the **Apply** button. The *domain1* subset for both plots should go away because, although the **Apply selection to all plots** check box is off, we have selected both plots in the plot list and the **Subset** window sets the SIL restriction for the selected plots if the afore-mentioned check box is off.
9. Select the plot entry for the Subset plot in the **plot list**.
10. Turn off the *domain 2* subset and click the **Apply** button. Note that it only turned off the *domain 2* subset for the first plot.
11. Turn on the **Apply selection to all plots** check box in the **Main** window.
12. Turn on the *domain 1* subset and click **Apply**. Notice that *domain 1* turns on and both plots now get the same SIL restriction.



Exercise 7c) Selections involving more than one category

The Subset window allows you to turn on subsets using more than one subset category. This capability lets you turn off both domains and materials at the same time or perhaps turn off material 2 in domain 1. This exercise will show you how to set a plot's SIL restriction using more than one subset category.

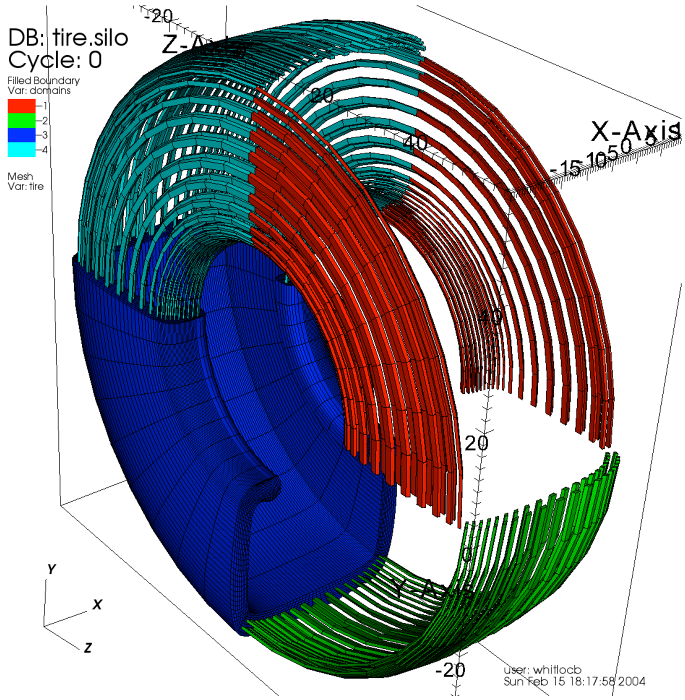
1. Use the plots left over from the previous exercise and open the **Subset** window again if you closed it.
2. Click on the check box next to the *tire* subset in the first panel to turn on all subsets.
3. Click on the *domains* category under *tire* in the first panel. This will cause the *domain* subsets to be listed in the middle panel.
4. Turn off the *domain1* subset.
5. Turn off the *domain4* subset
6. Click on the triangle to the left of the *domain4* subset. This will expand the *domain4* subset and show remaining categories that can be selected. The only option will be *Materials*.
7. Click on the *Materials* category under the *domain4* subset. This will cause the materials for the *domain4* subset to be listed in the third panel.
8. Turn on the *domain4, 2 Steel* and *domain4, 3 Cord* material subsets.
9. Click the **Apply** button.



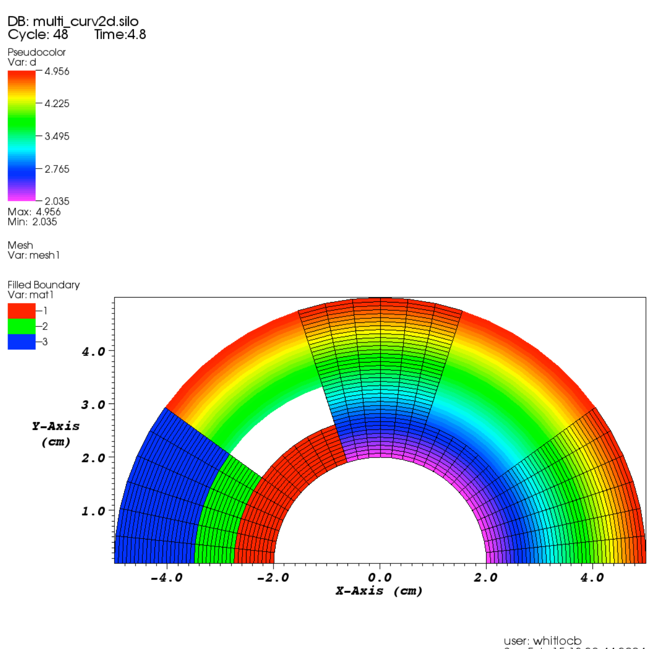
Exercise 7d) Practice creating more complex SIL restrictions

This exercise will give you more practice with creating SIL restrictions.

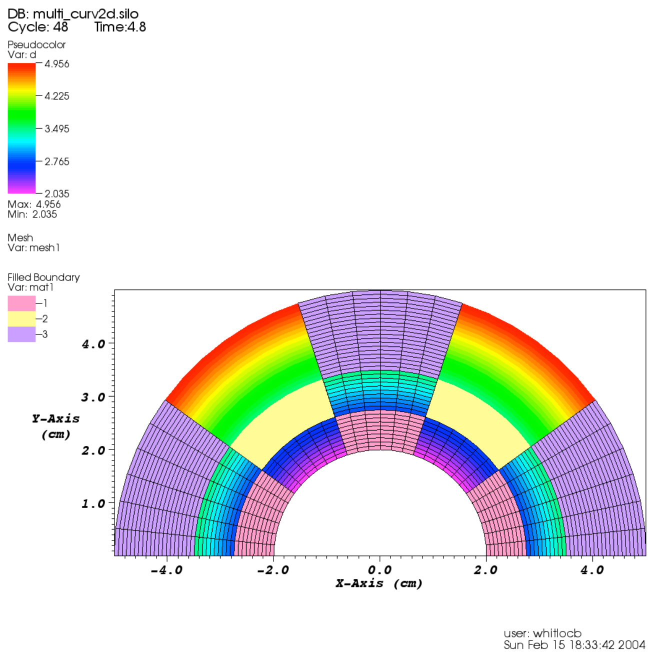
1. Use the plots from the previous exercise but change the SIL restriction so the image looks like the following image:



