



Large Scale Visualization with ParaView

ParaView 5.6

Kenneth Moreland Sandia National Laboratories



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).



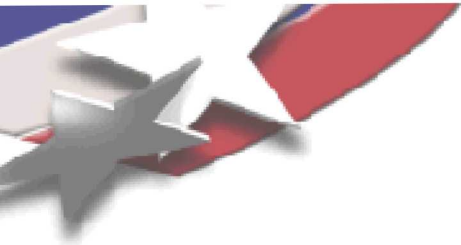
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

SAND 2017-9244 PE



Outline

- Introduction
- Basic Usage
- Batch Scripting
- Visualizing Large Models
- Catalyst

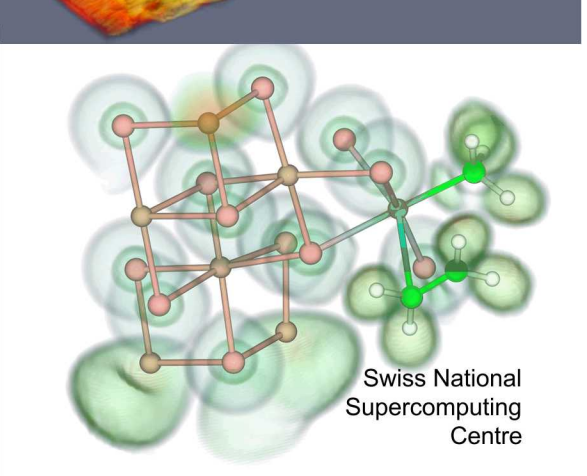
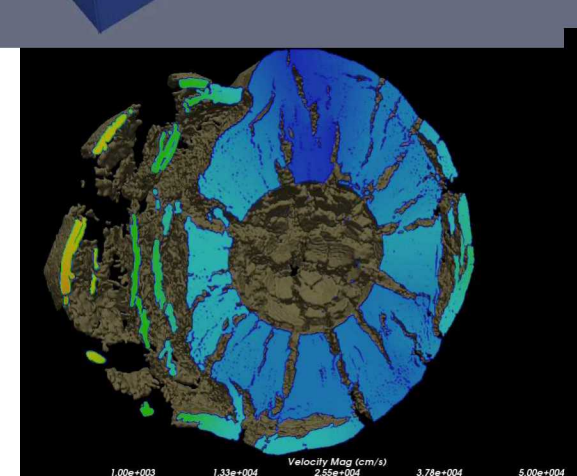
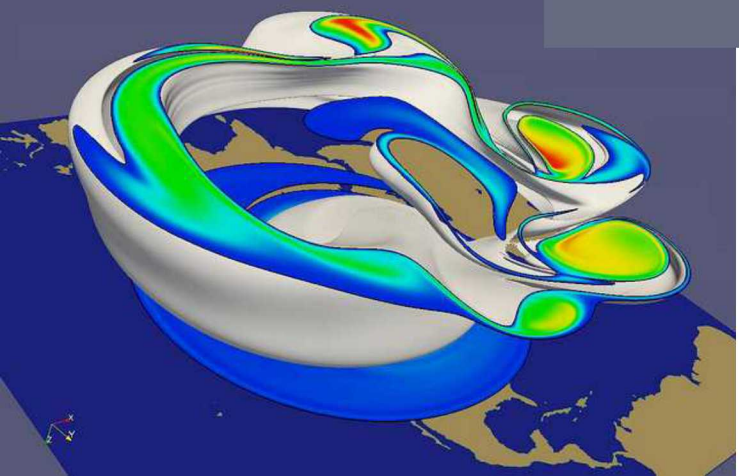
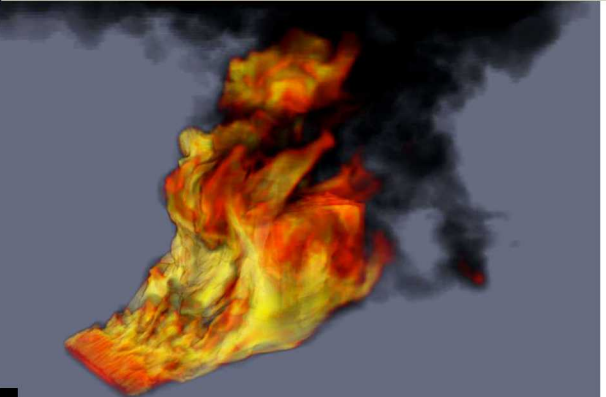
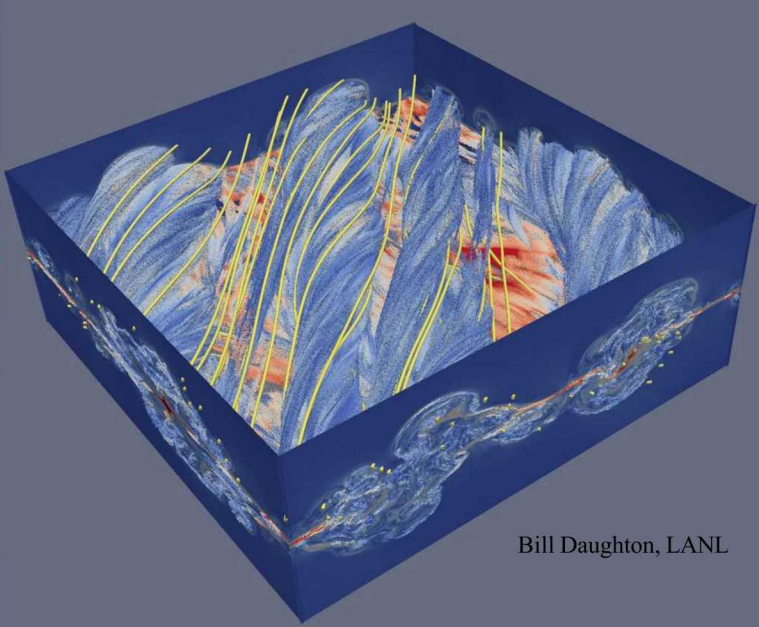
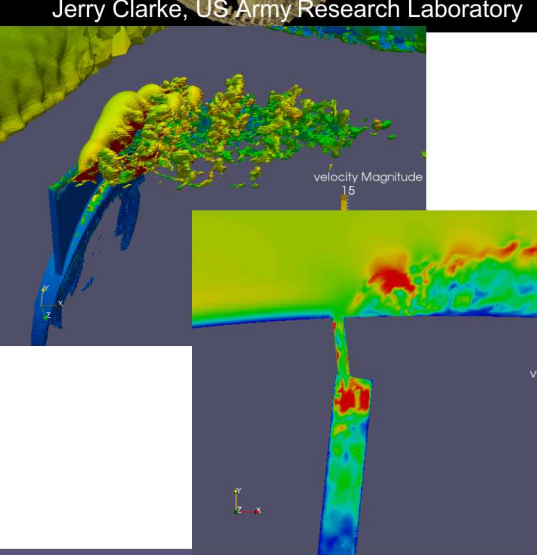
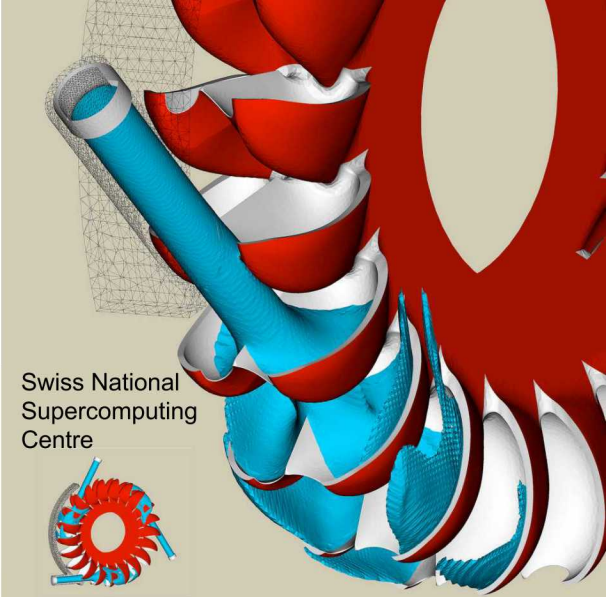
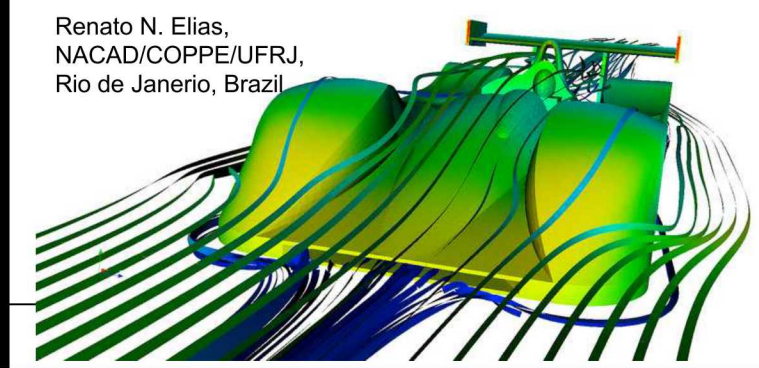
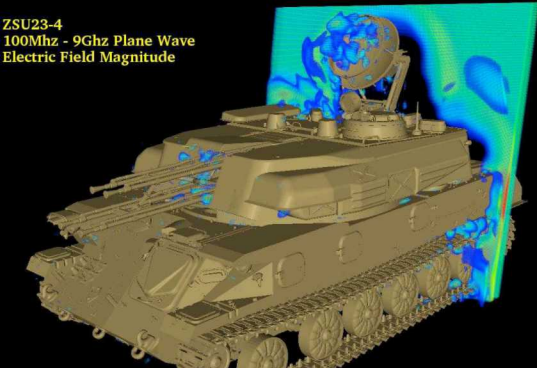


Introduction



What is ParaView?

- An open-source, scalable, multi-platform visualization application.
- Support for distributed computation models to process large data sets.
- An open, flexible, and intuitive user interface.
- An extensible, modular architecture based on open standards.
- A flexible BSD 3 Clause license
- Commercial maintenance and support.



In the top-left corner, there are several overlapping, semi-transparent stars in shades of red, white, and grey.

Current ParaView Usage

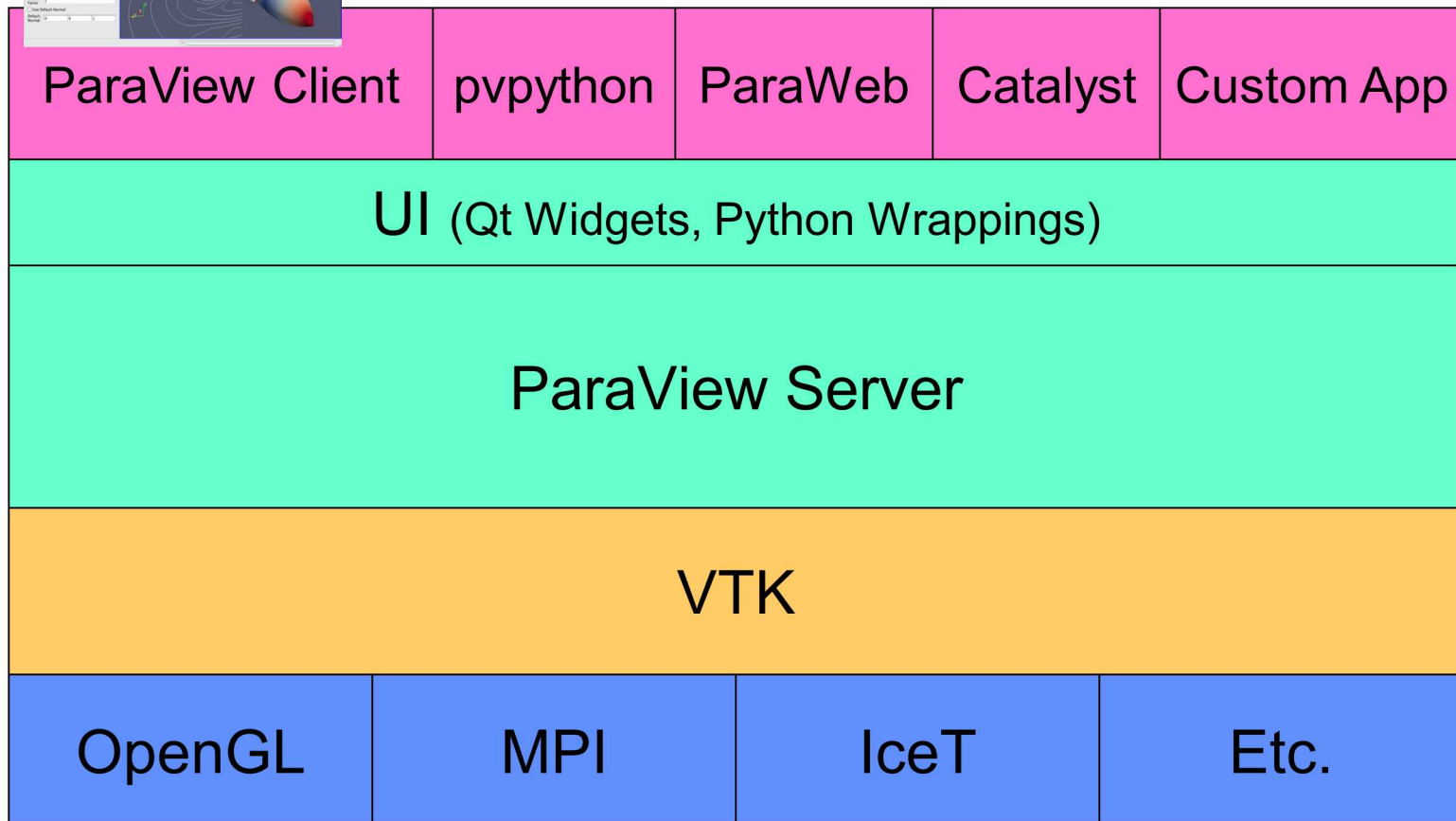
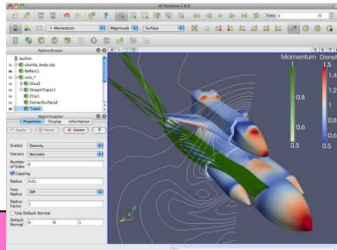
- Used by academic, government, and commercial institutions worldwide.
- Downloaded ~135K times per year.
- HPCwire Editors' Choice 2010/2016 and HPCwire Readers' Choice 2010/2012/2015 Awards for Best Visualization Product or Technology.



Data Ranges

- Used for all ranges of data size.
- Landmarks of usage:
 - 6 billion structured cells (2005).
 - 250 million unstructured cells (2005).
 - Billions of AMR cells (2008).
 - Scaling test over 1 Trillion cells (2010).
 - 6.33 billion unstructured cells in Catalyst (2016).

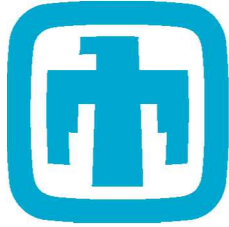
ParaView Application Architecture



ParaView Development

- Started in 2000 as collaborative effort between Los Alamos National Laboratories and Kitware Inc. Sandia has been a major contributor since 2005.
 - ParaView 0.6 released October 2002.
- Paraview 3.0 release in May 2007.
 - GUI rewritten to be more user friendly and powerful.
- ParaView 4.0 released in June 2013.
 - Properties panel redesign for smoother interaction.
- ParaView 5.0 released in January 2016.
 - Updated to OpenGL 3.2 features. Huge performance improvements.

Current Funding



Sandia
National
Laboratories



EXASCALE COMPUTING PROJECT

This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

- ARL
- ERDC
- US Army (SBIR)
- US Air Force (STTR)
- ONR
- Support Contracts
 - Electricity de France
 - Microsoft

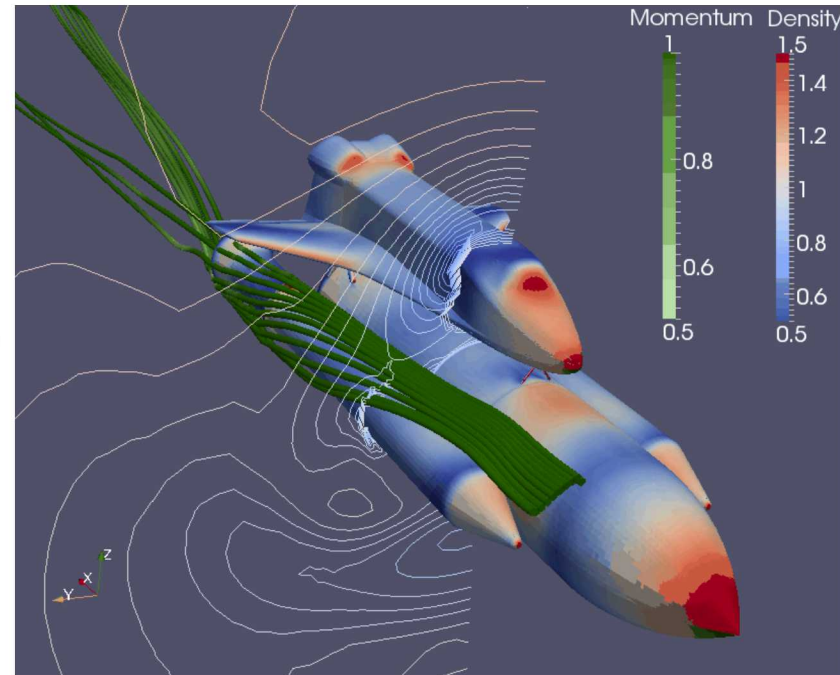
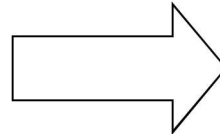
- Other contributors
 - Swiss National Supercomputing Centre
 - DOE SLAC
 - Ohio State
 - Mississippi State
 - RPI

Basics of Visualization

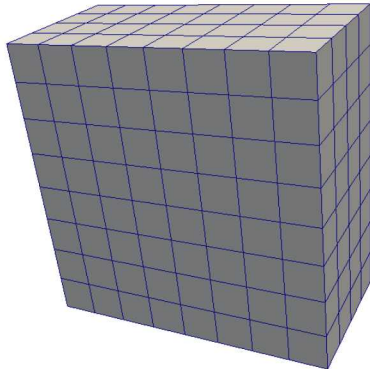
```

0265640 132304 133732 032051 037334 024721 015013 052226 001662
0265660 025537 064663 054606 043244 074076 124153 135216 126614
0265700 144210 056426 044700 042650 165230 137037 003655 006254
0265720 134453 124327 176005 027034 107614 170774 073702 067274
0265740 072451 007735 147620 061064 157435 113057 155356 114603
0265760 107204 102316 171451 046040 120223 001774 030477 046673
0266000 171317 116055 155117 134444 167210 041405 147127 050505
0266020 004137 046472 124015 134360 173550 053517 044635 021135
0266040 070176 047705 113754 175477 105532 076515 177366 056333
0266060 041023 074017 127113 003214 037026 037640 066171 123424
0266100 067701 037406 140000 165341 072410 100032 125455 056646
0266120 006716 071402 055672 132571 105645 170073 050376 072117
0266140 024451 007424 114200 077733 024434 012546 172404 102345
0266160 040223 050170 055164 164634 047154 126525 112514 032315
0266200 016041 176055 042766 025015 176314 017234 110060 014515
0266220 117156 030746 154234 125001 151144 163706 136237 164376
0266240 137055 062276 161755 115466 005322 132567 073216 002655
0266260 171466 126161 117155 065763 016177 014460 112765 055527
0266300 003767 175367 104754 036436 172172 150750 043643 145410
0266320 072074 000007 040627 070652 173011 002151 125132 140214
0266340 060115 014356 015164 067027 120206 070242 033065 131334
0266360 170601 170106 040437 127277 124446 136631 041462 116321
0266400 020243 005602 004146 121574 124651 006634 071331 102070
0266420 157504 160307 166330 074251 024520 114433 167273 030635
0266440 133614 106171 144160 010652 007365 026416 160716 100413
0266460 026630 007210 000630 121224 076033 140764 000737 003276
0266500 114060 042647 104475 110537 066716 104754 075447 112254
0266520 030374 144251 077734 015157 002513 173526 035531 150003
0266540 146207 015135 024446 130101 072457 040764 165513 156412
0266560 166410 067251 156160 106406 136770 030516 064740 022032
0266600 142166 123707 175121 071170 076357 037233 031136 015232
0266620 075074 016744 044055 102230 110063 033350 052765 172463

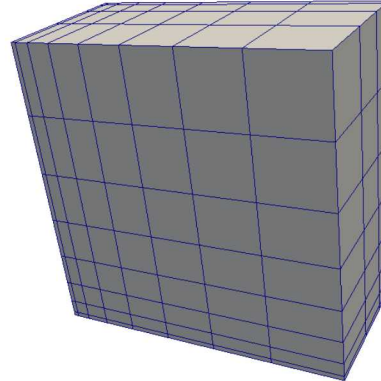
```



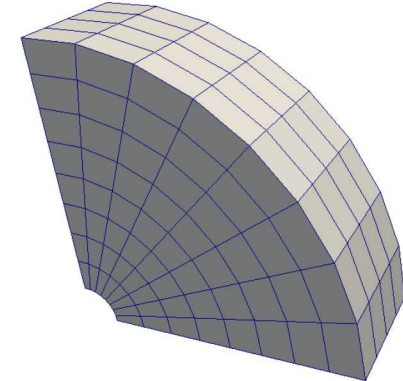
Data Types



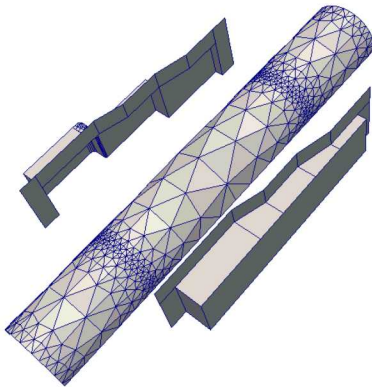
Uniform Rectilinear
(`vtkImageData`)



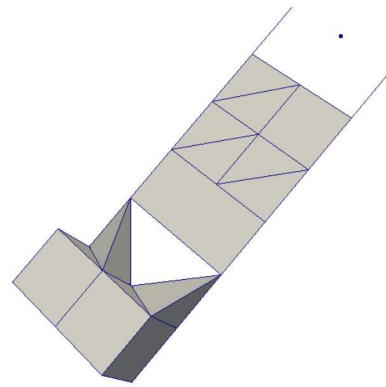
Non-Uniform Rectilinear
(`vtkRectilinearData`)