2. Delete both of the previous plots.
3. Open *multi_curv2d.silo*.
4. Create a Pseudolor plot of *d*.
5. Create a Mesh plot of *mesh1*.
6. Create a FilledBoundary plot of *mat1*.
7. Click the **Draw** button.
8. Set the SIL restriction for each plot until your image looks like this:



9. Change the colors for the FilledBoundary plot to rose, yellow, and lavender.
10. Change the SIL restriction for all plots until your image looks like this:



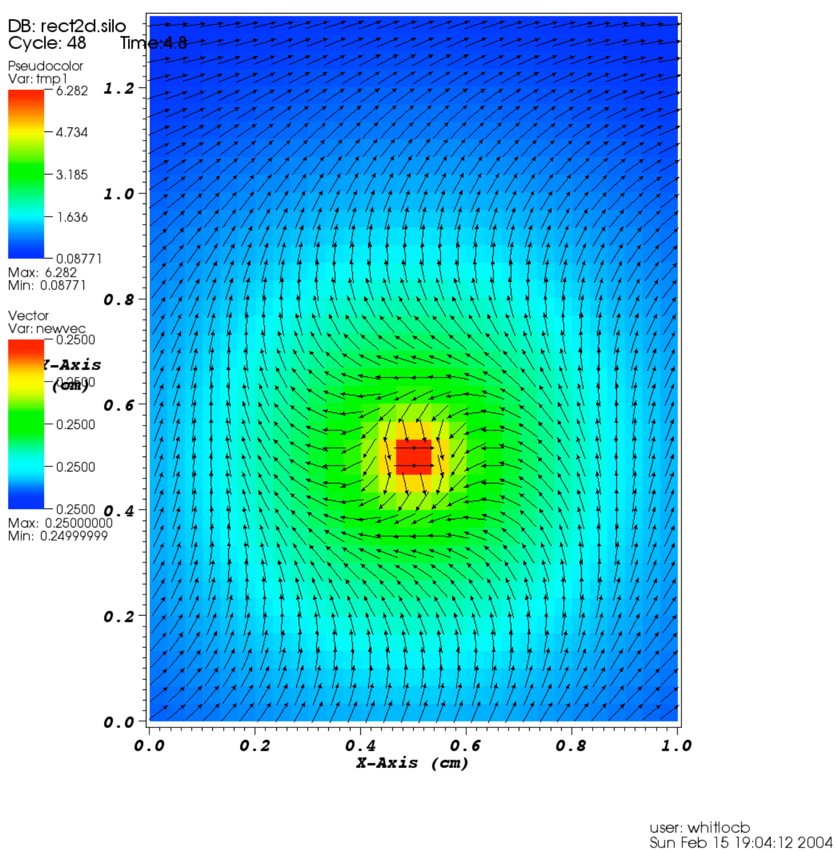
Exercise group 8: Quantitative Analysis

Exercise 8a) Creating expressions

VisIt provides powerful expressions that allow you to create new variables derived from values stored in your database. This exercise focuses on creating a few expressions using values stored in a database and plotting them.

1. Delete any plots that may be in the plot list.
2. Open *rect2d.silo* on your local computer.
3. Open the **Expression** window by selecting the **Expressions …** option from the **Main** window's **Controls** menu.
4. Click the **New** button to create a new blank expression. Change the name of the new expression to *log_p*. Change the definition of the new expression to *log(p).*
5. Click the **New** button to create a new blank expression. Change the name of the expression to *scaled_log_p*. Change the definition of the new expression to *(log_p / 1.626) * 2. * 3.14159. Tip: don't put a period after the 3.14159 when you define your expression.*
6. Click the **New** button to create a new blank expression. Change the name of the expression to *xc*. Change the definition of the new expression to *cos(scaled_log_p).*

7. Click the **New** button to create a new blank expression. Change the name of the expression to *yc*. Change the definition of the new expression to *sin(scaled_log_p)*.
8. Click the **New** button to create a new blank expression. Change the name of the new expression to *newvec*. Change the definition of the new expression to *{xc, yc} * 0.25*. Set the expression type to *Vector Mesh Variable*.
9. Create a Pseudocolor plot of *scaled_log_p* and click the **Draw** button.
10. Create a Vector plot of *newvec*.
11. Open the **Vector plot attributes** window.
12. Click on the **Constant** color button so all vectors will be the same color.
13. Type *0.18* into the **Scale** and **Head size** text fields on the **Form** tab.
14. Click on the **Stride** radio button on the **Location** tab and make sure the stride is *1* so all vectors are shown.
15. Click the **Constant** radio button on the **Rendering** tab and use black as the vector color.
16. Click the **Draw** button to make VisIt plot the new vector variable that we created from a few other derived variables.



Exercise 8b) Pick

VisIt's pick mode allows you to inspect the values for any node or zone in the visible plots by clicking on the nodes or cells that you want to know about. Pick works for 1D, 2D, and for 3D meshes and the variables defined on them but in this exercise, we will focus on 2D picking.