Curvilinear
(`vtkStructuredData`)



Polygonal
(`vtkPolyData`)



Unstructured Grid
(`vtkUnstructuredGrid`)

Multi-block

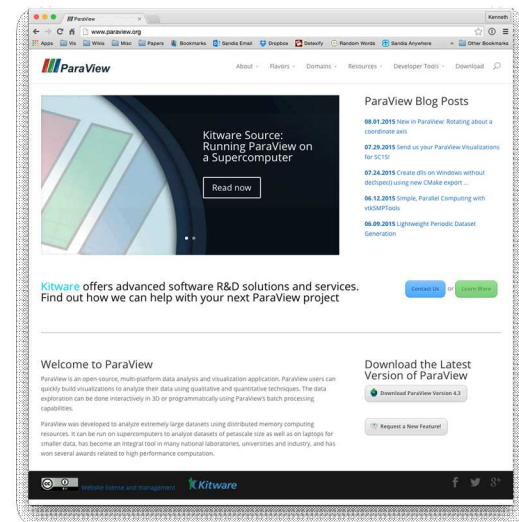
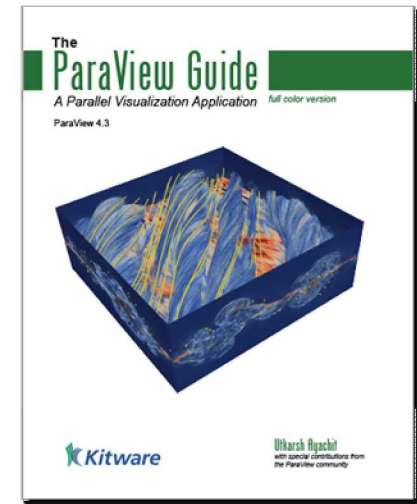
Hierarchical Adaptive
Mesh Refinement
(AMR)

Hierarchical Uniform
AMR

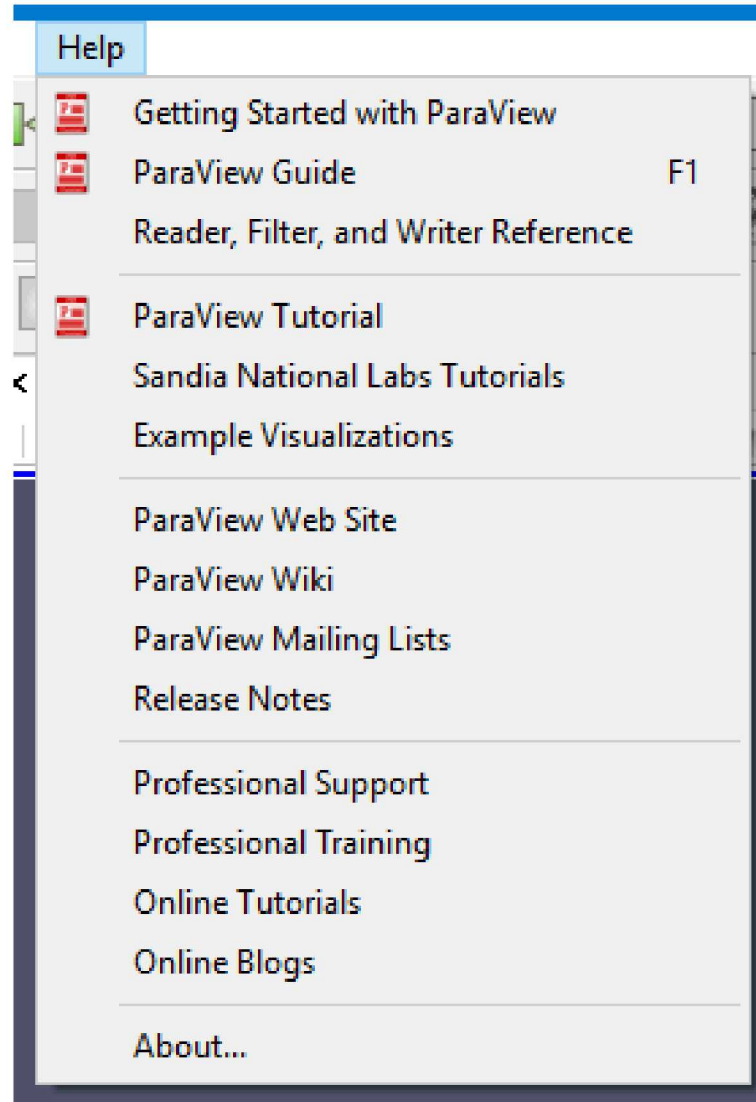
Octree

More Information

- Help Menu
- *The ParaView User's Guide*
 - <http://www.paraview.org/paraview-guide/>
- Tutorials
 - <http://www.paraview.org/tutorials/>
- The ParaView web page
 - www.paraview.org
- ParaView mailing list
 - paraview@paraview.org



Help Menu





Basic Usage

User Interface

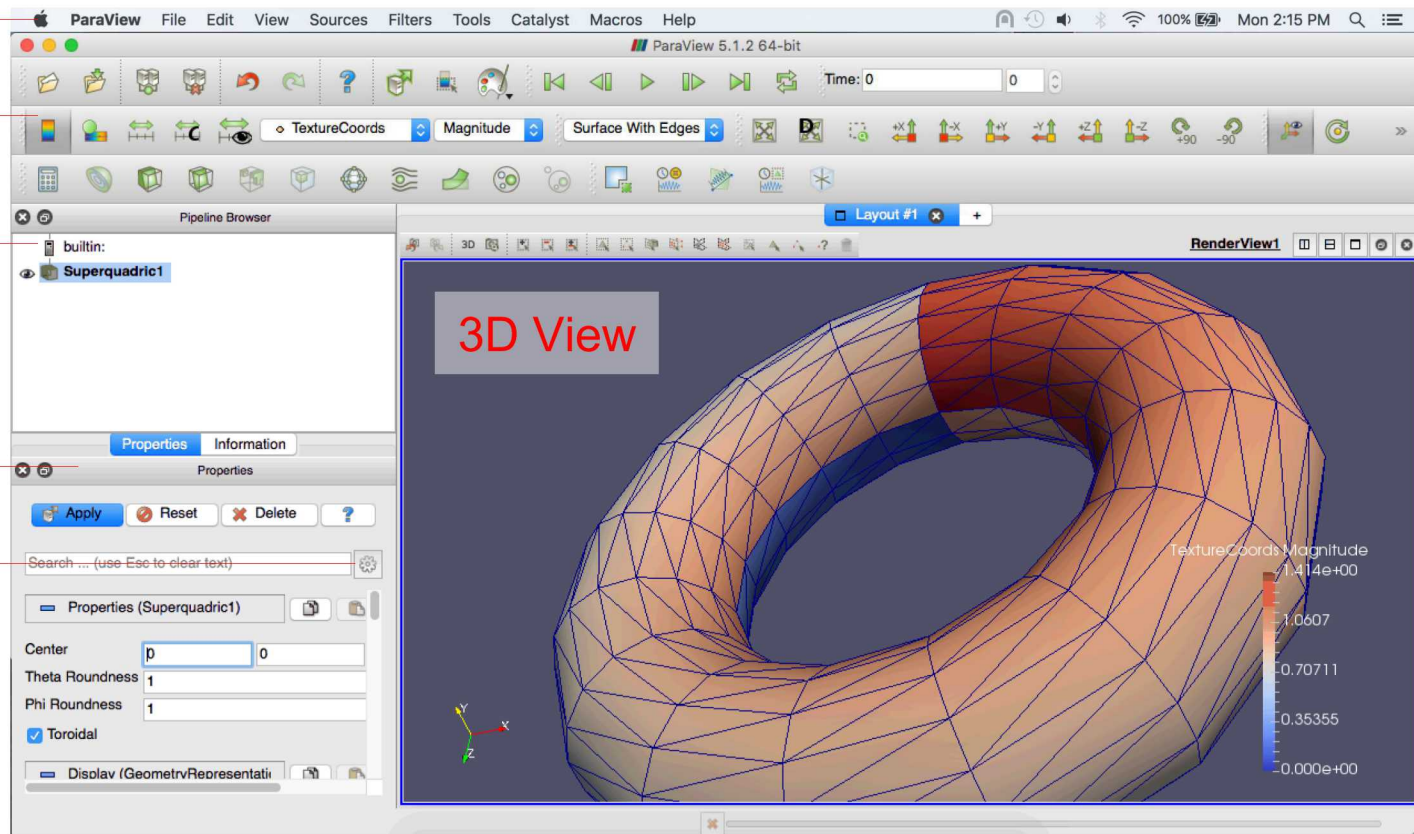
Menu Bar

Toolbars

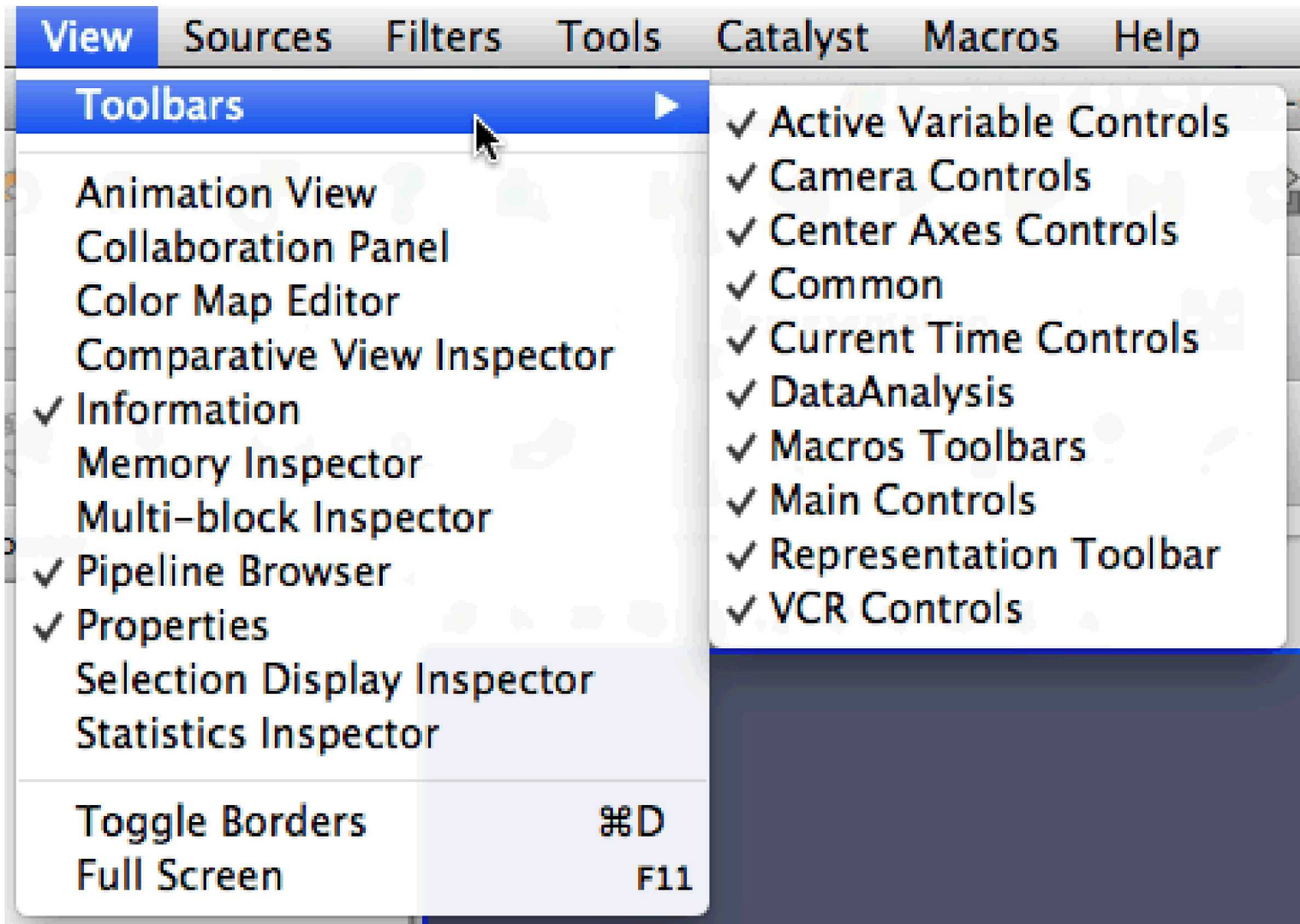
Pipeline Browser

Properties Panel

Advanced Toggle




Getting Back GUI Components



The screenshot shows the ParaView application's menu bar with the following items: View, Sources, Filters, Tools, Catalyst, Macros, and Help. The View menu is open, and the Toolbars submenu is also open, displaying a list of GUI components. A mouse cursor is positioned over the Toolbars menu item.

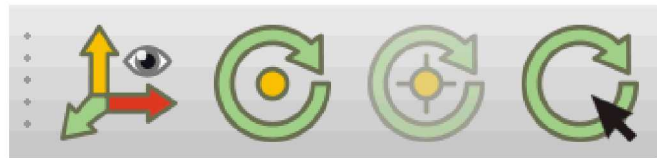
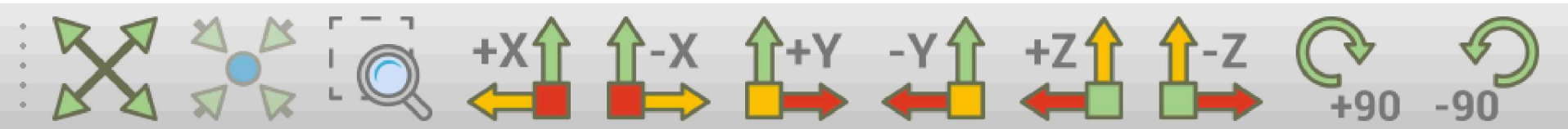
View Menu Item	Toolbars Submenu Item	Keyboard Shortcut
View	Active Variable Controls	
Sources	Camera Controls	
Filters	Center Axes Controls	
Tools	Common	
Catalyst	Current Time Controls	
Macros	DataAnalysis	
Help	Macros Toolbars	
	Main Controls	
	Representation Toolbar	
	VCR Controls	
	Animation View	
	Collaboration Panel	
	Color Map Editor	
	Comparative View Inspector	
	Information	
	Memory Inspector	
	Multi-block Inspector	
	Pipeline Browser	
	Properties	
	Selection Display Inspector	
	Statistics Inspector	
	Toggle Borders	⌘D
	Full Screen	F11

Creating a Cylinder Source


1. Go to the Sources menu, open the Geometric Shapes submenu, and select Cylinder.
2. Click the Apply button to accept the default parameters. 

Simple Camera Manipulation

- Drag left, middle, right buttons for rotate, pan, zoom.
 - Also use Shift, Ctrl, Alt modifiers.
 - Also try holding down x, y, or z.



Creating a Cylinder Source

1. Go to the Sources menu, open the Geometric Shapes submenu, and select Cylinder.
2. Click the Apply button to accept the default parameters. 
3. Increase the Resolution parameter.