1. Open *noise.silo*.
2. Create a Pseudocolor plot of *hgslice*.
3. Create a mesh plot of *Mesh2D*.
4. Click the **Draw** button.
5. Switch the vis window to node pick mode, which finds the closest node to any place that you click in the window and return the important information for that node. To switch into node pick mode, click on the **Node pick** button in the **Mode** toolbar or click on **Node pick** in the **popup** menu's **Mode** submenu.
6. Click somewhere on the plots in the vis window. VisIt's **Pick** window will appear with information about the pick point that is drawn in the vis window. The information will be about the Mesh plot since that is the plot that is selected in the plot list.
7. Select the Pseudocolor plot in the **plot list**.
8. Click somewhere on the plots in the vis window. The **Pick** window will display more information but this time the information will contain values for the *hgslice* variable since the Pseudocolor plot is selected. In general, select the plot for which you want pick information.
9. Since *Mesh2D* is a structured mesh, it would be nice if VisIt would display the logical node indices of the node that we picked.
10. Open the **Pick** window if it is not already open.
11. Click on the **Domain-Logical Coords** check box for both nodes and zones and click the **Apply** button.
12. Perform some more picks in the visualization window. The new pick information will contain the logical cell and node indices.
13. Switch the vis window into zone pick mode.
14. Perform some more picks in the vis window. The new pick information will contain the values for all nodes in the cell that you picked since *hgslice* is a nodal variable.
15. Now since there are undoubtedly many pick points in the vis window, clear the pick points by choosing **Clear pick points** from the **popup** menu's **Clear** menu.

Exercise 8c) Lineout

Lineout creates a curve by sampling values along a line and then using the values to plot a curve in another vis window. When you put a vis window in lineout mode, any line that you draw ends up generating a new curve. VisIt's lineout mode can only draw lines on 2D plots but there are other ways to create lineout curves for 3D data. This exercise will focus on 2D data.

1. Use the plots from the previous exercise.

2. Put the vis window into lineout mode using the **Mode** toolbar or the mode options in the **popup** menu.
3. Draw a line across the Pseudocolor plot of *hgslice*.
4. VisIt will open a new vis window and plot a curve in it. The curve was created by extracting the data along the line that you drew.
5. Select the **1x2** window layout so both windows can be put side by side.
6. Draw more lines in window 1 to create more curves.
7. Choose **Clear reference lines** from the popup menu's **Clear** menu to clear all of the reference lines in vis window 1.
8. Switch vis window 1 back into navigate mode.

Exercise 8d) Queries

VisIt provides a **Query** window that allows you to query values that can be calculated about a database. The **Query** window can also be used to precisely position pick points and lineouts.

1. Use the plots from the previous exercise.
2. Open the **Query** window by clicking the **Query…** option in the **Main** window's **Controls** menu.
3. Highlight the **2D area** query in the **Queries list** and click the **Query** button to make VisIt calculate the surface area for the currently selected plot in the **plot list**. The answer gets printed to the **Query results** text box in the **Query** window.
4. Highlight the **ZonePick** query in the **Queries** list.
5. Type *-4.3 4.78* into the **Query point** text field and then press the **Query** button. This causes VisIt to add a zone pick point at the specified point.
6. Highlight the **NodePick** in the Queries list.
7. Type *-3.6 3.6* into the **Query point** text field and press the **Query** button. This causes VisIt to return pick information for the node nearest the specified point. Again, the query results are printed in the **Query results** text box.
8. Highlight the **Lineout** query in the **Queries list**.
9. Type *-8.5 8.2 0* into the **Start point** text field.
10. Type *8.6 -9.3 0* into the **End point** text field.
11. Click the **Query** button. VisIt will create a lineout using the specified point and add a curve to vis window 2. Lineouts for 3D variables can be performed through the **Query** window!
12. Delete vis window 2.
13. Delete all of the plots from vis window 1.

Exercise 8e) Plot MinMax Query (optional)

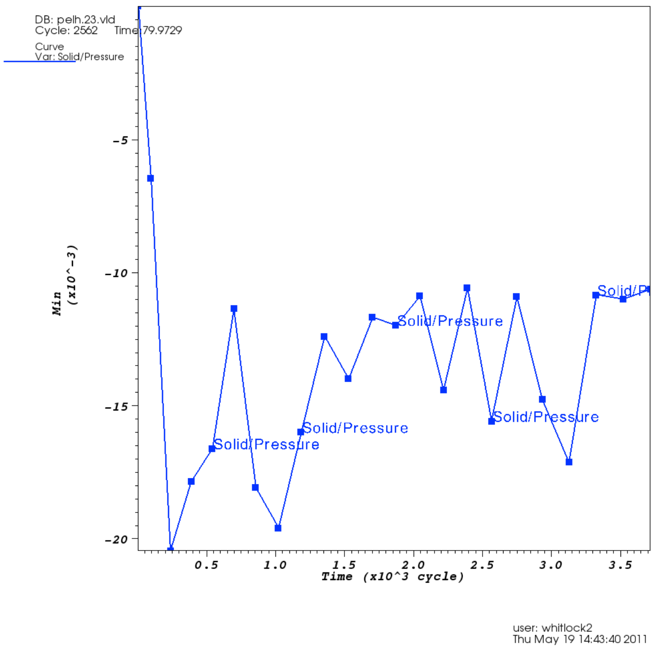
In this exercise, we use the MinMax query for a plot to determine the minimum and maximum values for a plot of a nodal variable. We want to find a zone that contains the node where the highest value occurs and plot only that one zone.

1. Open *noise.silo*.
2. Make a Pseudocolor plot of *hardyglobal* and click the **Draw** button.
3. Open the **Queries** window by selecting **Queries …** from the **Main** window's **Controls** menu.
4. Select the **MinMax** query from the **Queries list**.
5. Click the **Query** button to make VisIt calculate the minimum and maximum values in the plot.
6. Open the **Threshold operator attributes** window and type the maximum value reported by the query into the **Lower bound** text field.
7. Click the **Apply** button and answer **Yes** when VisIt asks if you want to apply the Threshold operator.
8. Delete the plot.

Exercise 8f) Time curve (optional)

In this exercise, we plot the maximum value over time as a curve.

1. Open *VISITCLASSDATA\velodyne\pelh.*.vld database*.
2. Create a Pseudocolor plot of *Solid/Pressure*.
3. Click the **Draw** button.
4. Open the **Query** window and select the *Max* query.
5. Click on the **Time Curve** button to make VisIt calculate the maximum value for each time step and then plot the result as a Curve plot.
6. Make window 2 active.
7. Open the Set save options window from the Main window's File menu.
8. Set the file type to *curve*.
9. Change the **Output directory** to the Desktop. Clicking on the "…" button is the easiest way to select the Desktop.
10. Click the **Save** button to make VisIt write a text file containing XY pairs of the max data over time.
11. Delete window 2.
12. Delete the plots in window 1.



Exercise group 9: Making it pretty

VisIt provides a lot of flexibility for changing the look and feel of plots so they can be beautified for presentations. Often, changing the color table, lighting, or adding annotations can improve the look of a plot and make it more appealing.

Exercise 9a) Color tables

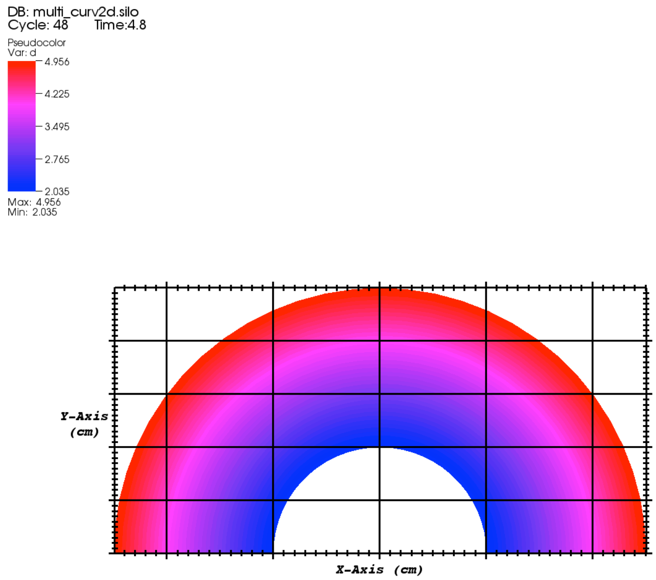
1. Open *multi_curve2d.silo*.
2. Create a Pseudocolor plot of d and click the **Draw** button.
3. Open the **Pseudocolor plot attributes** window.
4. Make the Pseudocolor plot use the **Default** color table, so it will use whichever color table is the active continuous color table. Click the **Apply** button and dismiss the window.
5. Open the **Color table** window by clicking on the **Color table …** option in the **Main** window's **Controls** menu.
6. Change the active continuous color table to *calewhite* by selecting *calewhite* from the list of **Continous color tables** near the top of the **Color table** window. Click the **Apply** button for the change to take effect. Notice how the Pseudocolor plot now uses the new active continuous color table.
7. Change the active continuous color table a few more times.
8. Type *democolortable* into the **Name** text field and then click the **New** button to create a new color table called *democolortable* that is based on the color table that is highlighted in the color table list in the **Manager** area.
9. Change the number of color control points by typing a new number into the **Number of colors** text field and pressing Enter to cause the color table to get a new number of **color control points**.
10. Move the **color control points** around by clicking on them and dragging them to a new location.

11. Change the colors for the **color control points** by right-clicking on them and selecting a new color from the **color palette**. You can also use the **Red**, **Green**, **Blue, Alpha** color sliders to set the color directly for the **active color control point**.
12. To make a **color control point** active, click on it with the mouse.
13. Click on the **Align** button if you want to evenly distribute the **color control points**.
14. Change the active continuous color table to be *democolortable* by selecting *democolortable* from the list of **Continuous color tables**.
15. Click the **Apply** button. The Pseudocolor plot in the vis window should now be using your new color table.

Exercise 9b) 2D Annotation settings

This exercise will familiarize you with the basics of setting 2D annotation options using the controls on the **2D** tab in the **Annotation** window. The controls on the **3D** tab set the annotation options for the 3D axes but since the procedure for setting their attributes is similar to setting the attributes for the 2D axes, we will not have an exercise to set the attributes for 3D axes.

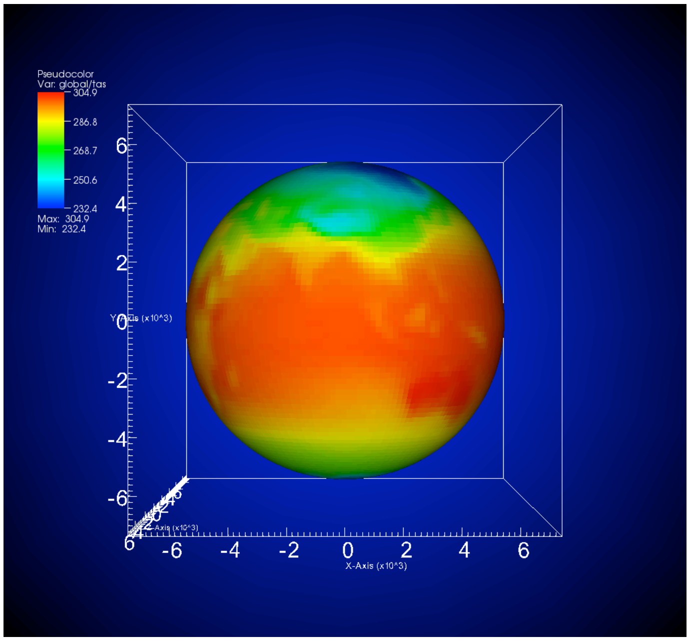
1. Use the plot that was set up in the previous exercise.
2. Open the **Annotation** window by clicking the **Annotation …** option in the **Main** window's **Controls** menu.
3. Click on the **2D** tab to see the controls for setting 2D annotations, which mainly allow you to set the attributes for the 2D plot axes.
4. Turn off the axes by clicking off **Show axes** and clicking the **Apply** button. Notice that the axes turn off in the vis window. Turn the axes back on.
5. Turn off the axis labels by clicking the **Title** and **Labels** check boxes on the **TX-Axis** and **Y-Axis tabs.** Click the **Apply** button. Note that the axis labels disappear but the axis titles are still visible.
6. Turn on grid lines for both X and Y dimensions by clicking the **Show grid** check box on the tabs for both the X and Y axes. Click the **Apply** button.
7. Choose a thicker line width to use for drawing the axes by picking a new line width from the **Line width** combo box on the **General 2D** tab. Click the **Apply** button.