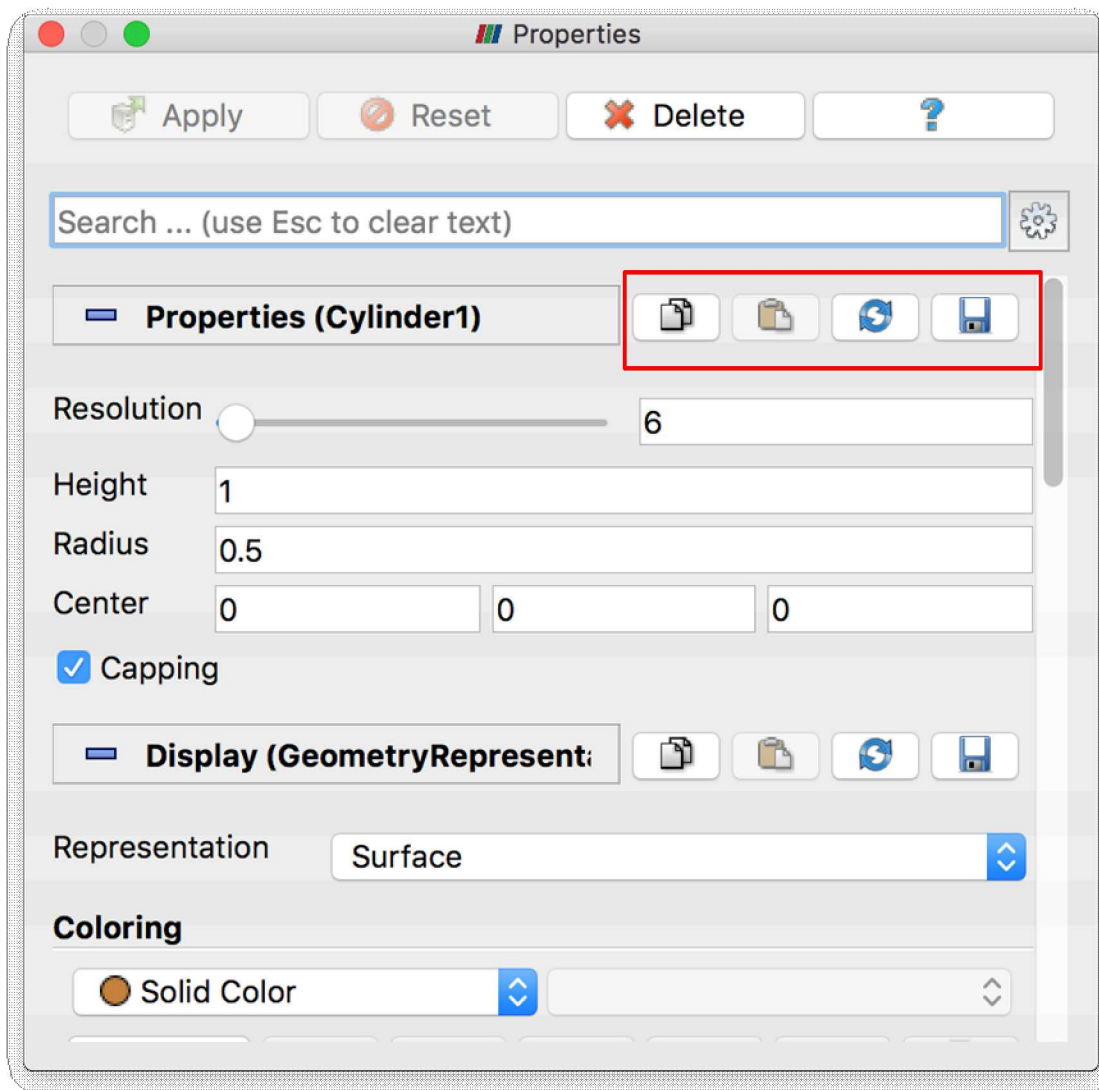


4. Click the  button again.

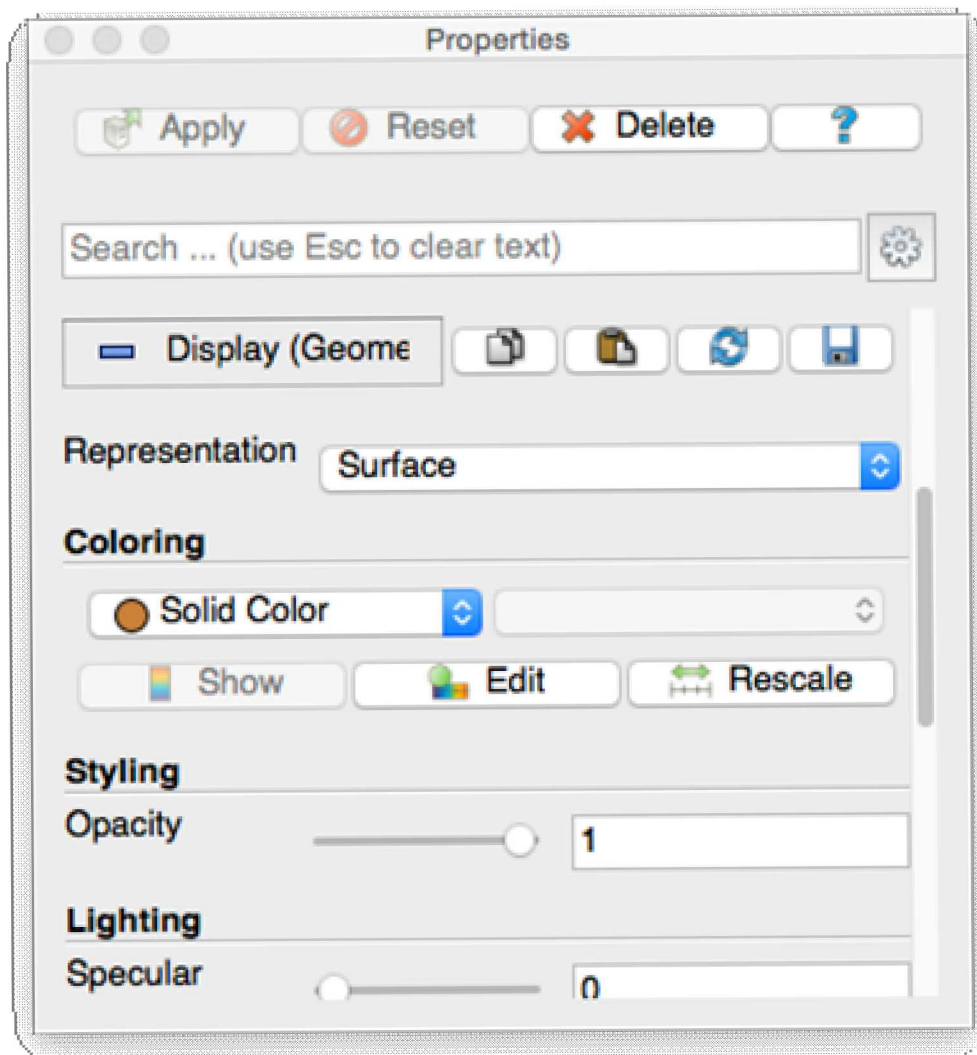
Pipeline Object Controls



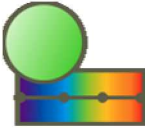
Copy/Paste/Reset/Save Parameters



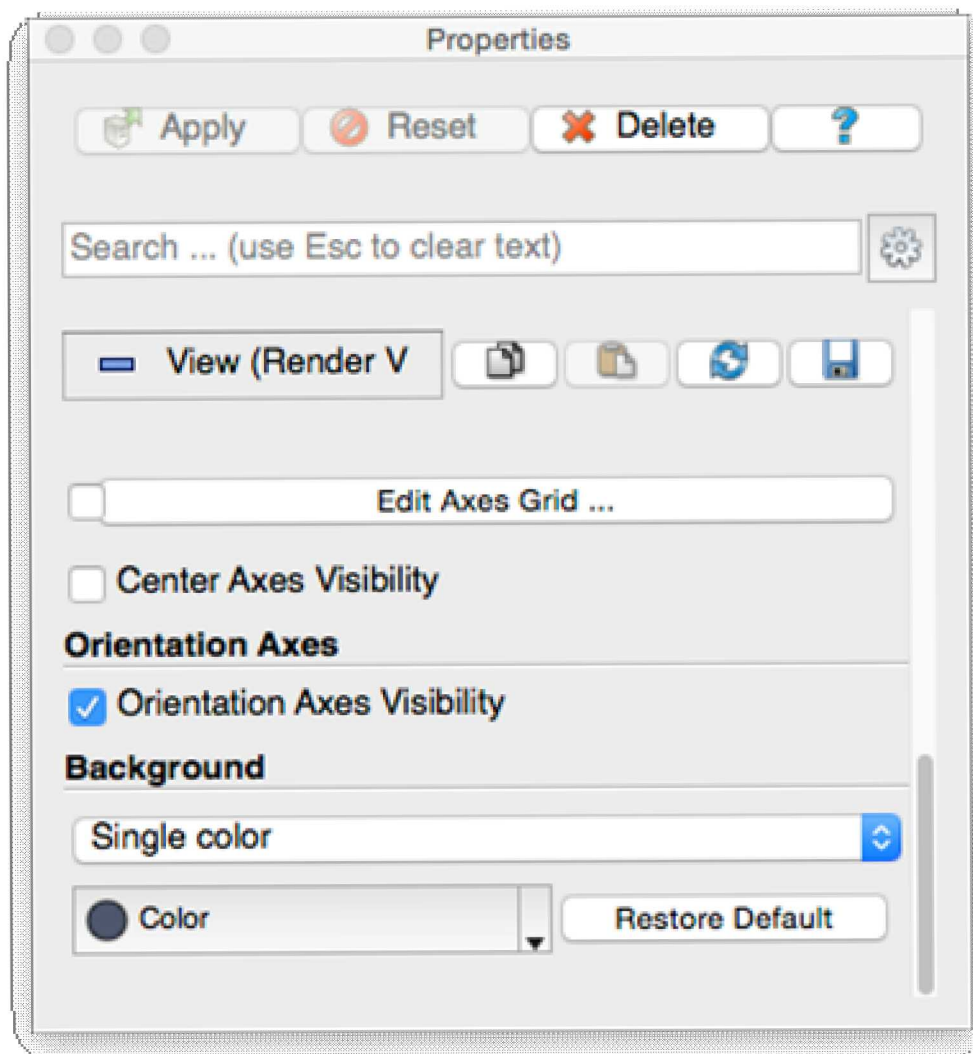
Display Properties



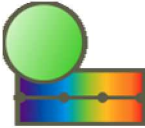
Change Render Properties

1. Scroll down to the Display group.
2. Click the  Edit button. (This button is replicated in the toolbar.)
3. Select a new color for the cylinder.

Render View Options

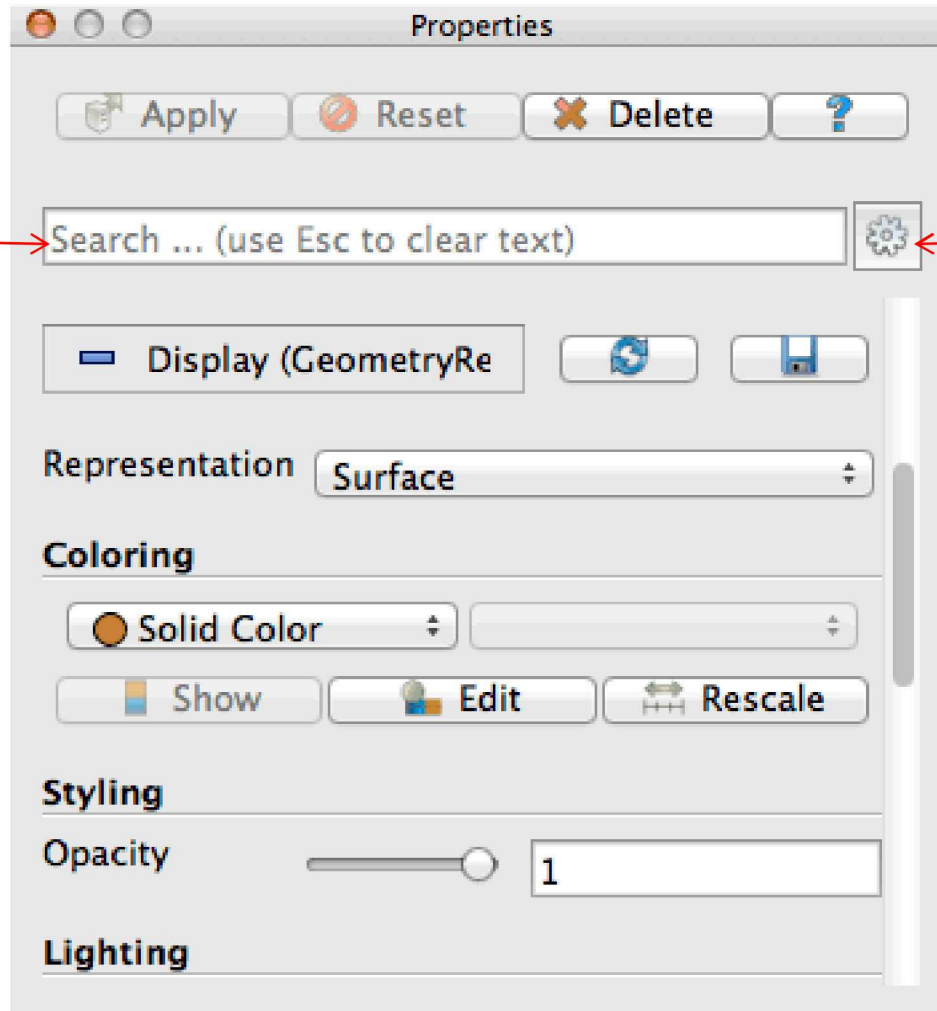


Change Render Properties

1. Scroll down to the Display group.
2. Click the  Edit button. (This button is replicated in the toolbar.)
3. Select a new color for the cylinder.
4. Scroll down to the View group.
5. Turn on the Axis Grid.

Advanced Properties

Search
Properties



Toggle
Advanced
Properties

Searching Properties

1. Type “specular” in the properties search box
2. Change Specular value to 1 (makes the cylinder shiny)



Searching Properties

1. Type “specular” in the properties search box
2. Change Specular value to 1 (makes the cylinder shiny)

Other interesting properties:

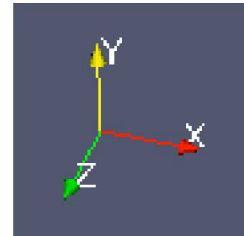
- Axes Grid
- Opacity

Using Auto Apply

1. Click Auto Apply. 
2. Change the Resolution parameter again. 
3. Note that the visualization automatically updates without having to hit Apply.

Changing the Color Palette

1. Make sure the orientation axes are visible in the lower left corner. 



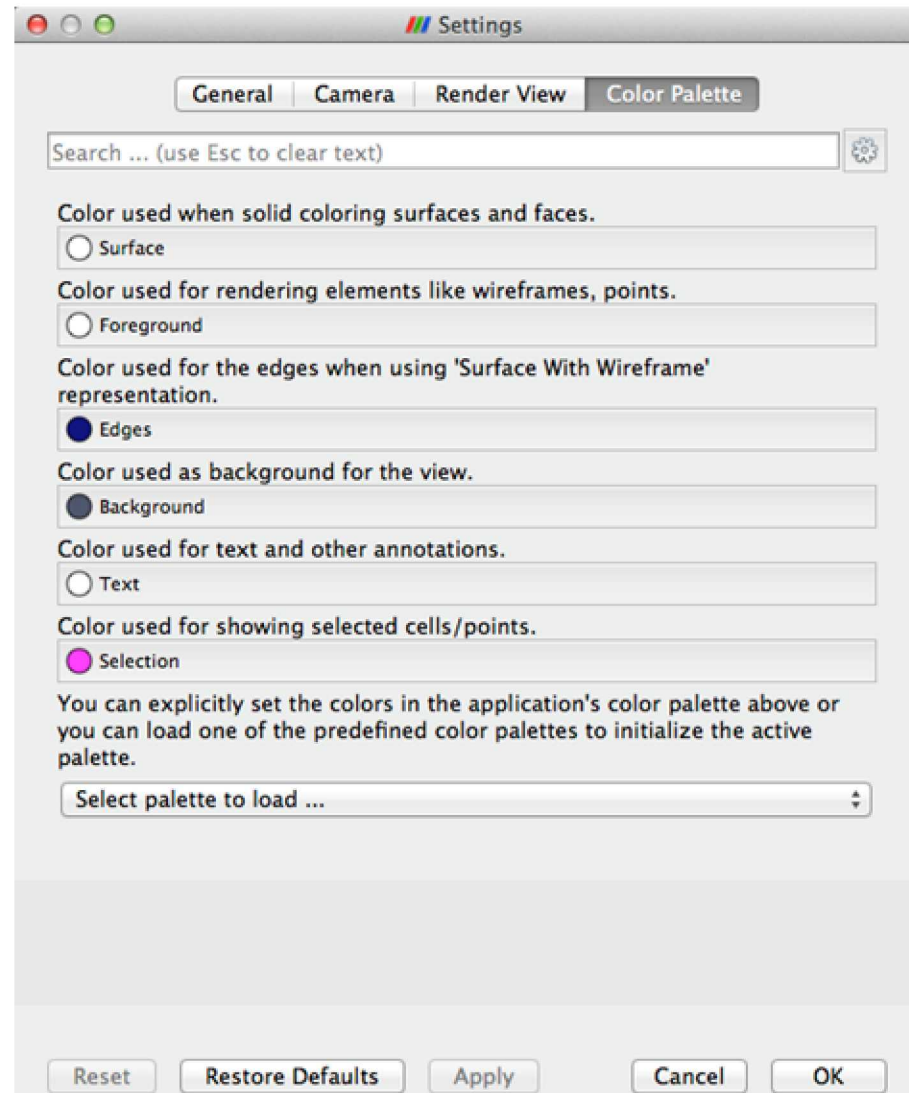
2. Click the color palette button  and change the colors.



3. Try several color palettes.

Color Palettes

 → Edit Current Palette...



Undo Redo



Undo



Redo




Camera
Undo





Camera
Redo

Creating a Cylinder Source

1. Go to the Sources menu, open the Geometric Shapes submenu, and select Cylinder.
2. Click the Apply button to accept the default parameters. 
3. Increase the Resolution parameter.



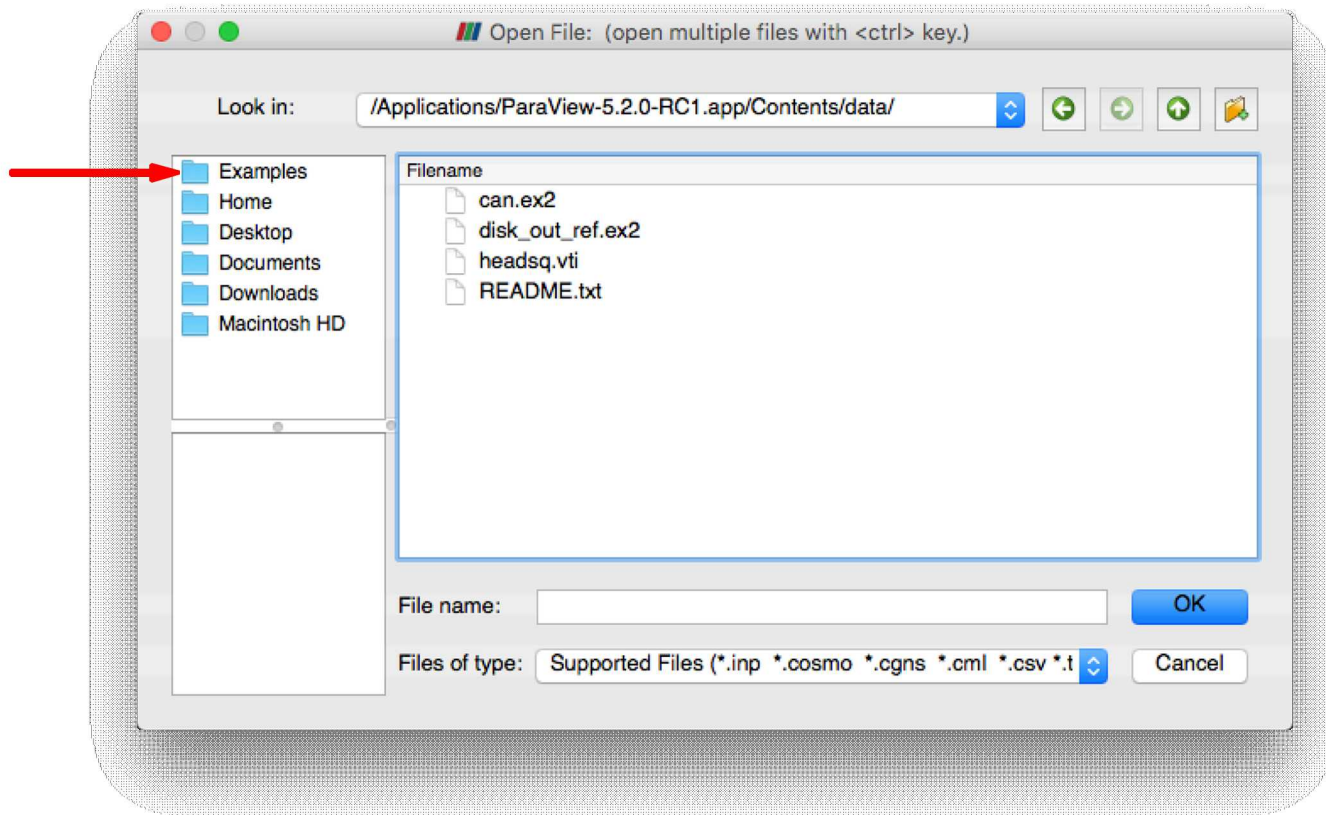
4. Click the  button again.
5. Delete the Cylinder. 

Supported Data Types

- ParaView Data (.pvd)
- VTK (.vtp, .vtu, .vti, .vts, .vtr)
- VTK Legacy (.vtk)
- VTK Multi Block (.vtm, .vtmb, .vtmg, .vthd, .vthb)
- Partitioned VTK (.pvtu, .pvti, .pvts, .pvtr)
- ADAPT (.nc, .cdf, .elev, .ncd)
- ANALYZE (.img, .hdr)
- ANSYS (.inp)
- AVS UCD (.inp)
- BOV (.bov)
- BYU (.g)
- CAM NetCDF (.nc, .ncdf)
- CCSM MTSD (.nc, .cdf, .elev, .ncd)
- CCSM STSD (.nc, .cdf, .elev, .ncd)
- CEAUcd (.ucd, .inp)
- CGNS (.cgns)
- CMAT (.cmat)
- CML (.cml)
- CTRL (.ctrl)
- Chombo (.hdf5, .h5)
- Claw (.claw)
- Comma Separated Values (.csv)
- Cosmology Files (.cosmo, .gadget2)
- Curve2D (.curve, .ultra, .ult, .u)
- DDCMD (.ddcmd)
- Digital Elevation Map (.dem)
- Dyna3D(.dyn)
- EnSight (.case, .sos)
- Enzo boundary and hierarchy
- ExodusII (.g, .e, .exe, .ex2, .ex2v..., etc)
- ExtrudedVol (.exvol)
- FVCOM (MTMD, MTSD, Particle, STSD)
- Facet Polygonal Data
- Flash multiblock files
- Fluent Case Files (.cas)
- GGCM (.3df, .mer)
- GTC (.h5)
- GULP (.trg)
- Gadget (.gadget)
- Gaussian Cube File (.cube)
- JPEG Image (.jpg, .jpeg)
- LAMPPS Dump (.dump)
- LAMPPS Structure Files
- LODI (.nc, .cdf, .elev, .ncd)
- LODI Particle (.nc, .cdf, .elev, .ncd)
- LS-DYNA (.k, .lsdyna, .d3plot, d3plot)
- M3DCI (.h5)
- MFIX Unstructured Grid (.RES)
- MM5 (.mm5)
- MPAS NetCDF (.nc, .ncdf)
- Meta Image (.mhd, .mha)
- Miranda (.mir, .raw)
- Multilevel 3d Plasma (.m3d, .h5)
- NASTRAN (.nas, .f06)
- Nek5000 Files
- Nrrd Raw Image (.nrrd, .nhdr)
- OpenFOAM Files (.foam)
- PATRAN (.neu)
- PFLOTTRAN (.h5)
- PLOT2D (.p2d)
- PLOT3D (.xyz, .q, .x, .vp3d)
- PLY Polygonal File Format
- PNG Image Files
- POP Ocean Files
- ParaDIS Files
- Phasta Files (.pht)
- Pixie Files (.h5)
- ProSTAR (.cel, .vrt)
- Protein Data Bank (.pdb, .ent, .pdb)
- Raw Image Files
- Raw NRRD image files (.nrrd)
- SAMRAI (.samrai)
- SAR (.SAR, .sar)
- SAS (.sasgeom, .sas, .sasdata)
- SESAME Tables
- SLAC netCDF mesh and mode data
- SLAC netCDF particle data
- Silo (.silo, .pdb)
- Spherical (.spherical, .sv)
- SpyPlot CTH
- SpyPlot (.case)
- SpyPlot History (.hscth)
- Stereo Lithography (.stl)
- TFT Files
- TIFF Image Files
- Tsurf Files
- Tecplot ASCII (.tec, .tp)
- Tecplot Binary (.plt)
- Tetrad (.hdf5, .h5)
- UNIC (.h5)
- VASP CHGCA (.CHG)
- VASP OUT (.OUT)
- VASP POSTCAR (.POS)
- VPIC (.vpc)
- VRML (.wrl)
- Velodyne (.vld, .rst)
- VizSchema (.h5, .vsh5)
- Wavefront Polygonal Data (.obj)
- WindBlade (.wind)
- XDMF and hdf5 (.xmf, .xdmf)
- XMol Molecule

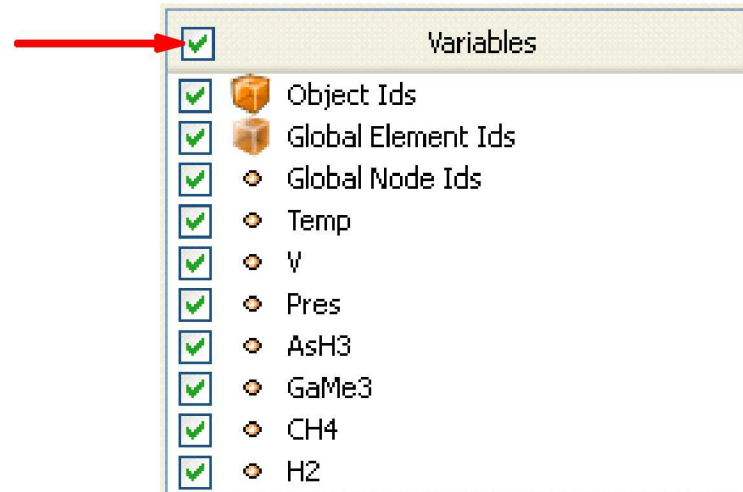
Load disk_out_ref.ex2

1. Open the file `disk_out_ref.ex2` from the Examples directory.



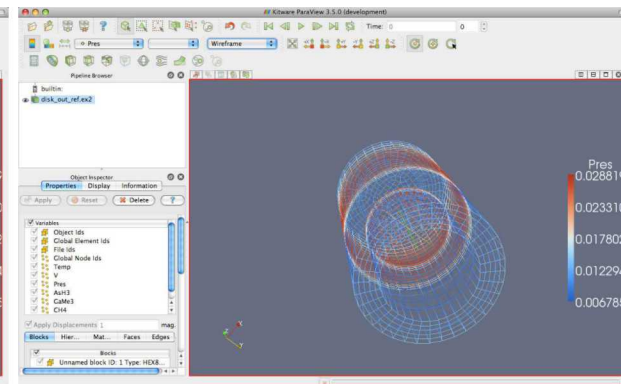
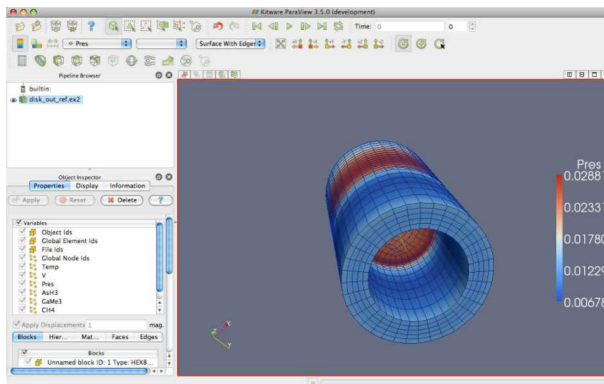
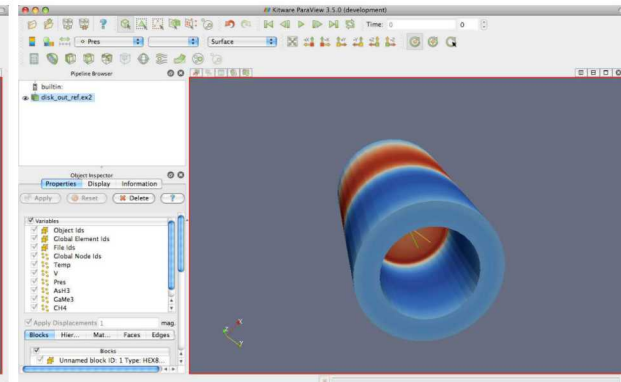
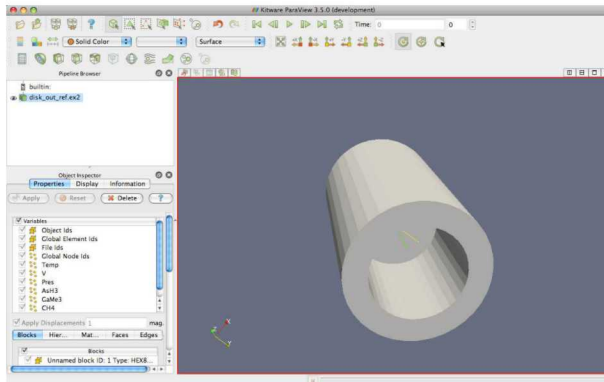
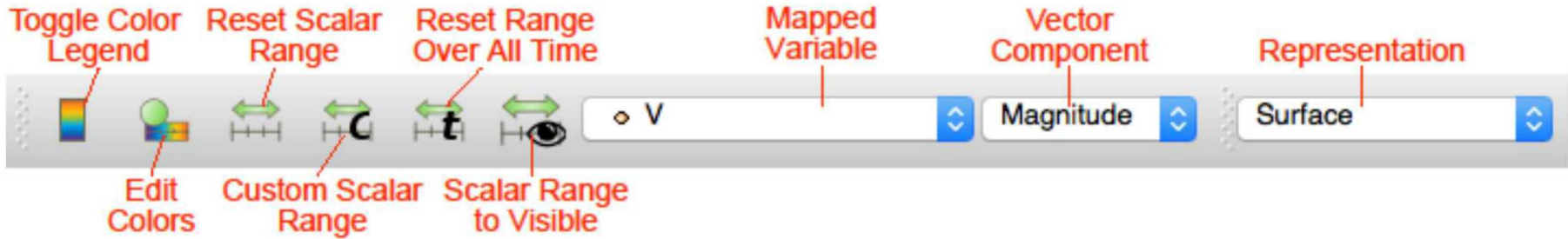
Load disk_out_ref.ex2