Exercise 9c) Setting window colors

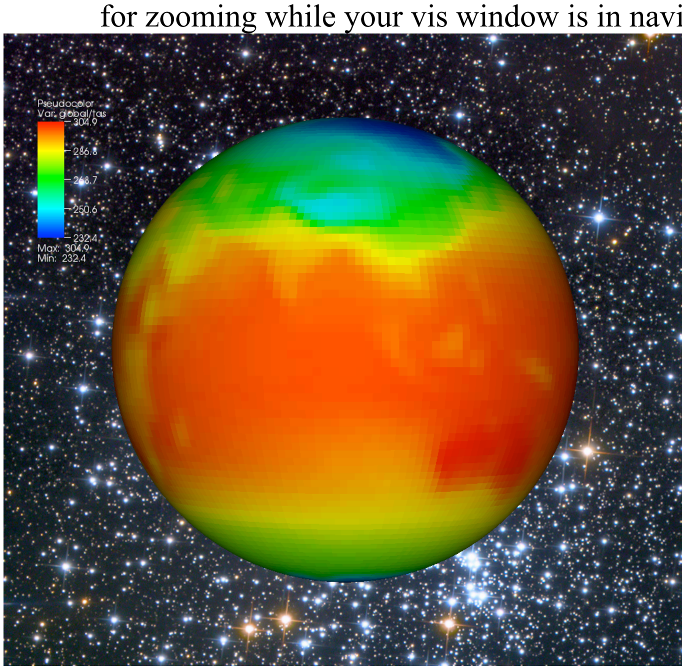
The **Colors** tab in the **Annotation** window allows you to set the background and foreground colors for the active vis window. Color selection is very crucial in the creation of a presentation quality image. The default window colors, while good for everyday use, are not very good for presentations because they are flat and do not help to convey depth. VisIt provides more interesting gradient color backgrounds that instantly improve the quality of an image that will be used in presentations.

1. Delete the plot from the previous exercise.
2. Open *tas_mean_T63.nc*
3. Create a Pseudocolor plot of *global/tas* and click the **Draw** button.
4. Click the **Gradient** radio button on the **Colors** tab of the **Annotation** window so you can change the gradient background settings. Click the **Apply** button. This should change the background to a radial blue and black gradient.
5. Change the **Foreground** color to white and click the **Apply** button.
6. Experiment with different gradient styles by choosing a new gradient style from the **Gradient style** combo box and clicking the **Apply** button. At the same time, experiment with new gradient colors.



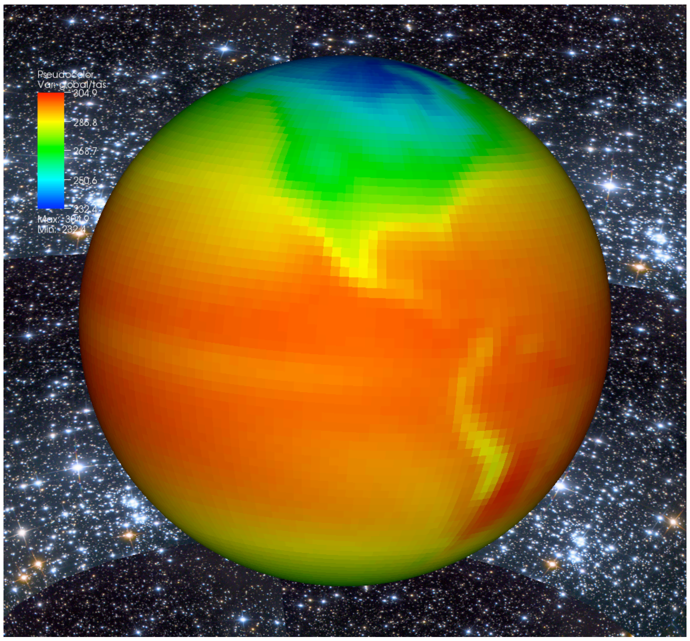
Experiment with image backgrounds.

7. Click on the **Image** radio button to select an image background.
8. Click on the "…" button for the **Background image** text field and select *VISITCLASSDATA\starfield.jpg.*
9. Turn off **Show axes** and **Show bounding box** check boxes on the **3D** tab.
10. Zoom in on the plot a little. If your mouse has a scroll wheel, you can use that for zooming while your vis window is in navigate mode.



Now turn the image background into an Image sphere background.

11. Click on the Image sphere radio button on the Colors tab.
12. Set the Repetitions in X to 4.
13. Set the Repetitions in Y to 4.
14. Rotate the plot around and watch the stars track with the camera.