1. Open the file disk_out_ref.ex2 from the Examples directory.
2. Load all data variables.



3. Click 

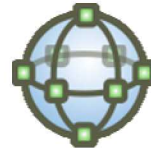
Data Representation



Common Filters



Calculator



Glyph



Contour



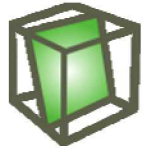
Stream Tracer



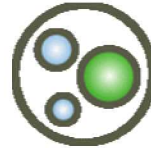
Clip



Warp (vector)



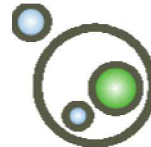
Slice



Group Datasets



Threshold

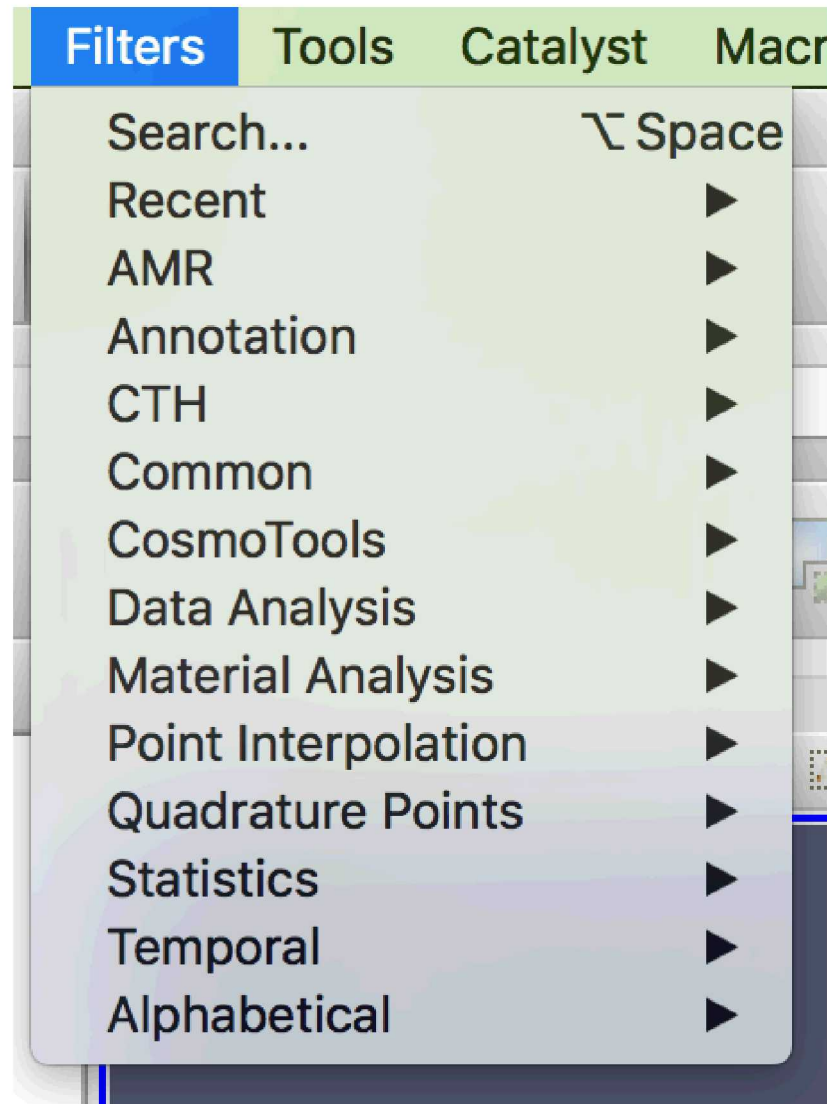


Extract Level

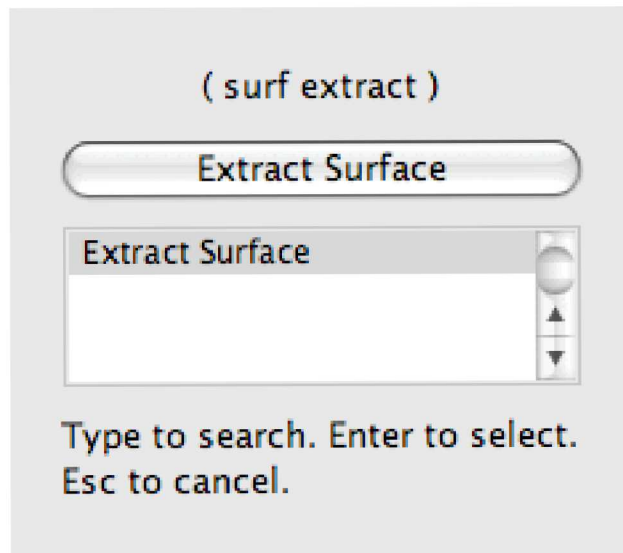


Extract Subset

Filters Menu




Quick Launch



- Used for searching for filters by name
- Keyboard shortcut
 - Ctrl-space for Windows & Linux
 - Alt-space for Mac

Apply a Filter

1. Make sure that disk_out_ref.ex2 is selected in the pipeline browser.
2. Select the contour filter. 

Apply a Filter

3. Change parameters to create an isosurface at Temp = 400K.

Change to Temp

Change to 400

Properties

Apply Reset Delete ?

Search...

Properties (Contour1)

Contour

Contour By: AsH3

Compute Normals

Compute Gradients

Compute Scalars

Isosurfaces

Value Range: [0.0804768, 0.184839]

0.132658146

Delete



Delete All

New Value


New Range

Scientific





Apply a Filter

1. Make sure that `disk_out_ref.ex2` is selected in the pipeline browser.
2. Select the contour filter. 
3. Change parameters to create an isosurface at $\text{Temp} = 400\text{K}$.
4. 

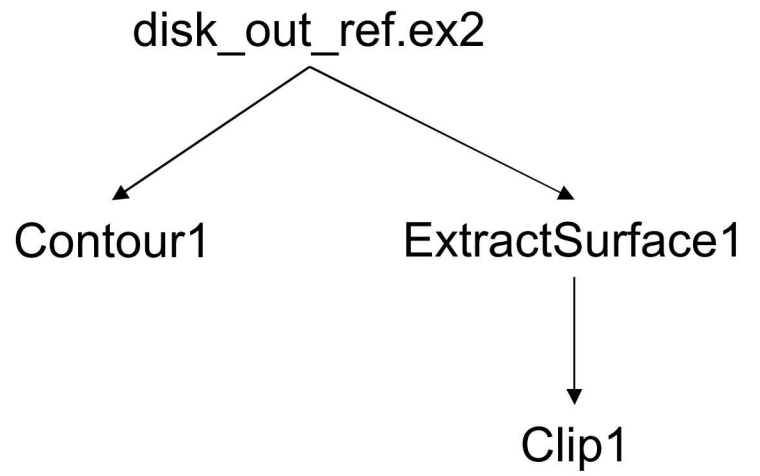
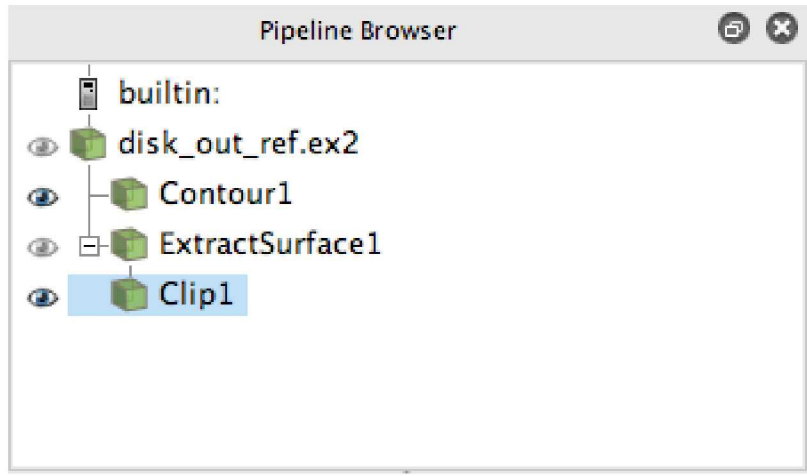
Create a Cutaway Surface

1. Select `disk_out_ref.ex2` in the pipeline browser.
2. From the quick launch, select Extract Surface.
3.  Apply

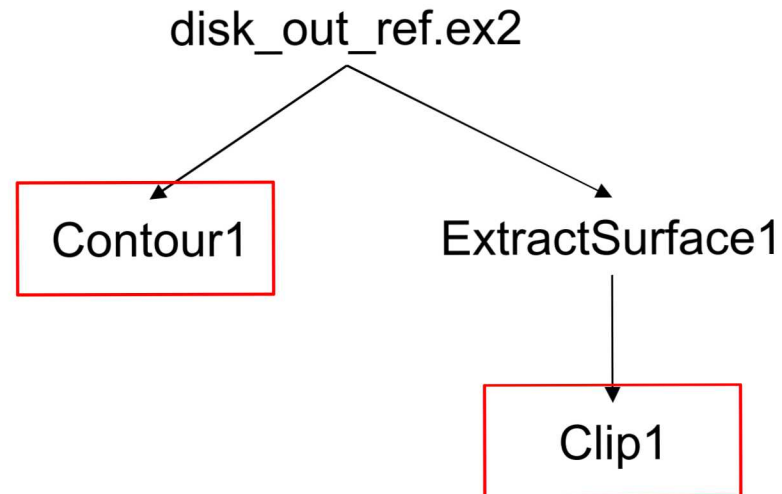
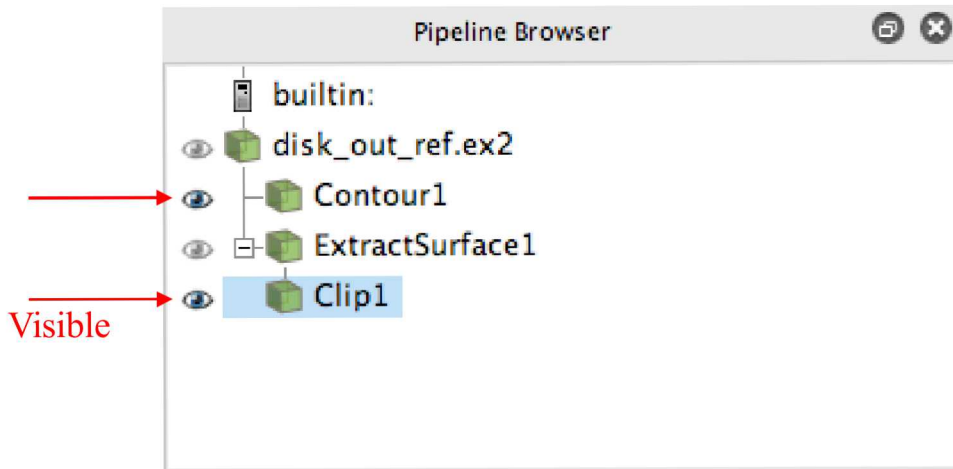
Create a Cutaway Surface

1. Select `disk_out_ref.ex2` in the pipeline browser.
2. From the quick launch, select Extract Surface.
3.  Apply
4. Create a clip filter. 
5. Uncheck Show Plane.  Show Plane
6.  Apply

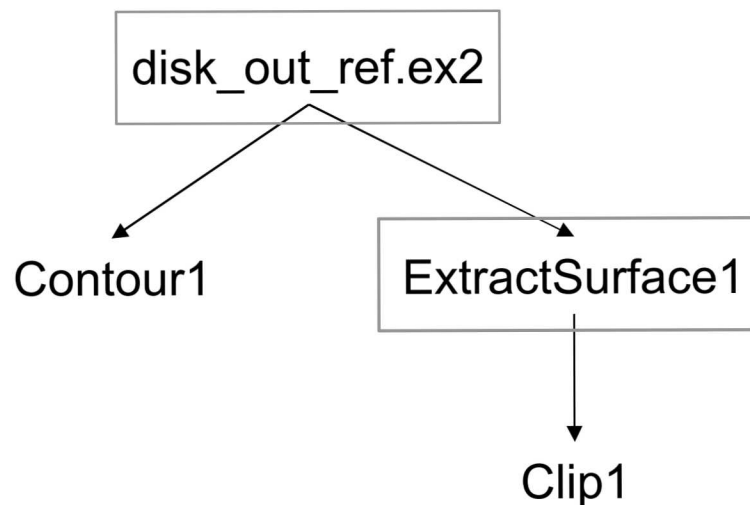
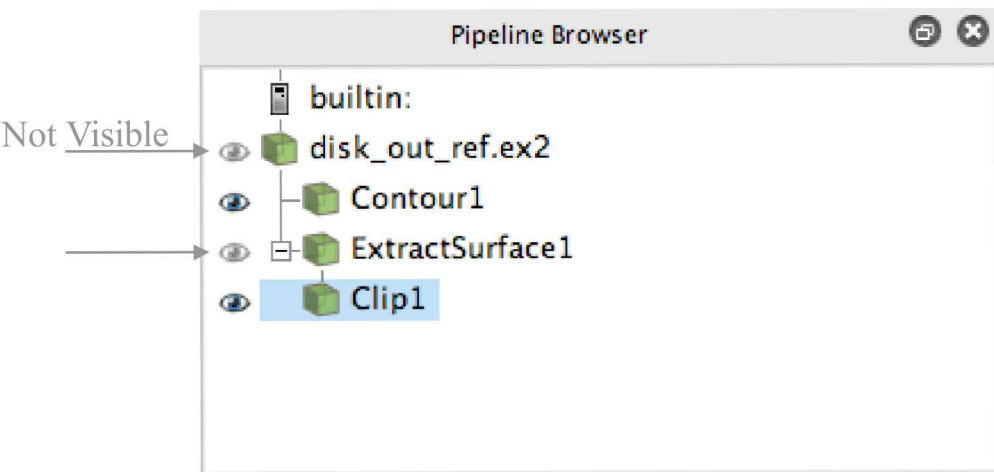
Pipeline Browser Structure



Pipeline Browser Structure



Pipeline Browser Structure

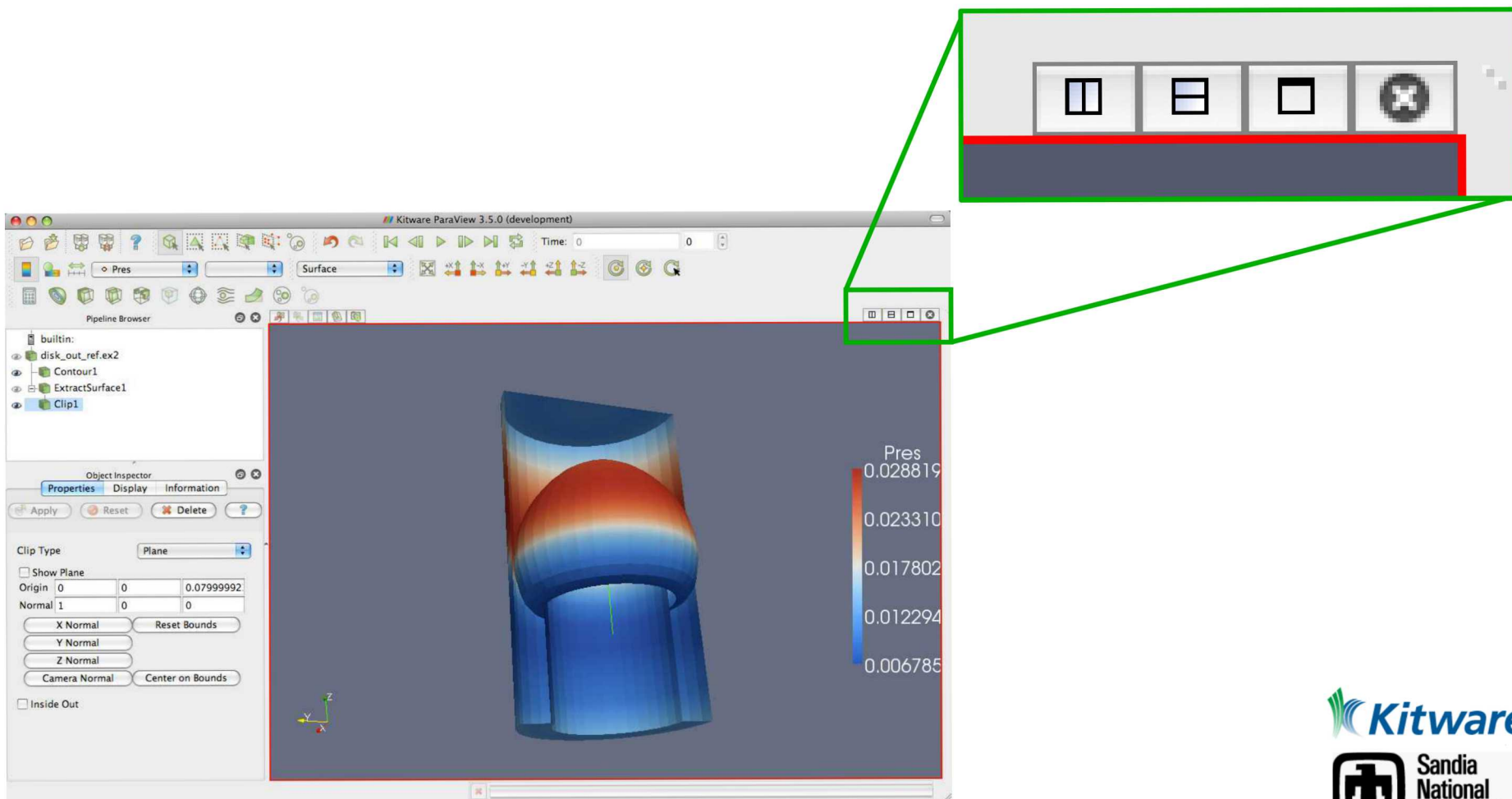


Reset ParaView





Edit → Reset Session





Multiview





Multiview

1. Open disk_out_ref.ex2. Load all variables. 
2. Add Clip filter. 
3. Uncheck Show Plane. 
4. 
5. Color surface by Pres.



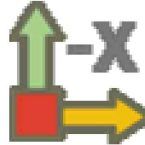
Multiview

6. Split the view horizontally. 
7. Make Clip1 visible. 
8. Color surface by Temp.

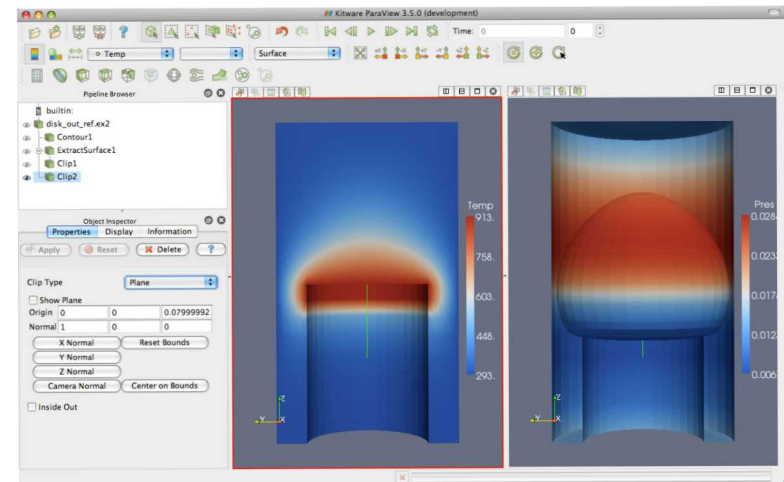
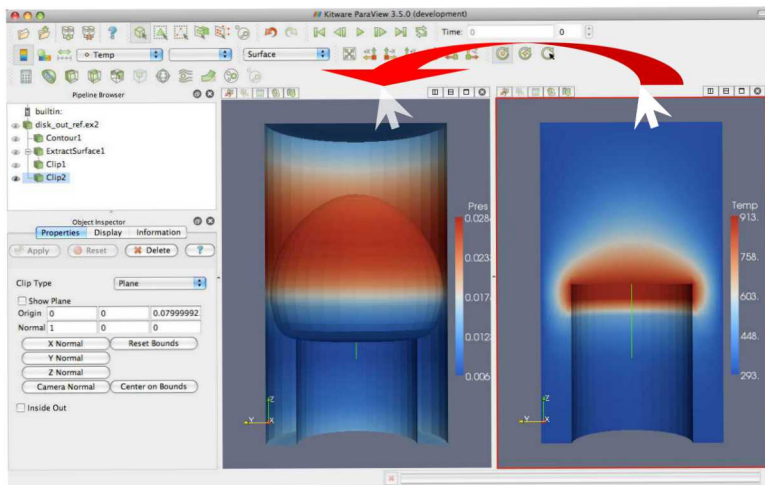
Multiview

6. Split the view horizontally. 
7. Make Clip1 visible. 
8. Color surface by Temp.
9. Right-click view, Link Camera...
10. Click other view.

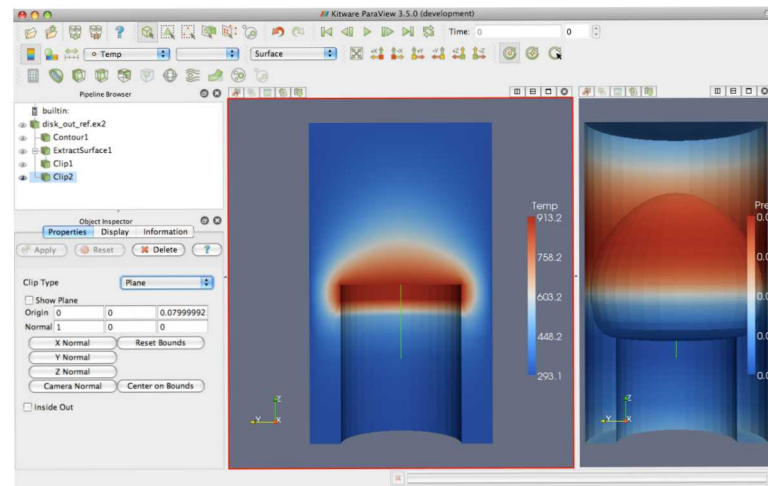
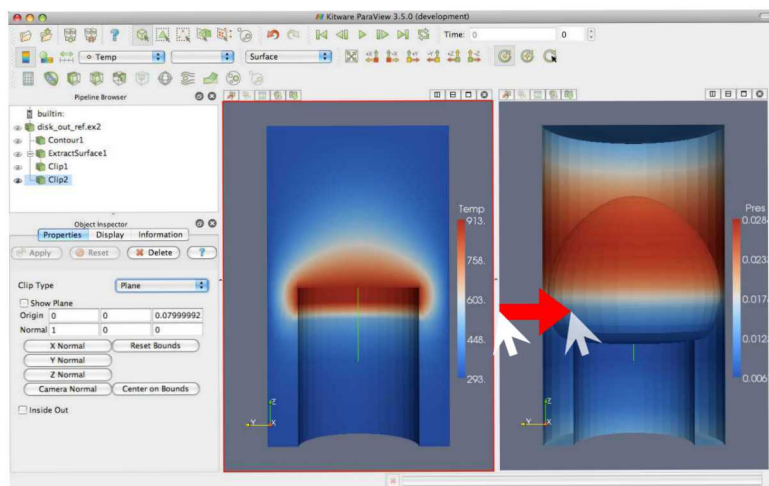
Multiview

6. Split the view horizontally. 
7. Make Clip1 visible. 
8. Color surface by Temp.
9. Right-click view, Link Camera...
10. Click other view.
11. Click  and zoom in a bit.

Modifying Views



Modifying Views






Reset ParaView





Edit → Reset Session



Streamlines




1. Open disk_out_ref.ex2. Load all variables. 
2. Add Stream Tracer. 
3. Change Seed Type to Point Source.
4. Uncheck Show Sphere. Show Sphere
5. 

Streamlines

1. Open disk_out_ref.ex2. Load all variables. 
2. Add Stream Tracer. 
3. Change Seed Type to Point Source.
4. Uncheck Show Sphere. Show Sphere
5. 
6. From the quick launch, select Tube*
7. 

*Alternatively you can just render the lines as tubes.

Getting Fancy

8. Select StreamTracer1.
9. Add Glyph filter. 
10. Change Glyph Type to Cone.
11. Change Scale Array to V.
12. Click reset  in Glyph Data Range.
13. Change Orientation Array to V.
14. 
15. Color by Temp.

Getting Answers

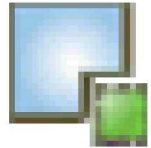
- Where is the air moving the fastest?
Near the disk or away from it? At the center of the disk or near its edges?
- Which way is the plate spinning?
- At the surface of the disk, is air moving toward the center or away from it?

Reset ParaView

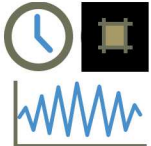
Edit → Reset Session



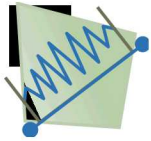
Common Data Analysis Filters



Extract Selection



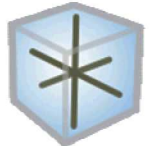
Plot Global Variables Over Time



Plot Over Line




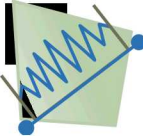


Plot Selection Over Time



Probe Location

Plotting

1. Open disk_out_ref.ex2. Load all variables. 
2. Clip,  uncheck, Show Plane , 
3. Select disk_out_ref.ex2.
4. Add Plot Over Line filter. 

3D Widgets

The screenshot displays the ParaView 3.14.1-1643-g2537cab 64-bit interface. The main window shows a 3D visualization of a cylinder with a heatmap overlay, indicating a temperature or pressure distribution. A white line is drawn across the top of the cylinder, and a green vertical line is drawn through the center of the cylinder's top surface. The interface includes a Pipeline Browser on the left, a Properties panel, and a toolbar at the top.

Pipeline Browser:

- builtin:
 - disk_out_ref.ex2
 - Clip1
 - PlotOverLine1

Properties (PlotOverLine1):




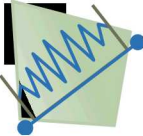

- Probe Type: High Resolution Line Source
- Show Line:
- Point1: X: -5.75, Y: -5.75, Z: -10
- Point2: X: 5.75, Y: 5.75, Z: 10.159999
- Resolution: 100

Note: Move mouse and use 'P' key to


Placing 3D Line Widget Endpoints

- Use the p key to place alternating points.
 - Ctrl+p places at nearest mesh point.
- Use the 1 or 2 key to place the start or end point.
 - Ctrl+1 or Ctrl+2 places at mesh point.
- Drag the endpoints.
 - Use x, y, or z key to constrain to axis.
- Use widgets in Properties panel
 - E.g. Use Z Axis button and then edit points to place from (0,0,0) to (0, 0, 10).

Plotting

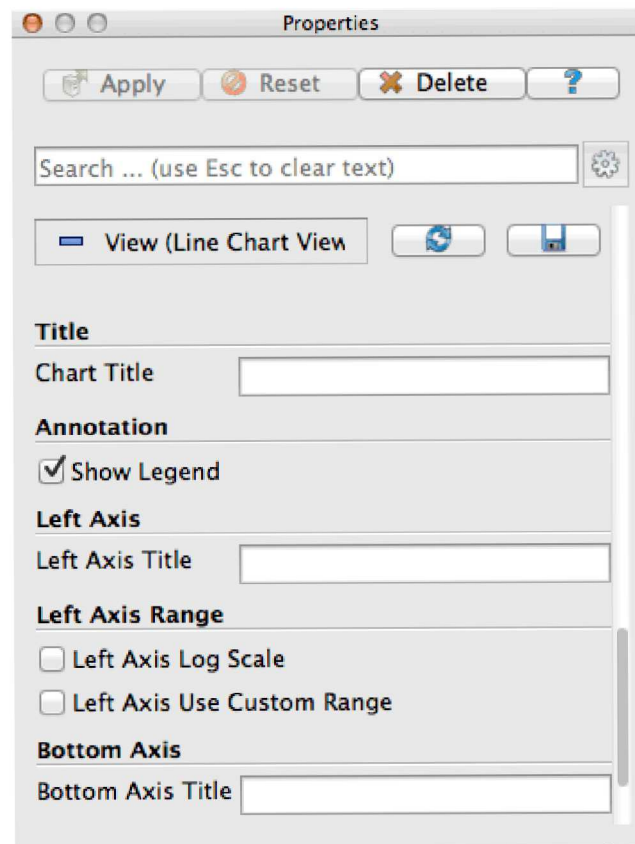
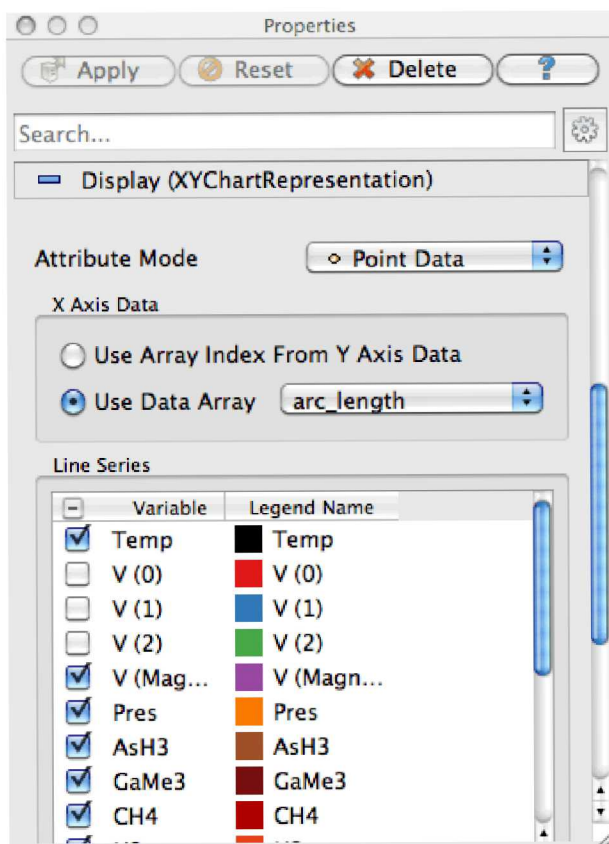
1. Open disk_out_ref.ex2. Load all variables. 
2. Clip,  uncheck, Show Plane , 
3. Select disk_out_ref.ex2.
4. Add Plot Over Line filter. 
5. Once line is satisfactorily located, 

Interacting with Plots

- Left, middle, right buttons to pan, zoom.
- Mouse wheel to zoom.
- Reset view to plot ranges. 

Plots are Views


- Move them like Views.
- Save screenshots.

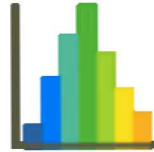


Adjusting Plots

1. In Display section of properties panel, turn off all variables except Temp and Pres.
2. Select Pres in the Display options.
3. Change Chart Axis to Bottom – Right.
4. Verify the relationship between temperature and pressure.

Histogram / Bar Chart

1. Select disk_out_ref.ex2.
2. Filters → Data Analysis → Histogram
3. Change *Temperature* to Temp.
4. 

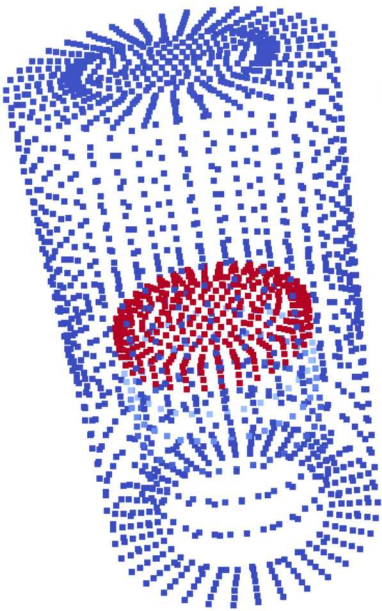


Reset ParaView

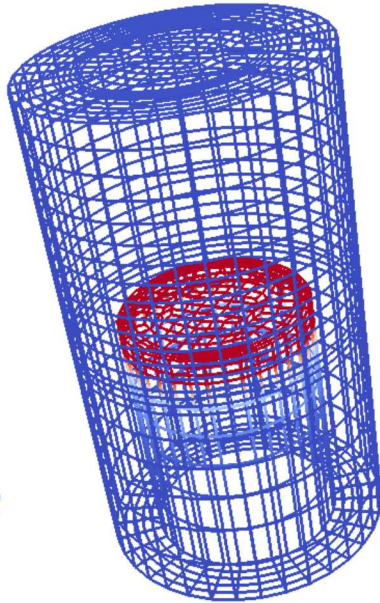
Edit → Reset Session



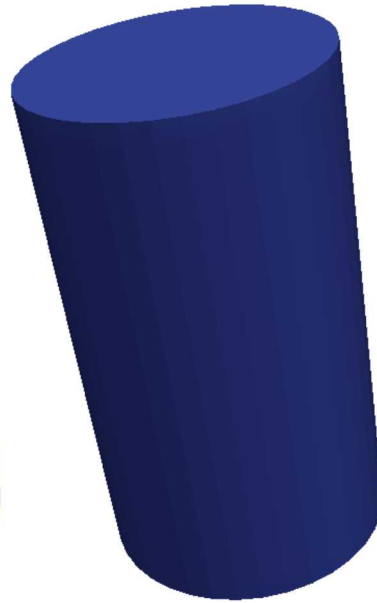
Geometry Representations



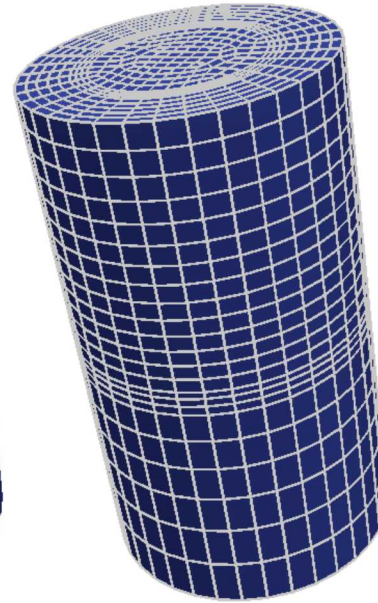
Points



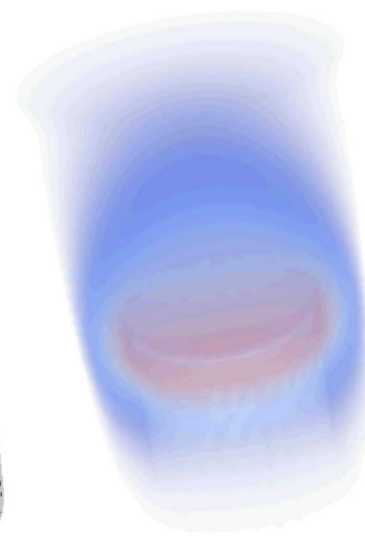
Wireframe



Surface




Surface
with Edges






Volume

Volume Rendering

1. Open disk_out_ref.ex2. Load all variables. 
2. Change variable viewed to Temp.
3. Change representation to Volume.
4. In the Are you Sure dialog box, click Yes.

Volume Rendering + Surface Geometry



1. Open disk_out_ref.ex2. Load all variables. 
2. Change variable viewed to Temp.
3. Change representation to Volume.
4. Add Stream Tracer.  
5. Optional: Add Tubes and Glyphs.

Transfer Function Editor



Color Map Editor

Search...

Interpret Values As Categories

Mapping Data

Data:

Use log scale when mapping data to colors

Enable opacity mapping for surfaces

Automatically rescale transfer functions to fit data

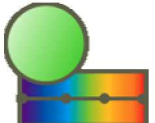

Color Mapping Parameters

Color Space

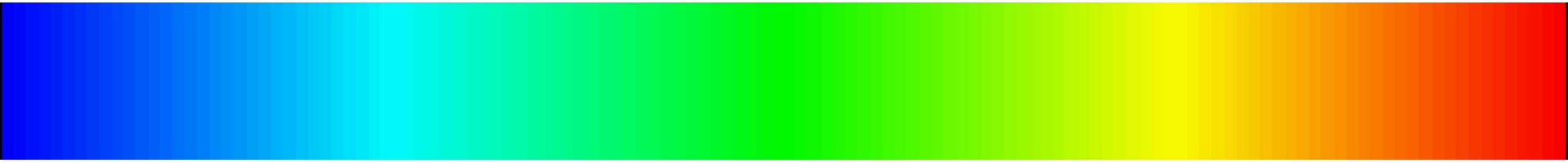
Nan Color

Save as default Update

Modify Transfer Function

1. Select disk_out_ref.ex2.
2. Click Edit Color Map .
3. Click Choose preset .
4. Select Black-Body Radiation.
Apply. Close.
5. Try adding and changing control points.

Rainbow Colors: Never Use Them

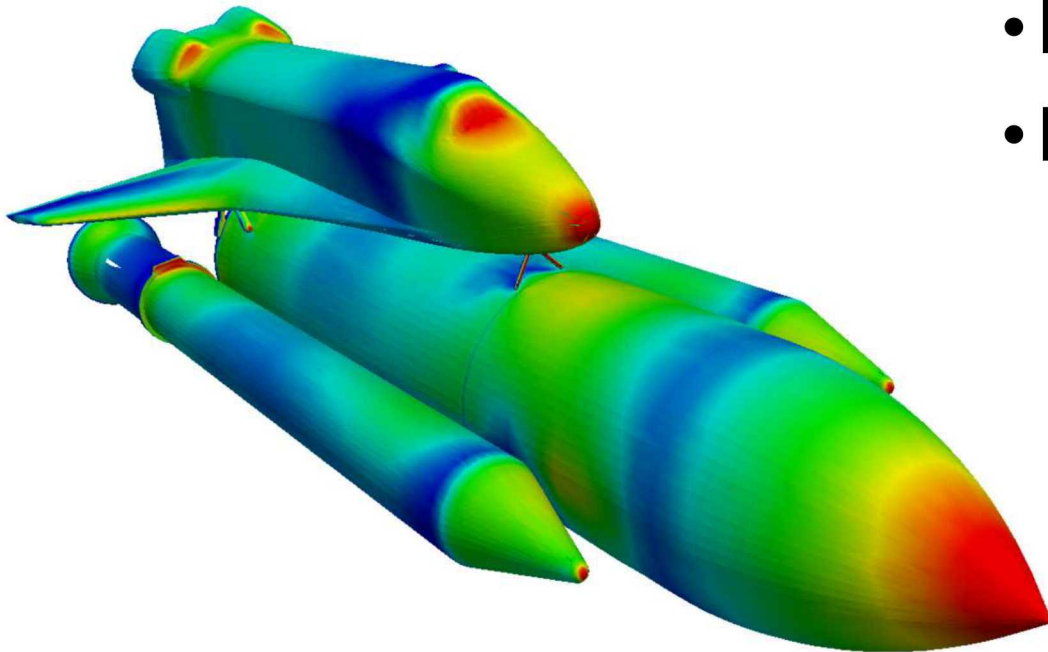


Pros:

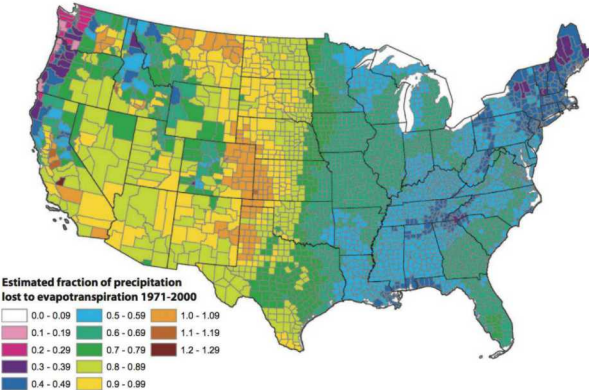
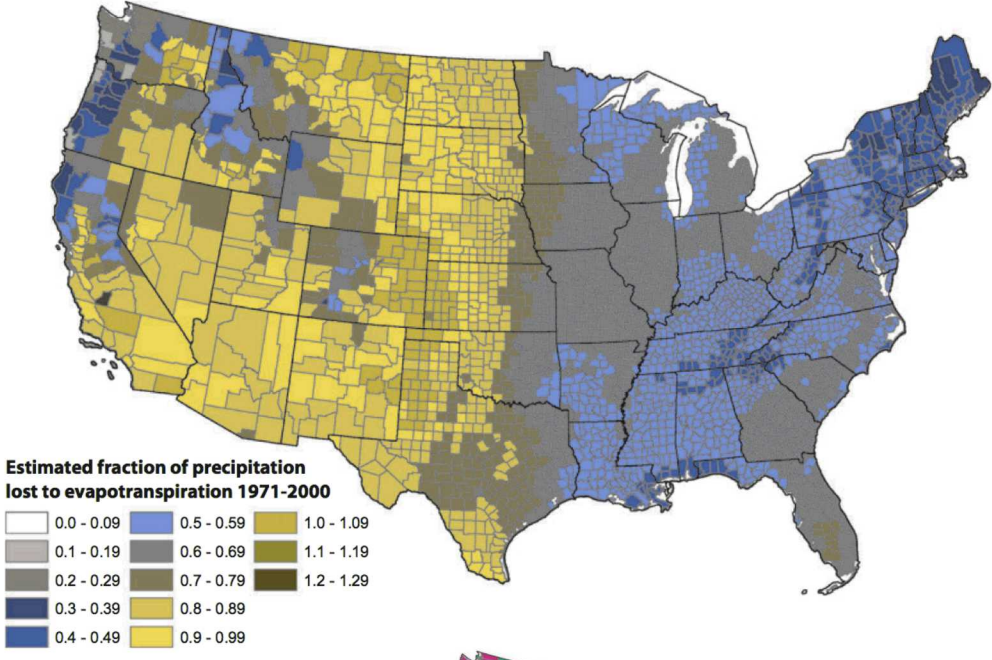
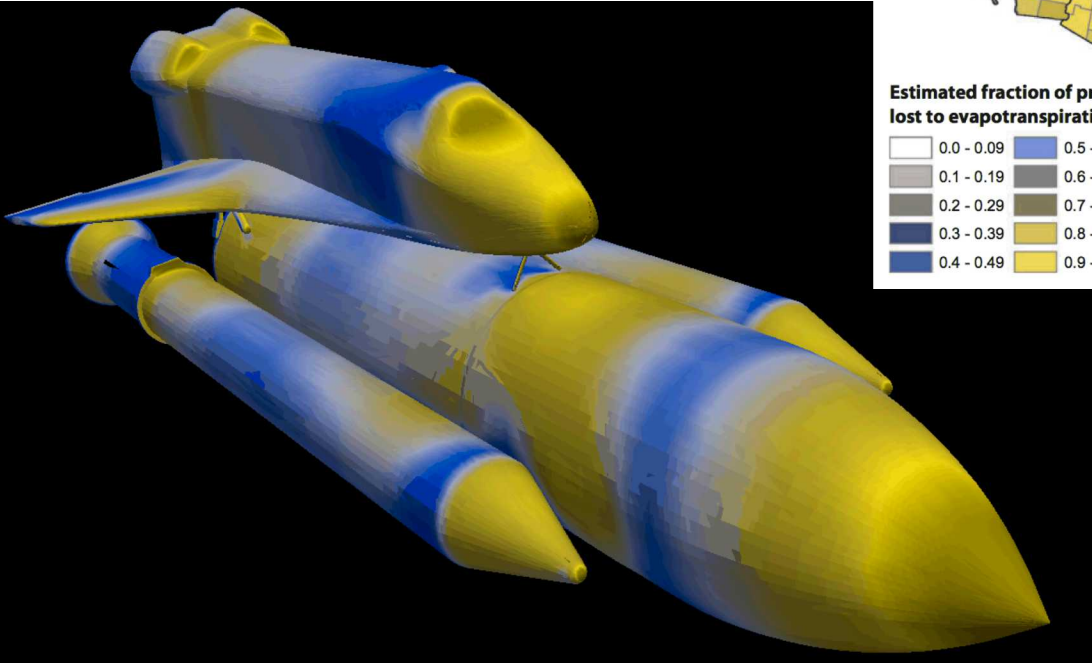
- Pretty
- Good dynamic range