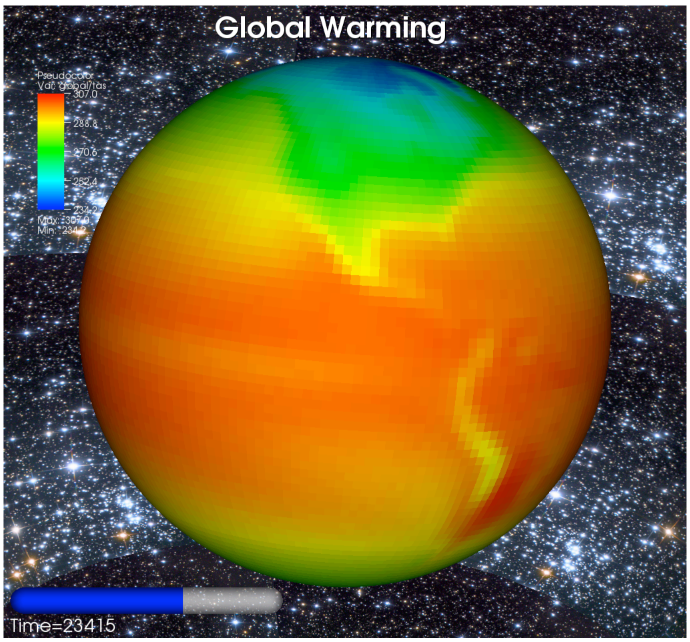


Exercise 9d) Annotation objects

VisIt provides four simple annotation objects that can be used to improve the quality of an image destined for a presentation. The first annotation object is a time slider that is useful for showing the progress through time in an animation. The second annotation object is a 2D text object that can be placed anywhere in the vis window and can display arbitrary text. The 2D text annotation is especially useful for adding classification banners to an image. The third annotation lets you show images such as graphs or project logos. The fourth annotation is a line/arrow annotation that can be used to point out features in a visualization.

1. Use the plot from the previous exercise
2. Click on the **Objects** tab in the **Annotation** window.
3. Click on the **Time slider** button to create a new time slider object. Click the **Apply** button and look for the time slider in the lower left corner of the annotation window.
4. Increase the height of the time slider to 7% by incrementing the value in the **Height** spin box a few times. Click the **Apply** button.
5. Select a new start color by clicking on the **Start color** button and choosing a new color from the color palette. Click the **Apply** button.
6. Click the **Text** button to create a new 2D text annotation object. Note that the window changes so the attributes for the 2D text annotation show instead of the attributes for the time slider.

7. Type *Global Warming* into the **Text** text field.
8. Click the **Bold** check box to make the text be bold.
9. Click the **Shadow** check box to make the text have a slight shadow under it.
10. Click the **Apply** button.
11. Type *0.31 0.95* into the **Lower** left text field and click the **Apply** button to move the annotation to the upper middle of the vis window.
12. Change the width to be *34%* using the **Width** spin button and click the **Apply** button.
13. Use the **Animation slider** to change the time so you can see the time slider annotation update a few times.
14. Use the **Animation slider** to get back to the first time state.

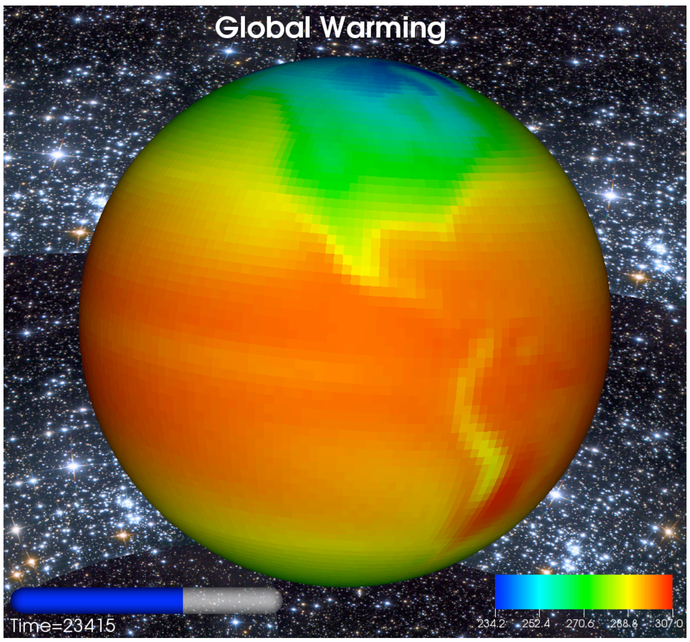


Exercise 9e) Legends (optional)

If you want to include VisIt's legends in your visualization then you may want to customize them or change where they appear. This exercise shows how to make basic changes to the legend.

1. Use the plots from the previous exercise.
2. Click on the **Objects** tab in the **Annotations** window.
3. Select the Legend annotation object for the Pseudocolor plot of *global/tas*. This will make the bottom half of the window change so it displays legend properties.
4. Change the legend to be horizontal by selecting **Horizontal, Text on Bottom** for the legend **Orientation** on the **Position** tab.

5. Move the legend to the lower right part of the vis window by turning off the **Let VisIt manage legend position** check box. Then type *0.72 0.10* into the **Legend position** text box.
6. Set the legend's **Y-scale** to *60%* on the **Position** tab.
7. Turn off some extra titles by turning off the **Draw title** and **Draw min/max** check boxes on the **Appearance** tab.
8. Set the **Font height** to *0.03* to give somewhat larger labels.
9. When you are done, click the **Reset** button in the **Annotation** window.
10. Delete your plots.



Exercise 9f) Lights (optional)

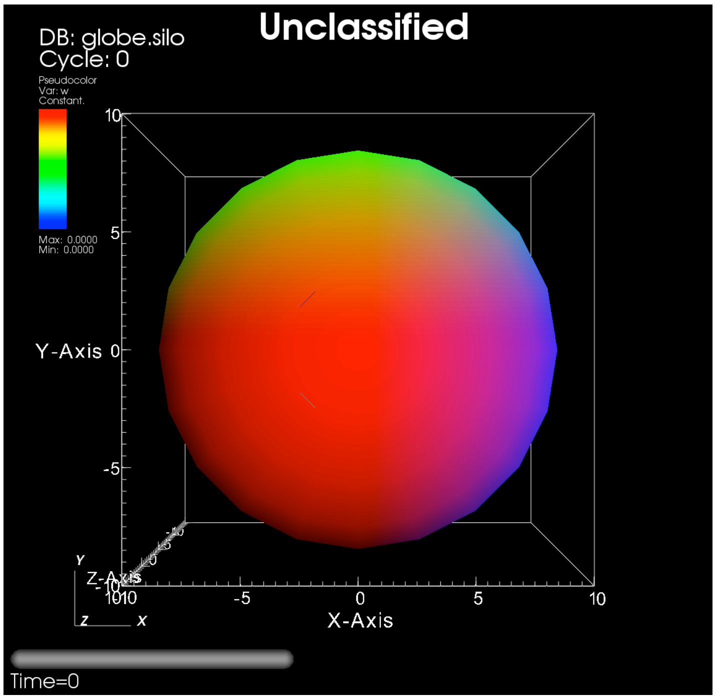
Lights can make a lot of difference in the brightness of a plot that uses lighting like the Pseudocolor plot. If a plot is too dark, changing the light or adding another light can vastly improve the plot's appearance. This exercise is not so concerned with making a plot look better as it is with showing you how to use the lighting controls so we will use multiple colored lights.