Cons:

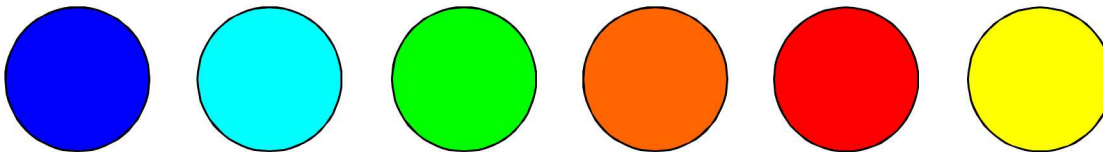
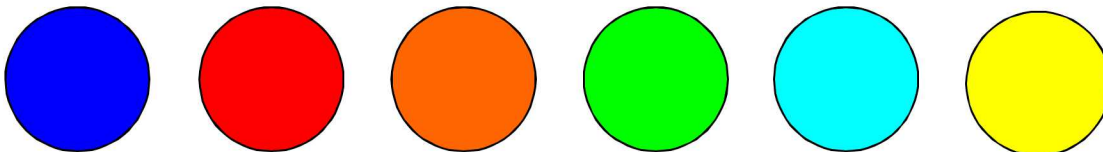
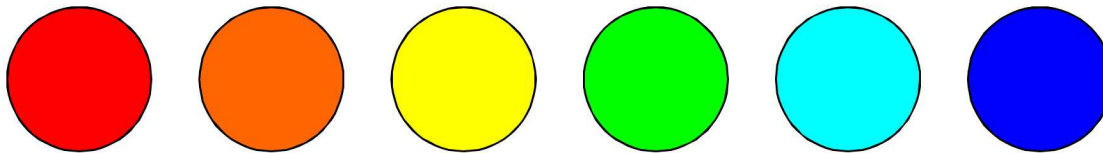
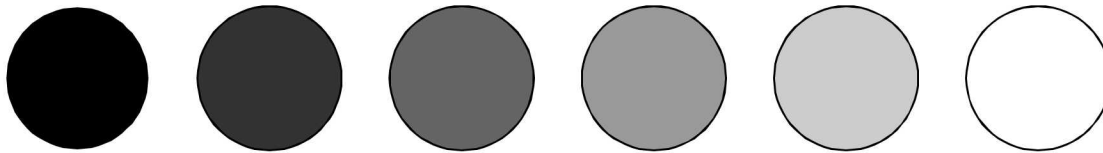
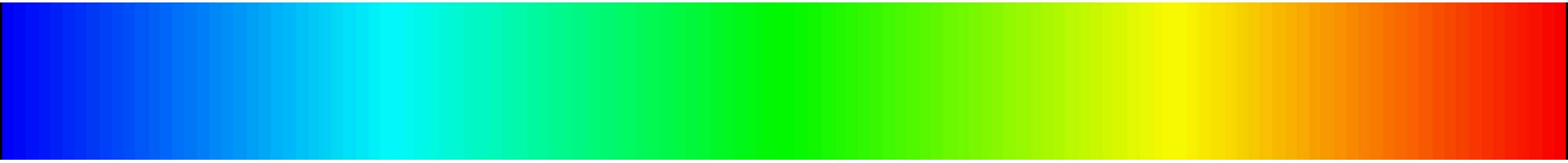
- Garish
- Bad for color blindness
- No implicit order
- Not perceptually even



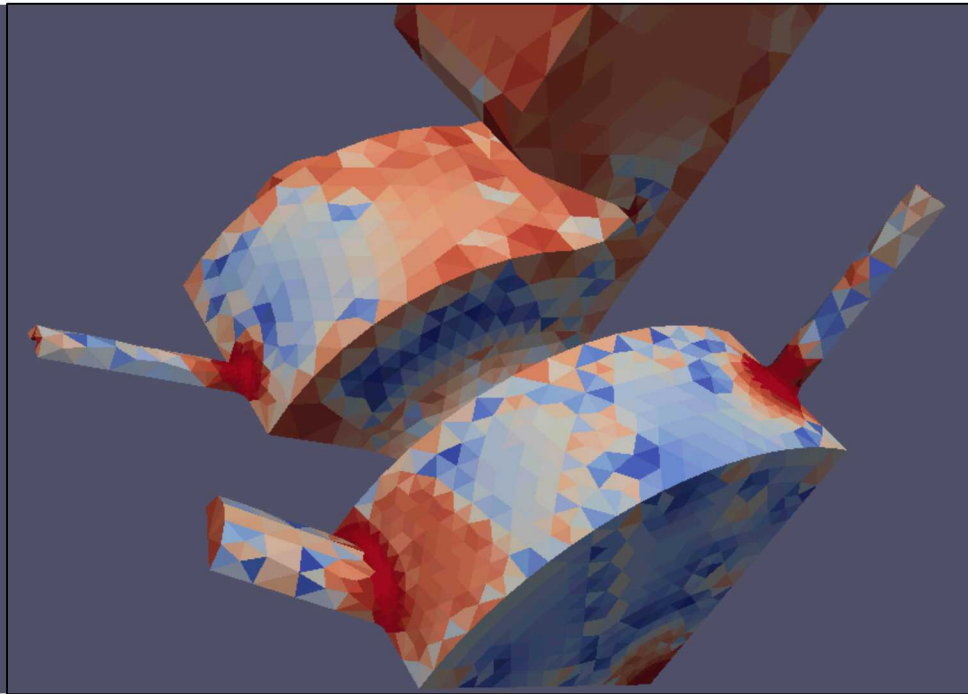
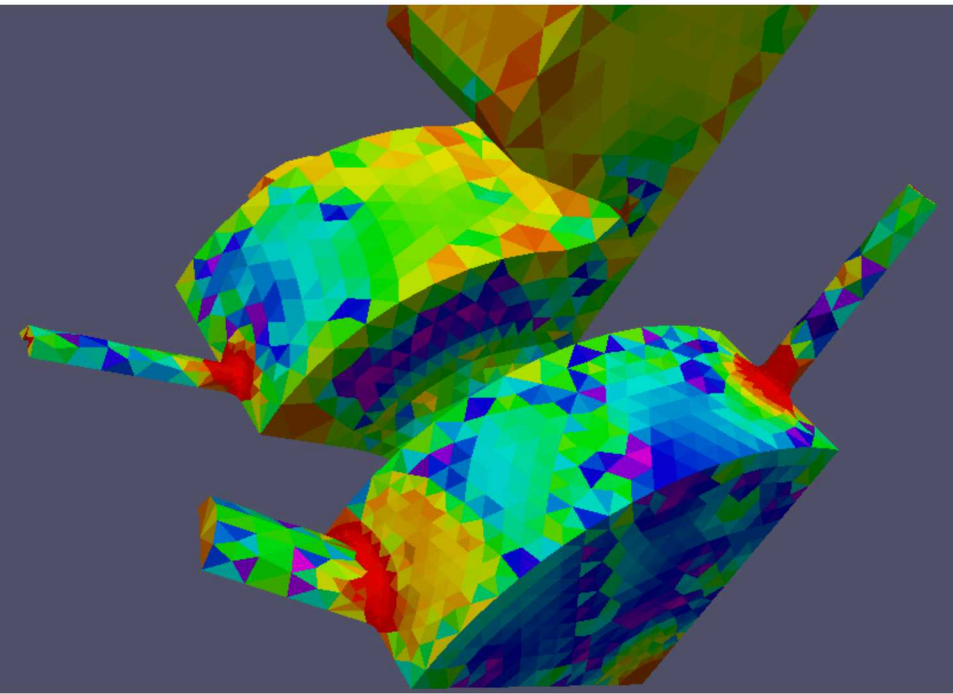
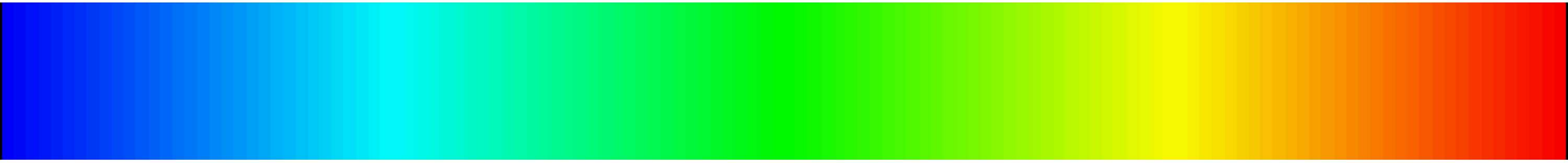
Rainbow: Color Deficiencies



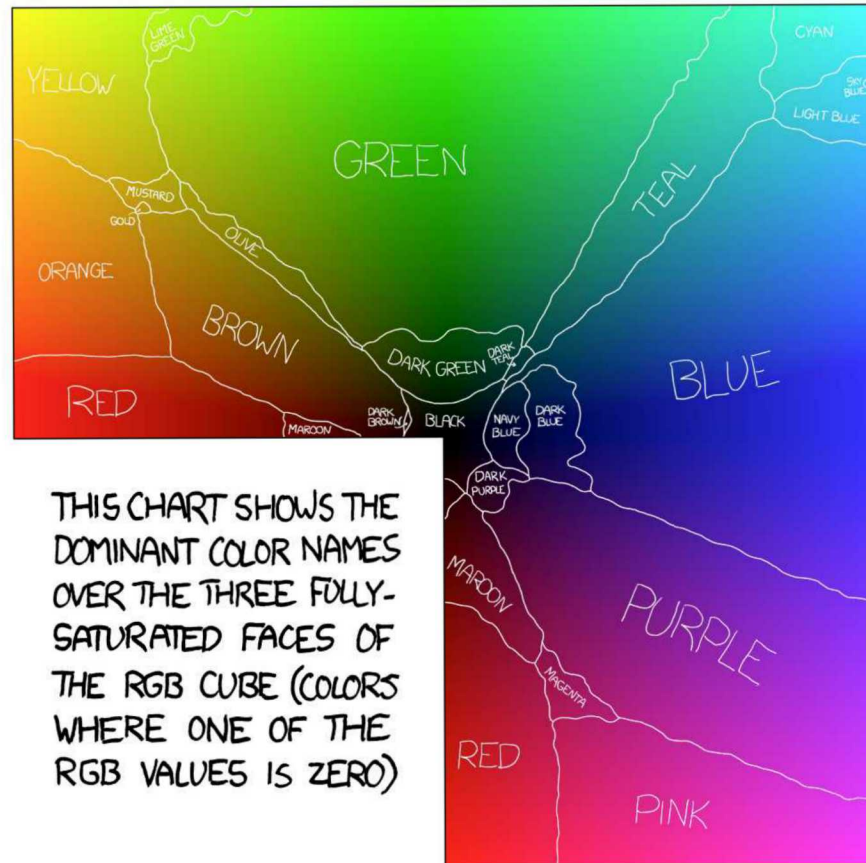
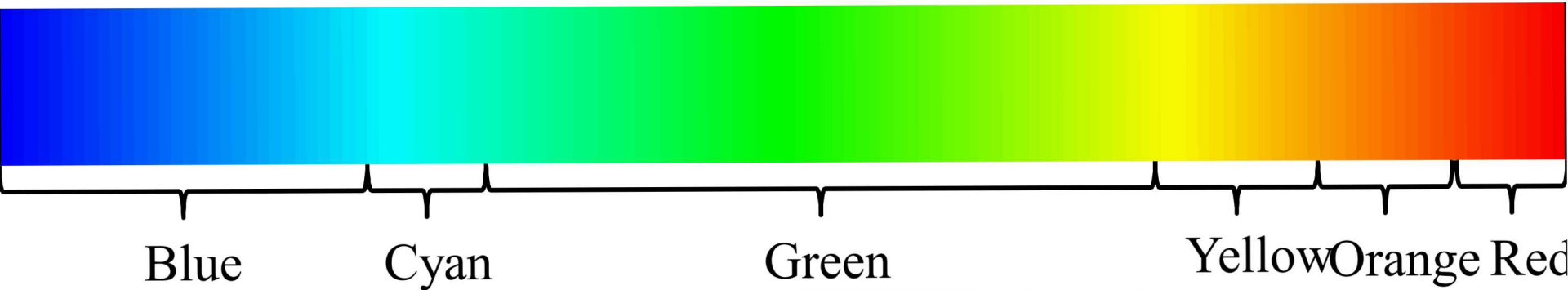
Rainbow: Implicit Ordering



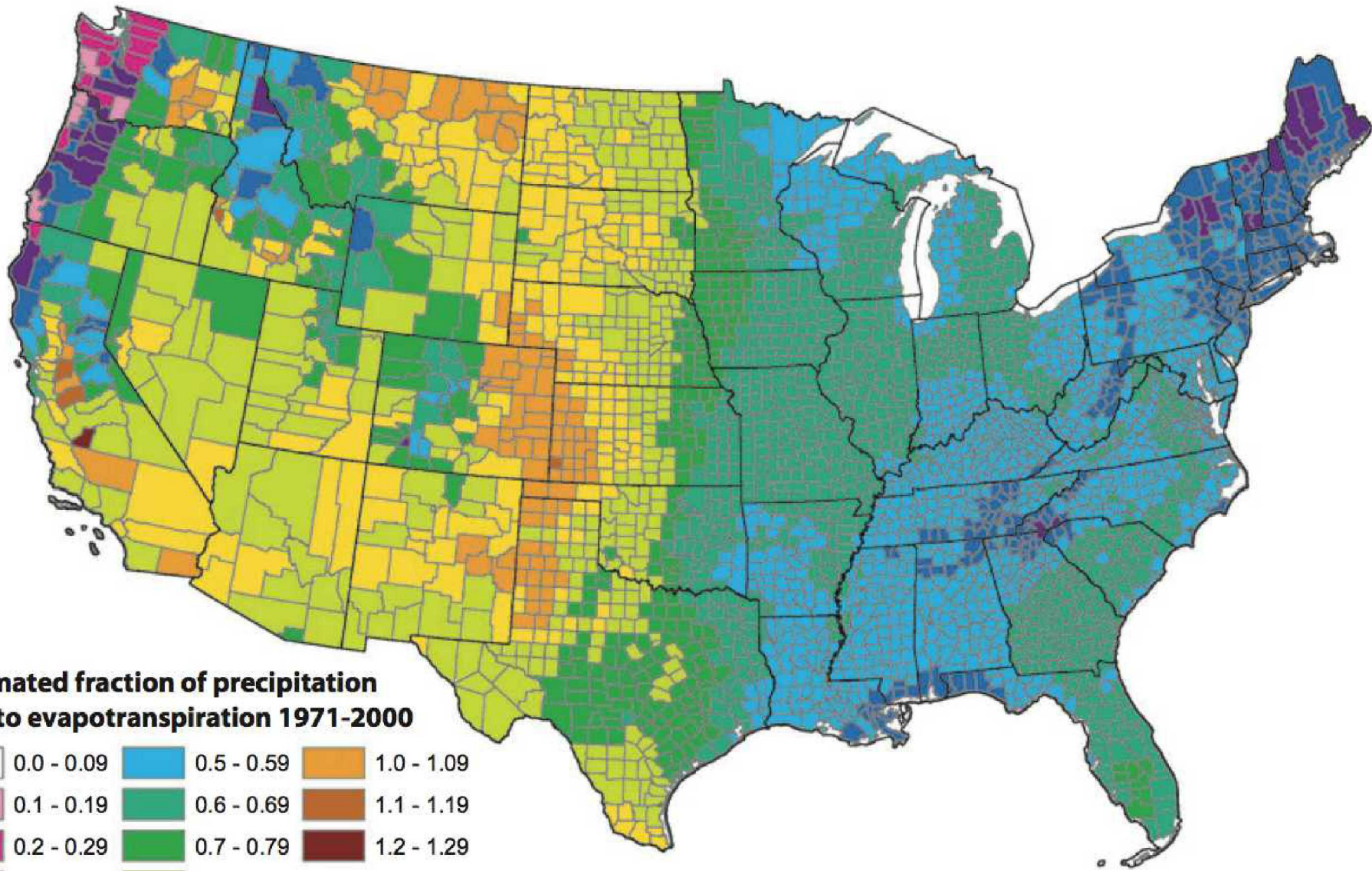
Rainbow: Implicit Ordering

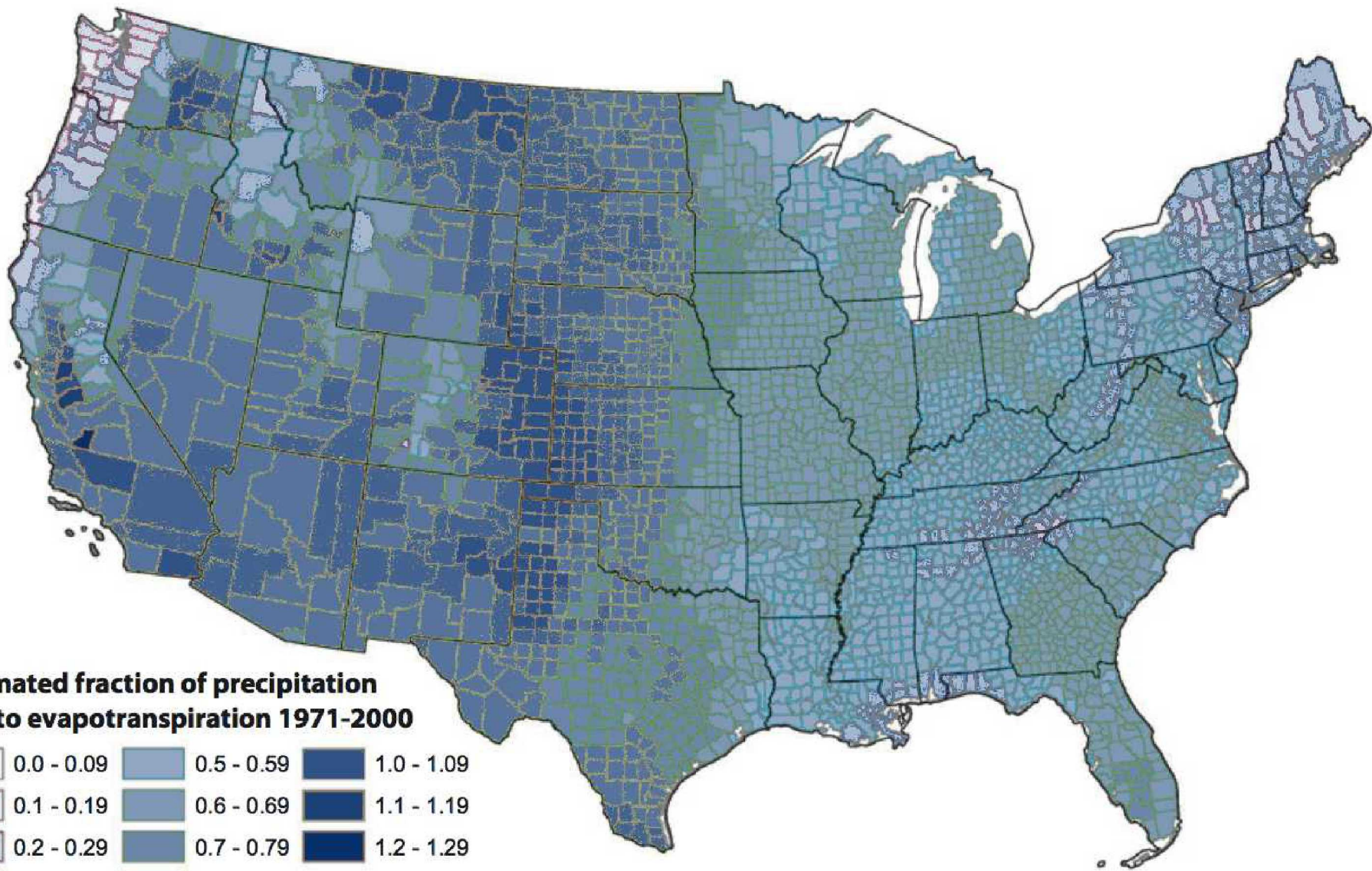


Rainbow: Perceptually Uneven



THIS CHART SHOWS THE DOMINANT COLOR NAMES OVER THE THREE FULLY-SATURATED FACES OF THE RGB CUBE (COLORS WHERE ONE OF THE RGB VALUES IS ZERO)





**Estimated fraction of precipitation
lost to evapotranspiration 1971-2000**

0.0 - 0.09	0.5 - 0.59	1.0 - 1.09
0.1 - 0.19	0.6 - 0.69	1.1 - 1.19
0.2 - 0.29	0.7 - 0.79	1.2 - 1.29
0.3 - 0.39	0.8 - 0.89	
0.4 - 0.49	0.9 - 0.99	

Reset ParaView

Edit → Reset Session



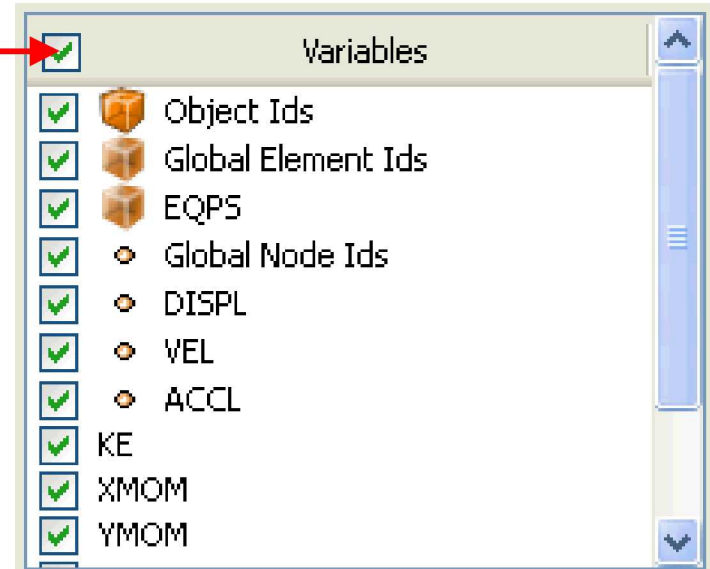
Loading Data with Time

1. Open the file can.ex2.
2. Select all variables.

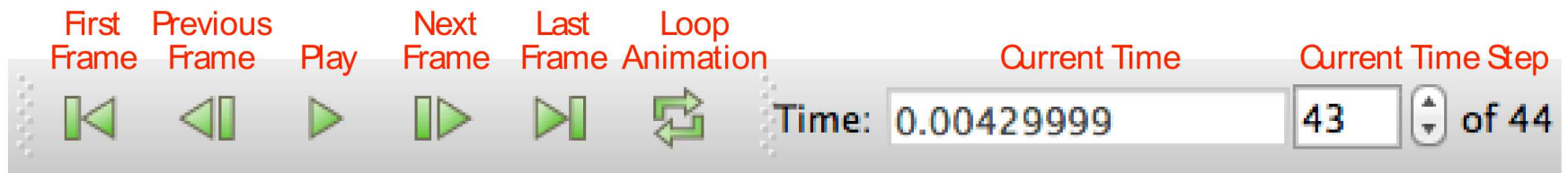
3.  Apply

4. 




5. 






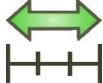
Animation Toolbar




Animation Pitfall

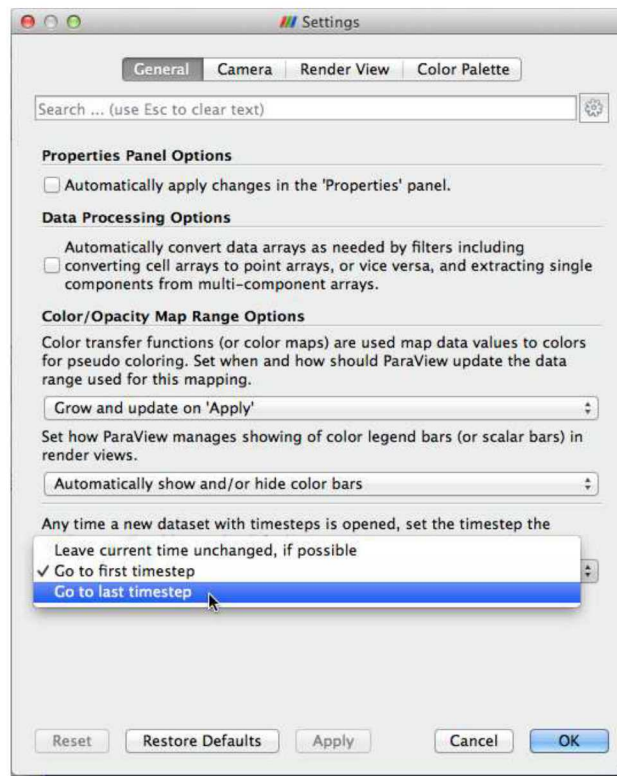
1. Go to first time step. 
2. Color by EQPS variable.
3. Play  (or skip to last time step ).

Animation Pitfall

1. Go to first time step. 
2. Color by EQPS variable.
3. Play  (or skip to last time step ).
4. Fix with Rescale to Data Range. 

Data Range Workarounds

- Go to representative time and hit 
- In Settings change On File Open to Goto last timestep.



Data Range Workarounds

- Set a custom range.



Data Range Workarounds

- Rescale to range over all timesteps



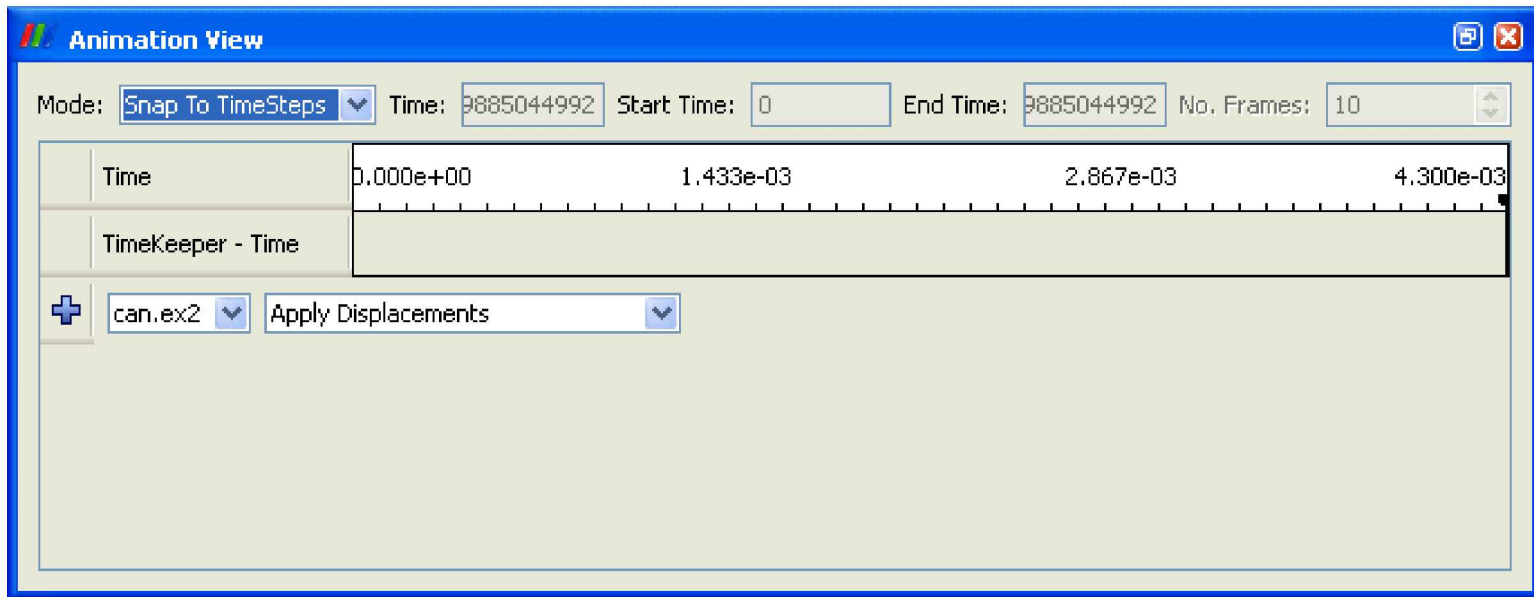
Reset ParaView

Edit → Reset Session



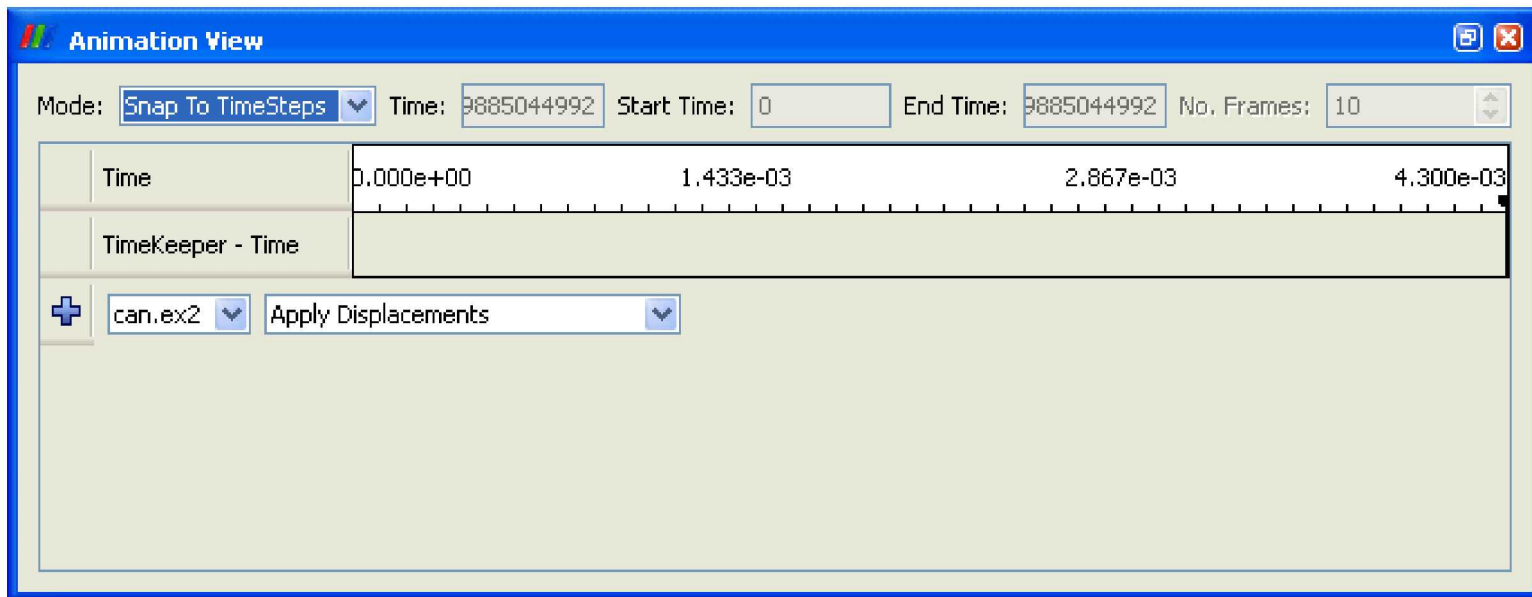
Animation View

View → Animation View



Animation View

View → Animation View



Animation Modes: Sequence, Real Time,
and Snap To TimeSteps

Changing Animation Timing

1. Open can.ex2. Load all variables.



2.  An icon showing a yellow cube with a green arrow pointing up and a red arrow pointing right, with "+Y" next to the green arrow.

3. View → Animation View

4. Change Mode to Real Time.

- Default animation duration is 10 sec.

5.  A green right-pointing triangle icon.

Changing Animation Timing

1. Open can.ex2. Load all variables.



2.  A small icon showing a yellow cube with a green arrow pointing up and a red arrow pointing right, with "+Y" written above the green arrow.

3. View → Animation View

4. Change Mode to Real Time.




- Default animation duration is 10 sec.

5.  A small green right-pointing triangle icon.


6. Change Duration to 60 sec.

7.  again.

Smoothing the Animation

1. Make sure can.ex2 highlighted.
2. Filters → Temporal → Temporal Interpolator
3.  Apply
4. Split view.  Show can.ex2 in one and TemporalInterpolator1 in the other. Link the cameras.
5. 







Adding Text Annotation

1. If you fell behind, you can reset ParaView and reload can.ex2.
2. Sources → Annotation → Text
3. Type a message in text edit box
4. A blue rectangular button with rounded corners. On the left side, there is a small icon of a shield with a green checkmark. To the right of the icon, the word "Apply" is written in a black, sans-serif font.



Text Position

Text Position

Use Window Location



Lower Left Corner

<input type="text" value="0.025267"/>		<input type="text" value="0.010000"/>	
---------------------------------------	---	---------------------------------------	---

Annotate Time

1. Sources → Annotation → Annotate Time
2. 

Save Screenshot/Animation



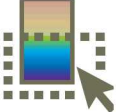
1. Choose File →  Save Screenshot
2. Complete the following dialogs.
3. Chose File →  Save Animation
4. Complete the following dialogs.

Reset ParaView

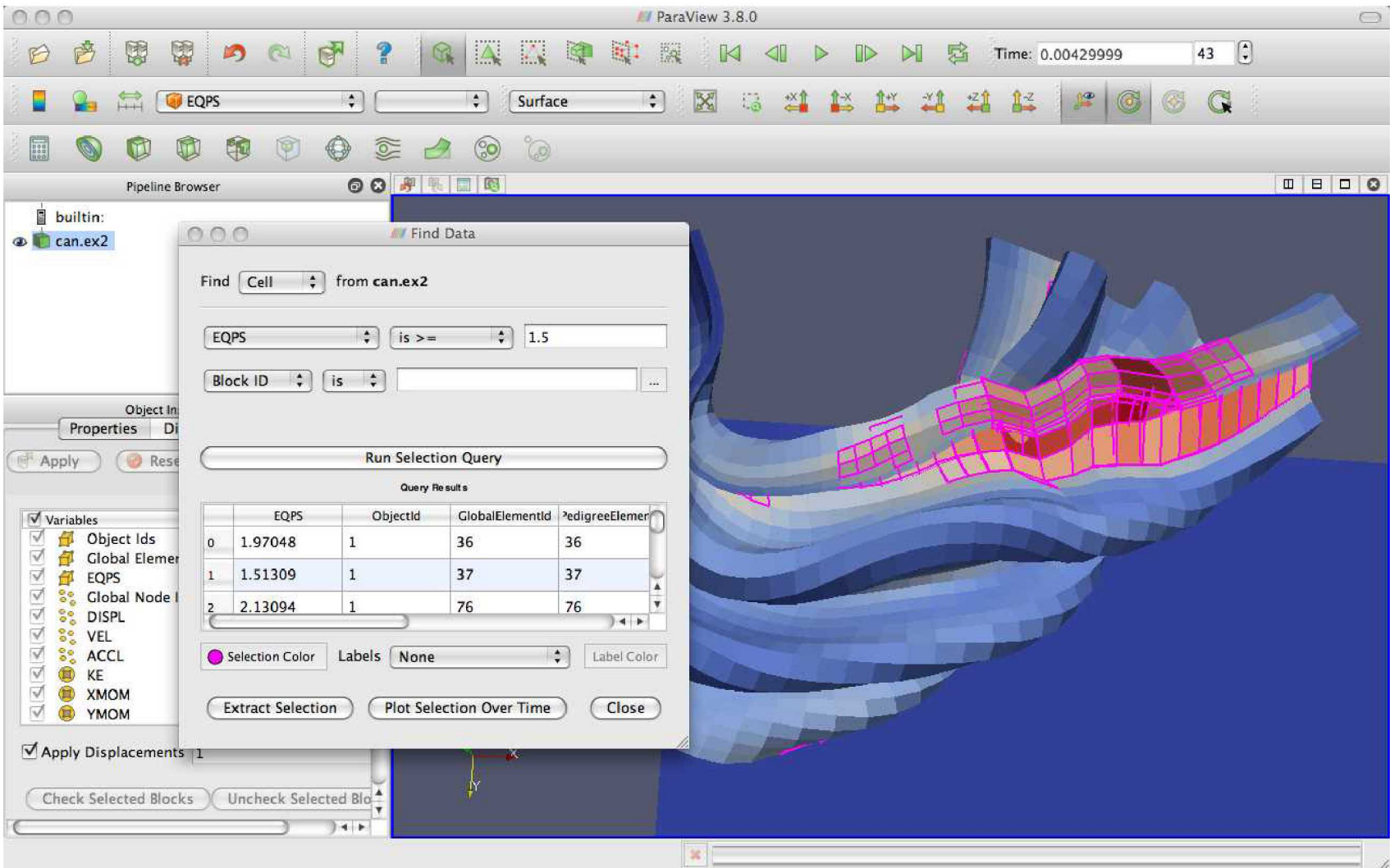
Edit → Reset Session



Query-Based Selection

1. Open can.ex2. All variables. 
2. Go to last time step. 
3. Edit → Find Data. 
4. Top combo box: find Cells.
5. Next row: EQPS, is \geq , and 1.5.
6. Click Run Selection Query.

Query-Based Selection



ParaView 3.8.0

Time: 0.00429999 43

EQPS Surface

Pipeline Browser

- builtin:
 - can.ex2

Object Inspector

Properties

Apply Reset

Variables

- Object Ids
- Global Element
- EQPS
- Global Node I
- DISPL
- VEL
- ACCL
- KE
- XMOM
- YMOM

Apply Displacements 1

Check Selected Blocks Uncheck Selected Blo

Find Data

Find Cell from can.ex2

EQPS is >= 1.5

Block ID is

Run Selection Query

Query Results

	EQPS	ObjectId	GlobalElementId	PedigreeElement
0	1.97048	1	36	36
1	1.51309	1	37	37
2	2.13094	1	76	76

Selection Color Labels None Label Color

Extract Selection Plot Selection Over Time Close

Brush Selection



Face Cell Selection
(shortcut: s)



Face Point Selection
(shortcut: d)



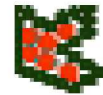
Through Cell Selection
(shortcut: f)



Through Point Selection
(shortcut: g)



Select Cells (polygon)



Select Points (polygon)



Block Selection
(shortcut: b)



Interactively Select Cells



Interactively Select Points

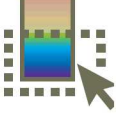


Hover Point Query

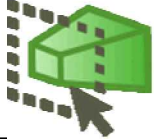
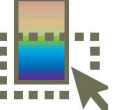

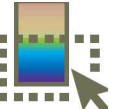



Hover Cell Query


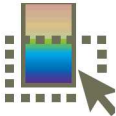
Selections

1. Open Find Data. 
2. Make various brush selections.
3. Observe results in the Find Data dialog box.
4. Play with the Invert Selection and Show Frustum options.


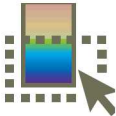
Frustum vs. Id Selections

1. Make a Select Cells Through 
2. Turn on Show Frustum in Find Data.  Rotate 3D view.
3. Play 
4. Hit Freeze Selection in Find Data. 
5. Play 

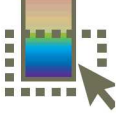


Adding Labels

1. Go to the last time step. 
2. Open Find Data. 
3. Create query Global ID is min. Click Run Selection Query.
4. In the Cell Labels chooser, select EQPS.

Adding Labels

1. Go to the last time step. 
2. Open Find Data. 
3. Create query Global ID is min. Click Run Selection Query.
4. In the Cell Labels chooser, select EQPS.
5. When you are done, turn off the EQPS labels.

Plot Selection Over Time

1. Open Find Data. 
2. Create query EQPS is max. Click Run Selection Query.
3. Click Plot Selection Over Time 
4. 

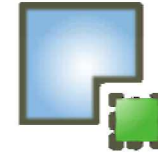
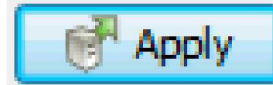
Reset ParaView

Edit → Reset Session



Extracting a Selection

1. Open can.ex2. Load all variables.
2. Perform a sizeable selection
3. Add the extract selection
- 4.





Reset ParaView



Edit → Reset Session




Make an Animation

1. Sources → Geometric Shapes → Sphere, 
2. Make animation view visible.
3. Change No. Frames to 50.
4. Select Sphere1, Start Theta, press 



Make an Animation

1. Sources → Geometric Shapes → Sphere, 
2. Make animation view visible.
3. Change No. Frames to 50.
4. Select Sphere1, Start Theta, press 
5. Double-click Sphere1 – Start Theta
6. Make New keyframe.
7. First keyframe value→360, second keyframe time→0.5 value→0.
8. Click OK.

Animating Two Properties

1. Open Sphere1 – Start Theta.
2. Delete the first keyframe (at time 0).
3. Click OK.
4. Select Sphere1 – End Theta, press .
5. Open Sphere1 – End Theta.
6. Change second key frame time to 0.5.

Camera Orbit





1. Place the camera where the orbit should start.
2. Make animation view visible.
3. Select Camera – Orbit, press 
4. Accept default values (click OK).
5. 

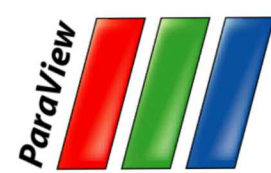
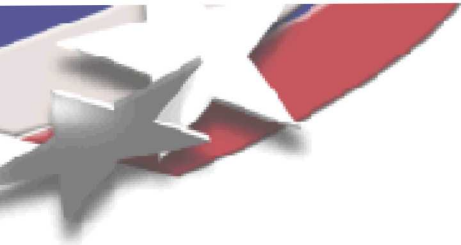
Reset ParaView

Edit → Reset Session



Following Data

1. Open can.ex2. Load all variables. 
2. 
3. Make animation view visible.
4. Select Camera – Follow Data, press 
5. 



Batch Scripting



Python Batch Scripting

- Interpreter: View → Python Shell.
 - Shell does tab completion and history.
 - External programs pvpython and pvbatch.
 - Can add bindings to external interpreters.
- Run Python scripts or bind to macros.
- Can trace actions or capture state.
- (Batch scripting is a different set of bindings than programmable sources/filters.)