1. Open *globe.silo* on your local computer.
2. Create a Pseudocolor plot of *w*.
3. Open the **Pseudocolor plot attributes** window and make the Pseudocolor plot use the *xray* color table.
4. Click the **Apply** button.
5. Open the **Annotation** window, click the **Colors** tab, and click the **Solid** radio button so the vis window will use a solid background color. Change the background color to black and click the **Apply** button. Dismiss the **Annotation** window. The vis window should now have a big white globe on a black background.

6. Reset the view using the **Reset view** toolbar button in the vis window.
7. Turn on auto apply mode in the **Main** window by clicking the **Auto apply** check box above the plot list. This will eliminate the need to keep clicking the **Apply** button and will work better for designing lights, which is largely a trial and error exercise.
8. Open the **Lighting** window by clicking on the **Lighting …** option in the **Main** window's **Controls** menu.
9. The **Lighting** window will have on its left side an image (the test sphere) that looks similar to the plots that we've set up. In fact, we set the plots up to look like the test sphere in the lighting window so you can see more easily how lighting affects a plot. Nestled in with the test sphere, there is a small blue arrow that you can drag around to change the lighting. Drag the small blue arrow and release it and watch lighting on the test sphere change. Since **Auto update** mode is on, the vis window will also update.
10. Move the blue arrow a few times to see what happens.
11. Type *0 0 -1* into the **Direction** text field and press Enter. This will reset the light so it shines into the scene. Change the color to red.
12. Select light 2 by clicking on 2 in the Active light combo box.
13. Type *0 -1 0* into the **Direction** text field and press the Enter key.
14. Change the light's color to green.
15. Turn the light on by clicking the **Enabled** check box. The vis window should have a mostly red sphere with a soft green highlight coming down from above.
16. Select light 3 by clicking on 3 in the **Active light** combo box.
17. Type *-1 0 0* into the **Direction** text field and press the Enter key.
18. Change the light's color to blue.
19. Turn the light on by clicking the **Enabled** check box. The vis window should have a mostly red sphere with a soft green highlight coming down from above and a soft blue highlight coming from the right.
20. Rotate the plot in the vis window. Notice how the highlights never seem to change because the lights are fixed in space because they are camera lights.
21. Change the blue light to be an object light by selecting *Object* from the **Light type** combo box.
22. Make light 2 (the green light) be the active light by selecting 2 from the **Active light** combo box.
23. Change the light type to *Object*.
24. Make light 2 (the red light) be the active light by selecting 1 from the **Active light** combo box.
25. Change the light type to *Object*.
26. Click on the **Preview** radio button in the **Lighting** window to show all lights and drag the mouse over the test sphere to rotate the test sphere. Note that the lights are now fixed to the object.
27. Rotate the plot in the vis window. The behavior will be the same as for the **Lighting** window's test sphere
28. Click the **Reset** button in the **Lighting** window to restore the default light attributes, which are the light attributes that were last saved. Since we never

clicked the **Make default** button for lights, this will restore VisIt's lighting to use a single white camera light.

29. Turn off **Auto apply** in the **Main** window.
30. Delete the plot.



Exercise Group 10: Animation and Keyframing

Exercise 10a) Session files

VisIt session files save all of the information necessary for VisIt to create the same plots later.

1. Open *wave.visit* on your local computer.
2. Create a Pseudocolor plot of *pressure*.
3. Create a Mesh plot of *quadmesh*.
4. Click the **Draw** button.
5. Open the **Annotation** window and make the vis window have a gradient background and change the vis window foreground color to white.
6. Save a session file by choosing **Save session …** from the **Main** window's **File** menu. You will be prompted for the name of the session file so save the session file as *wave.session*.
7. Quit VisIt.
8. Start VisIt again.
9. Click **Restore session …** from the **Main** window's **File** menu. Locate and select the *wave.session* session file that you previously saved. Once you select the session file, VisIt will read in the session file and restore everything to where you were when you saved the session file.

Exercise 10b) Movie generation using session files

VisIt provides a utility for generating movies called visit –movie. If you open up a command line shell, you can execute visit –movie with the name of a session file and VisIt will create a movie of the plots described by the session file. In Windows, VisIt session files can be used to automatically generate movies. If you right click on a session file, there are options to generate movies at different resolutions. If you choose to generate a movie like this, Windows runs visit –movie with the session file that you clicked on.

Windows users:

1. Open a **Command prompt** window by choosing **Command prompt** window from the **Accessories** menu in the **Programs** menu in the **Windows Start** menu. Another way to find it is to type *cmd* into the Windows search box.
2. Type *cd C:\Program files\LLNL\VisIt 2.2.2* to change the active directory to the directory where VisIt was installed. If that is not the right path for your system, substitute the right path.
3. Type *visit –movie –sessionfile wave.session –geometry 800x800 –format mpeg* to generate an 800 pixel * 800 pixel MPEG movie of the plots described by the *wave.session* session file.
4. Open **My Computer** on the Windows desktop or in the **Windows Start** menu. Navigate to *C:\Program files\LLNL\VisIt 2.2.2* where VisIt was installed. This is where VisIt session files are saved by default.
5. Right click on *wave.session* and choose the **Generate 480x480 movie** option. This will cause VisIt to generate an MPEG movie for the visualization described by the *wave.session* session file.

UNIX and MacOS X users:

1. Go to a command prompt window.
2. Type *cd ~/.visit* to change the active directory to where your session file is located.
3. VisIt may have been installed elsewhere or you may have it in your path and have no need to specify its full path but type the command that you run to launch visit with the following arguments: *-movie –sessionfile wave.session –geometry 1000x1000 –format mpeg*. This will generate an 800 pixel * 800 pixel MPEG movie for the plots described by the *wave.session* file.

Exercise 10c) Making a movie with the Save movie wizard

VisIt's GUI provides a **Save movie** wizard that allows you to save a movie of your current visualization without having to enter any cryptic commands. The wizard also has the benefit that you can use the same compute engine to make your movie that you've used to set up your visualization. This can reduce your wait time if you run in an environment where processors are scarce.

1. Restore the *wave.session* session file using the **Restore session** option in the **File** menu.
2. Open the **Save movie** wizard from the **File** menu.
3. Choose **New simple movie** and click the **Next** button.
4. Choose **MPEG** format and **Use current window** size.
5. Click the **Right arrow** button to add a request for MPEG movies to the **Output** list.
6. Let's also make some smaller resolution frames. Choose **JPEG** images format.
7. Click the **Specify movie size** radio button.
8. Enter *400* into the **Width** text field and press the *Enter* key. This will make the height update to a value that preserves the shape of the visualization window.
9. Click the **Right arrow** button to add the JPEG frames to the **Output** list.
10. Click the **Next** button.
11. Choose the **No** radio button when the wizard asks whether you would like to make a stereo movie and click the **Next** button.
12. Choose a movie name and output directory or leave the default values and click the **Next** button.
13. Choose **Now, use currently allocated processors** and click the **Finish** button to make VisIt generate your MPEG movie and JPEG images.

Exercise 10d) Keyframing

VisIt allows you to set plot, view, and database keyframes for your animation. This exercise will familiarize you with setting plot attribute and view keyframes.

1. Open the **Animation** window by clicking **Animation …** in the **Main** window's **Controls** menu.
2. Click the **Cache animation for faster playback** check box so VisIt caches the animation while generating it so when the animation starts playing again from the beginning, it plays faster because it does not have to be regenerated by the compute engine.
3. Open *wave.visit*.
4. Create a Pseudocolor plot of *pressure*.
5. Create a Mesh plot of *quadmesh*.
6. Click the **Draw** button.
7. Open the **Keyframe** window by clicking on the **Keyframing …** option in the **Main** window's **Controls** menu.
8. Turn on Keyframing by clicking the **Keyframing enabled** check box. Click the **Apply** button in the **Keyframe** window.
9. Select the Mesh plot in the **Plot list** and double click it to open the **Mesh plot attributes window**.
10. Make the mesh plot use yellow lines. Turn off the **Use foreground** check box and change the **Mesh color** to yellow. Click the **Apply** button.
11. Rotate the view a little in the vis window.

12. Turn on the **Use view keyframes** check box in the **Keyframe** window. Click the **Apply** button.
13. Click the **Add view keyframe** button in the **Keyframe** window to set a view keyframe.
14. Since we're keyframing a database with multiple time states, VisIt will have more than one possible time slider: a time slider for *wave.visit* and a time slider for the keyframing frame. The active time slider should be the keyframing time slider. Drag the **Animation slider** to frame 37.
15. Double click on the Pseudocolor plot's plot entry in the **plot list**. This will select the Pseudocolor plot and also open the **Pseudocolor plot attributes** window. Change the plot's opacity to 10% and click the **Apply** button. This will set a plot attributes keyframe for the Pseudocolor plot at the current animation frame.
16. Rotate the plot a little in the vis window and zoom in a little too.
17. Click the **Add view keyframe** button in the **Keyframe** window to set a view keyframe at the current animation frame.
18. Drag the **Animation slider** to frame 70.
19. Double click the Mesh plot's plot entry in the **plot list**. When the **Mesh plot attributes** window opens, change its Mesh color to be red. Make the plot use a heavier line width too and click the **Apply** button to set a plot attributes keyframe for the Mesh plot.
20. Double click the Pseudocolor plot's plot entry in the **plot list**. When the **Pseudocolor plot attributes** window opens, set the plot's opacity to 100% and click the **Apply** button.
21. Rotate the plot a little in the vis window.
22. Click the **Add view keyframe** button in the **View** window to set a view keyframe.
23. Save a session file called *keyframe.session* so you could restart VisIt later and get back all of the setup that you've just done.
24. Click the **Play** button and watch the animation generate and play.
25. Once you've watched the animation enough, click the **Stop** button and drag the **Animation slider** back to frame 0. Delete all of the plots.
26. Disable keyframing in the **Keyframe** window.

Exercise 10e) Executing Scripts (optional)

VisIt's Python command line interface (CLI) is a Python interpreter that imports the VisIt Python module when it starts up. This allows what is an ordinary Python interpreter to control VisIt via scripting by calling functions that give VisIt instructions to open databases, create plots, etc. You can enter Python commands into VisIt's **Command** window in order to execute Python scripts.

1. Open *noise.silo*.
2. Create a Pseudocolor plot of *grad_magnitude*.
3. Add a Slice operator and turn off its **Project to 2D** check box.
4. Click the **Draw** button.
5. Now for a small demonstration of why scripting is so powerful. Type the following code into the Command window:

```
s = SliceAttributes()
s.originType = s.Percent
for i in xrange(0,100,5):
     s.originPercent = i
     SetOperatorOptions(s)
```

6. Click the **Execute** button in the **Command** window to make VisIt interpret the script.
7. The above code should make a slice plane move through the data in 3D.
8. To make the slice plane move through each dimension, type the following code at the Python prompt:

```
normals = ((1,0,0), (0,1,0), (0,0,1))
for n in normals:
     s.normal = n
     for i in xrange(0,100,2):
         s.originPercent = i
         SetOperatorOptions(s)
```

9. Click the **Execute** button in the **Command** window to make VisIt interpret the script.