Reset ParaView

Edit → Reset Session

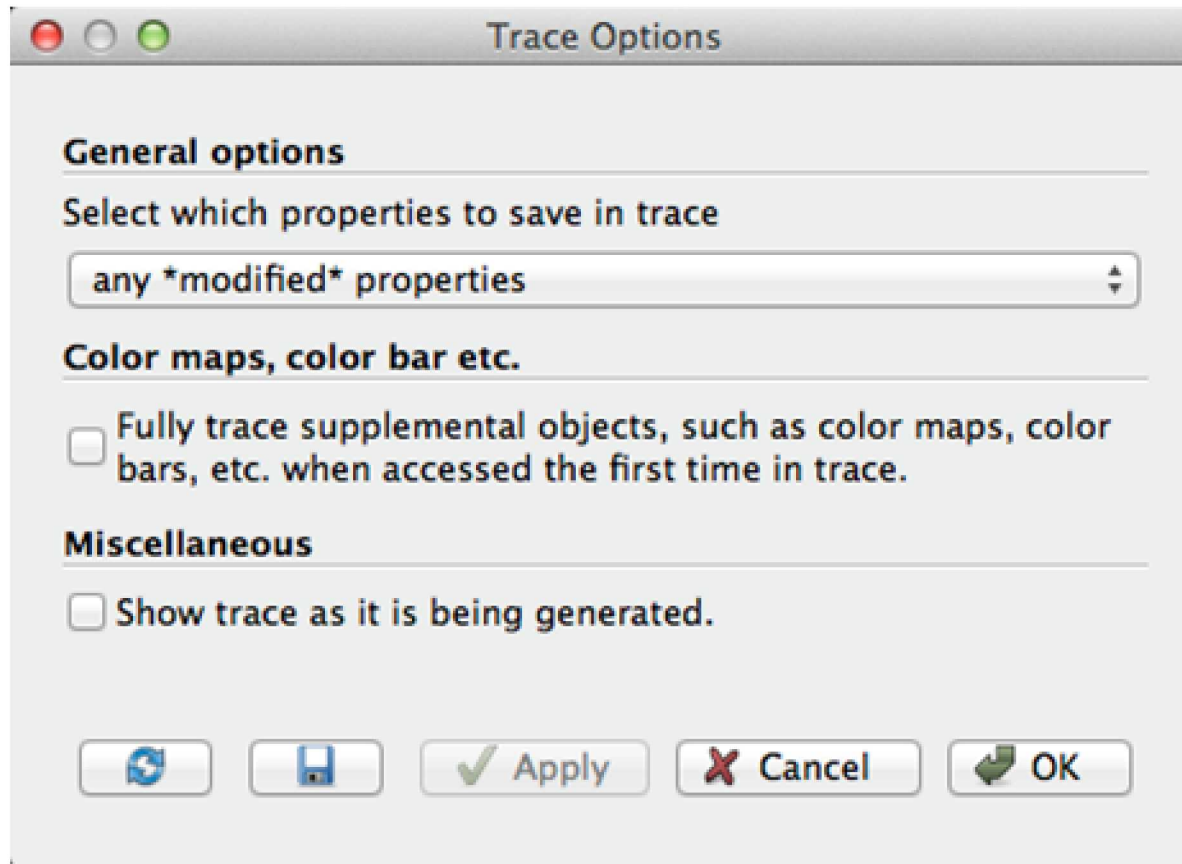


In the top-left corner, there are several overlapping, semi-transparent stars in white, red, and blue, resembling the colors of the American flag.

Tracing ParaView State

1. Select Tools → Start Trace.


Python Tracing Options



Tracing ParaView State

1. Select Tools → Start Trace.
2. Accept defaults (click OK).
3. Build a simple pipeline (e.g. create a sphere source and clip it).
4. Select Tools → Stop Trace.
5. An editing window will open.

Adding a Macro

1. In Python edit window menu, select File → Save As Macro...
2. Chose a descriptive name. Save in directory provided.
3. Close Python edit window.
4. Select Edit → Reset Session. 
5. Activate your macro (on toolbar or Macros menu).

Reset ParaView

Edit → Reset Session



Creating a Pipeline

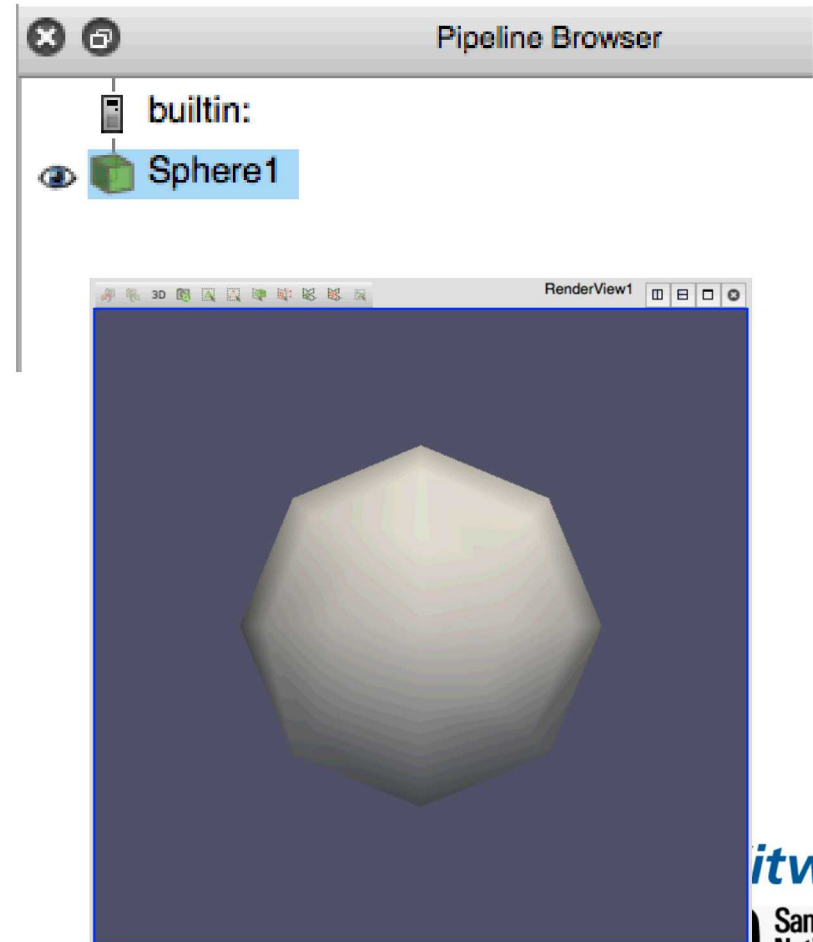
- In Python Shell create a source

```
sphere = Sphere()
```

```
Show()
```

```
Render()
```

```
ResetCamera()
```



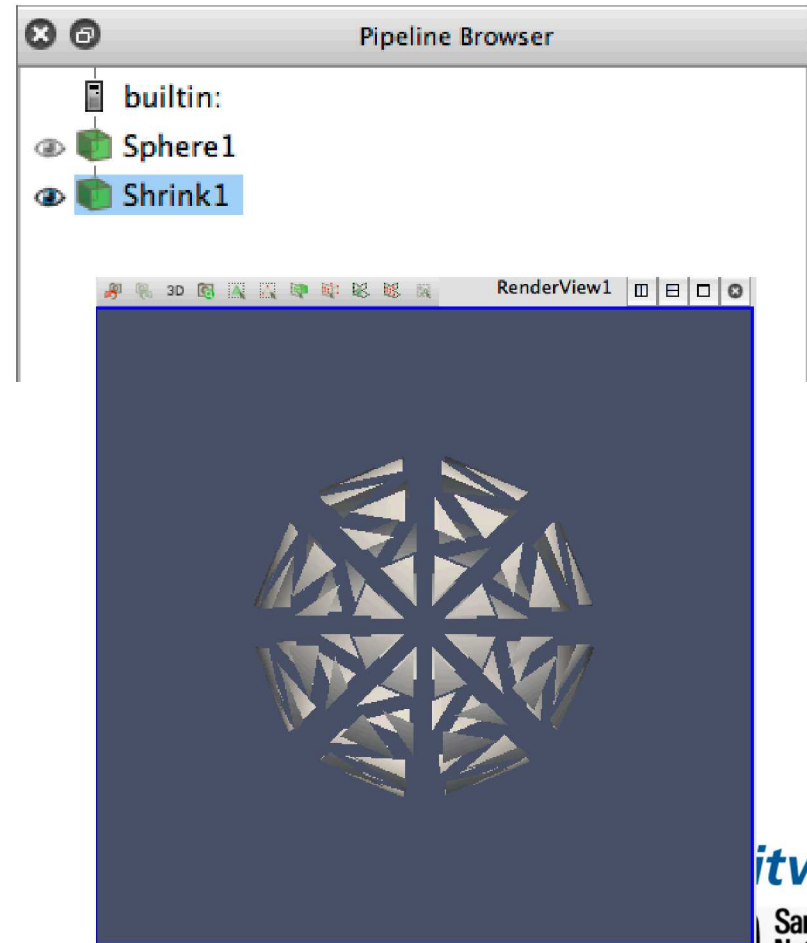
Creating a Pipeline

- In Python Shell create a source

```
sphere = Sphere()  
Show()  
Render()  
ResetCamera()
```

- Then feed that into a filter

```
Hide()  
shrink = Shrink()  
Show()  
Render()
```



Changing Properties

- Get listing of sphere's properties

```
dir(sphere)
```

- Examine and then change one

```
print sphere.ThetaResolution
```

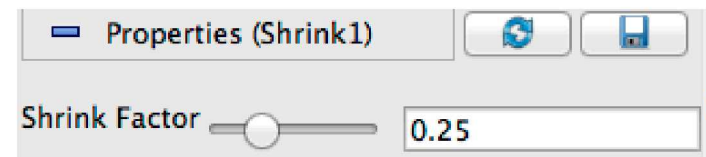
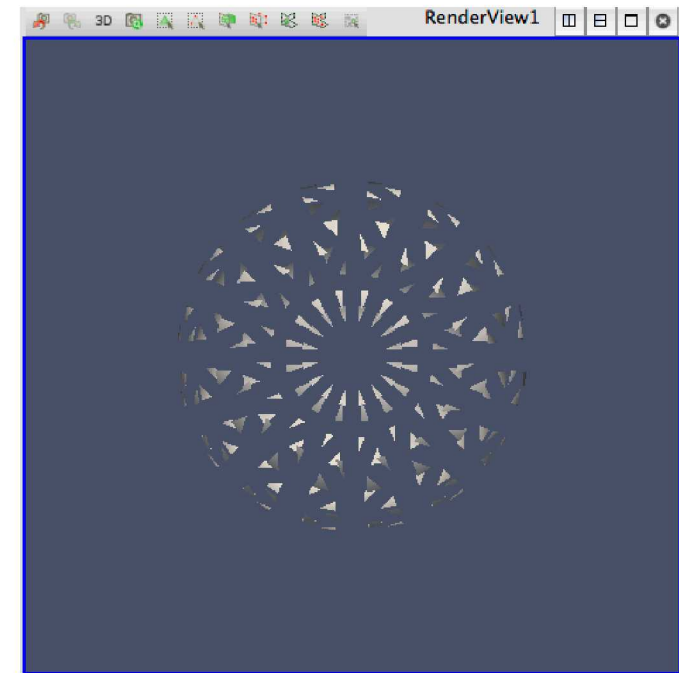
```
sphere.ThetaResolution = 16
```

```
Render()
```

- Change a different filter

```
shrink.ShrinkFactor = 0.25
```

```
Render()
```



Branching and Merging Pipeline

- Feed sphere's output into a second filter

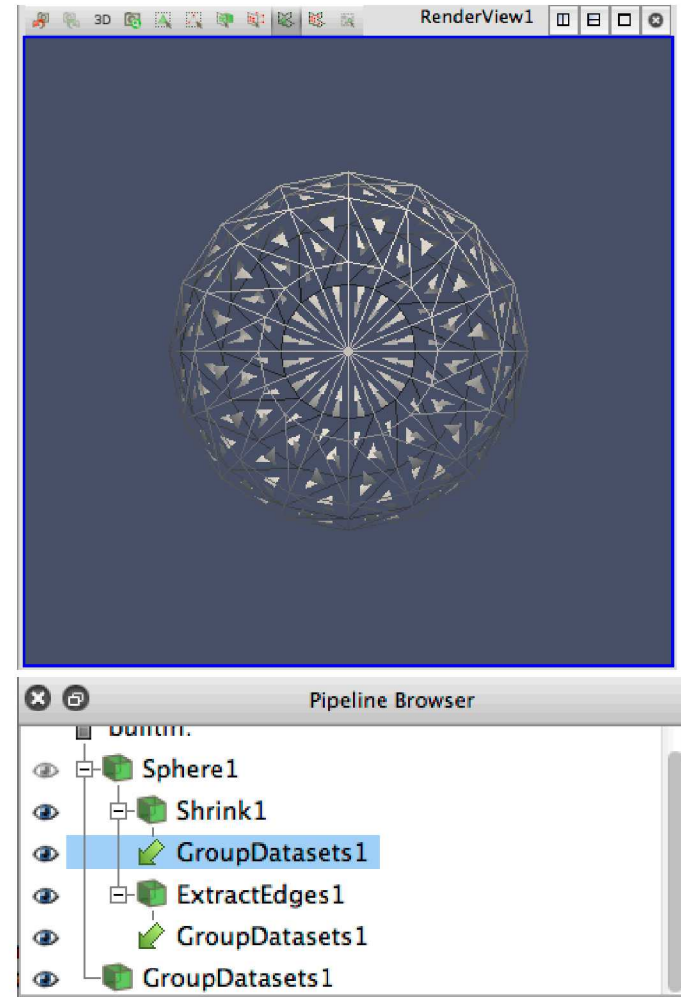
```
wireframe = ExtractEdges (
    Input=sphere)

Show ()
Render ()
```

- Feed two filters into one

```
group = GroupDatasets (
    Input=[shrink,wireframe])

Show ()
Hide (shrink)
Hide (wireframe)
Render ()
```



Finding Pipeline Objects

- `GetSources ()` Returns a list of objects listed in the Pipeline Browser
- `GetActiveSource ()` Returns the object selected in the Pipeline Browser
- `SetActiveSource ()` Sets the object selected in the Pipeline Browser to the given object

Reset ParaView

Edit → Reset Session



Reading Files

- Open a dataset.

```
reader = OpenDataFile('<path>/disk_out_ref.ex2')  
Show()  
Render()  
ResetCamera()
```

- Use file browser to help identify `<path>`.
- Common locations:
 - Mac: `/Applications/ParaView-X.X.X.app/Contents/examples`
 - Windows: `C:/Program Files/ParaView X.X.X/examples`
 - Linux: `/usr/local/lib/paraview-X.X/examples`

Querying Attributes

- Examine all point data ranges

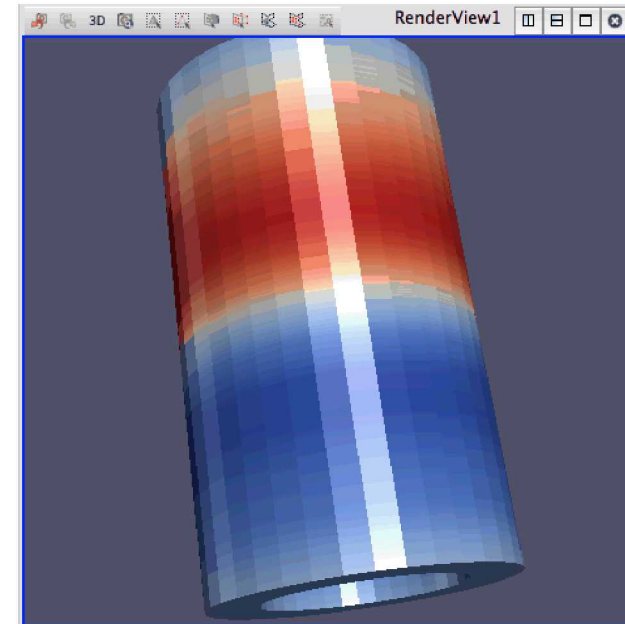
```
pd = reader.PointData
for ai in pd.values():
...     print ai.GetName(), ai.GetNumberOfComponents(),
...     for i in xrange(ai.GetNumberOfComponents()):
...         print ai.GetRange(i),
...     print
```

```
AsH3 1 (0.08047682046890259, 0.18483947217464447)
CH4 1 (0.0, 0.0011702391784638166)
GaMe3 1 (0.00022284436272457242, 0.007213925942778587)
GlobalNodeId 1 (1.0, 8499.0)
H2 1 (0.8076131343841553, 0.9176878929138184)
PedigreeNodeId 1 (1.0, 8499.0)
Pres 1 (0.006785521749407053, 0.028818512335419655)
Temp 1 (293.1499938964844, 913.1500244140625)
V 3 (-19.949113845825195, 19.949113845825195) (-19.949113845825195, 19.949113845825195)
(-21.1259765625, 6.7119011878967285)
```

Coloring Data

- Get a hold of and manipulate display properties

```
readerRep = GetRepresentation()  
readerRep.DiffuseColor = [0, 0, 1]  
readerRep.SpecularColor = [1, 1, 1]  
readerRep.SpecularPower = 128  
readerRep.Specular = 1  
Render()
```



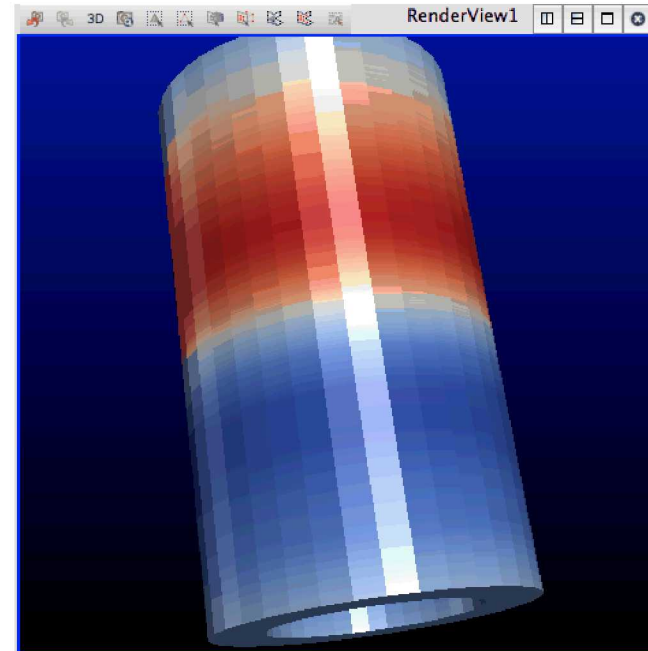
- Now assign a color transfer function

```
ColorBy(readerRep, ('POINTS', 'Pres'))  
UpdateScalarBars()  
Render()
```

Controlling the View

Get a hold of, then manipulate view properties

```
view = GetActiveView()  
view.Background = [0, 0, 0]  
view.Background2 = [0, 0, 0.6]  
view.UseGradientBackground = True  
Render()
```



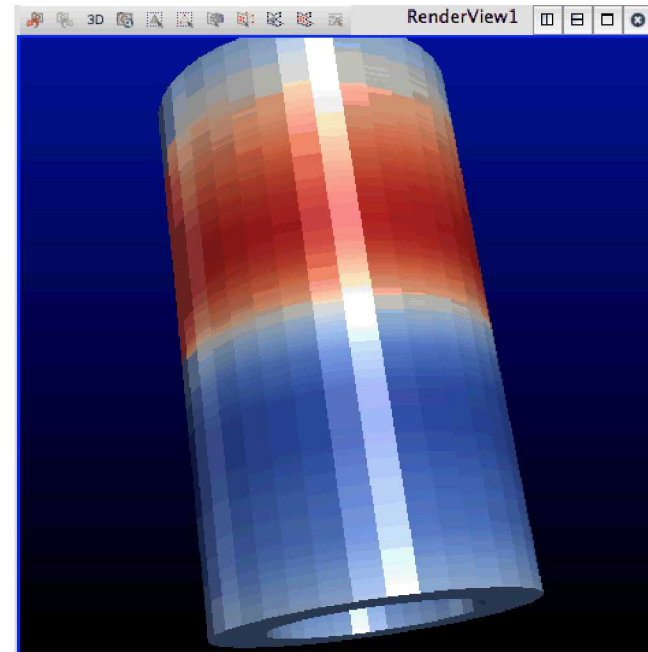
Controlling the View

Get a hold of, then manipulate view properties

```
view = GetActiveView()
view.Background = [0, 0, 0]
view.Background2 = [0, 0, 0.6]
view.UseGradientBackground = True
Render()
```

Animate the camera

```
x, y, z = view.CameraPosition
print x, y, z
for iter in xrange(0, 10):
    x = x + 1
    y = y + 1
    z = z + 1
    view.CameraPosition = [x, y, z]
    print x, y, z
    Render()
```



Save Results

Save Data

```

plot = PlotOverLine()
plot.Source.Point1 = [0,0,0]
plot.Source.Point2 = [0,0,10]
writer = CreateWriter('<path>/plot.csv')
writer.UpdatePipeline()

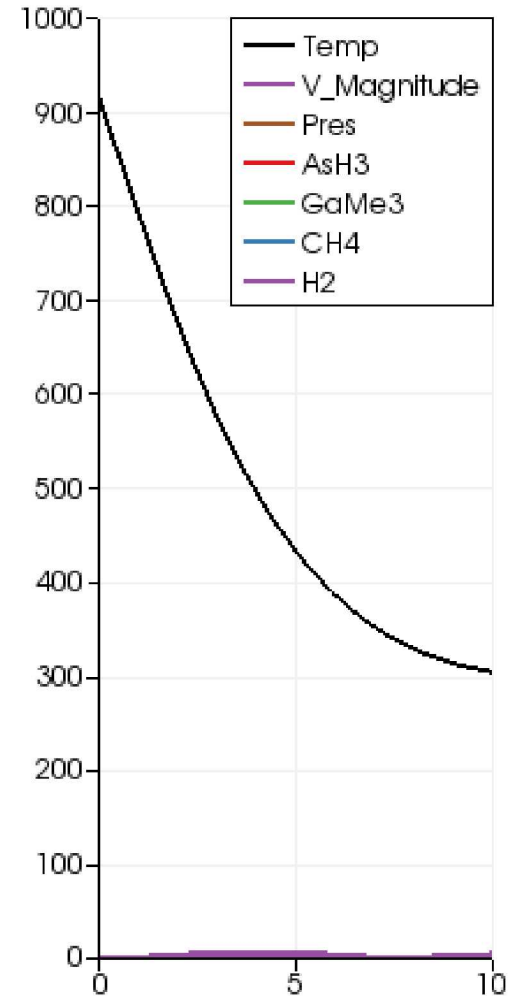
```

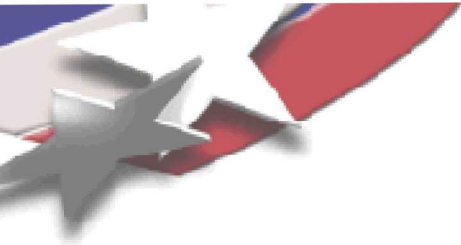
Then save screen capture of plot

```

plotView = CreateView('XYChartView')
Show(plot)
Render()
SaveScreenshot('<path>/plot.png')

```





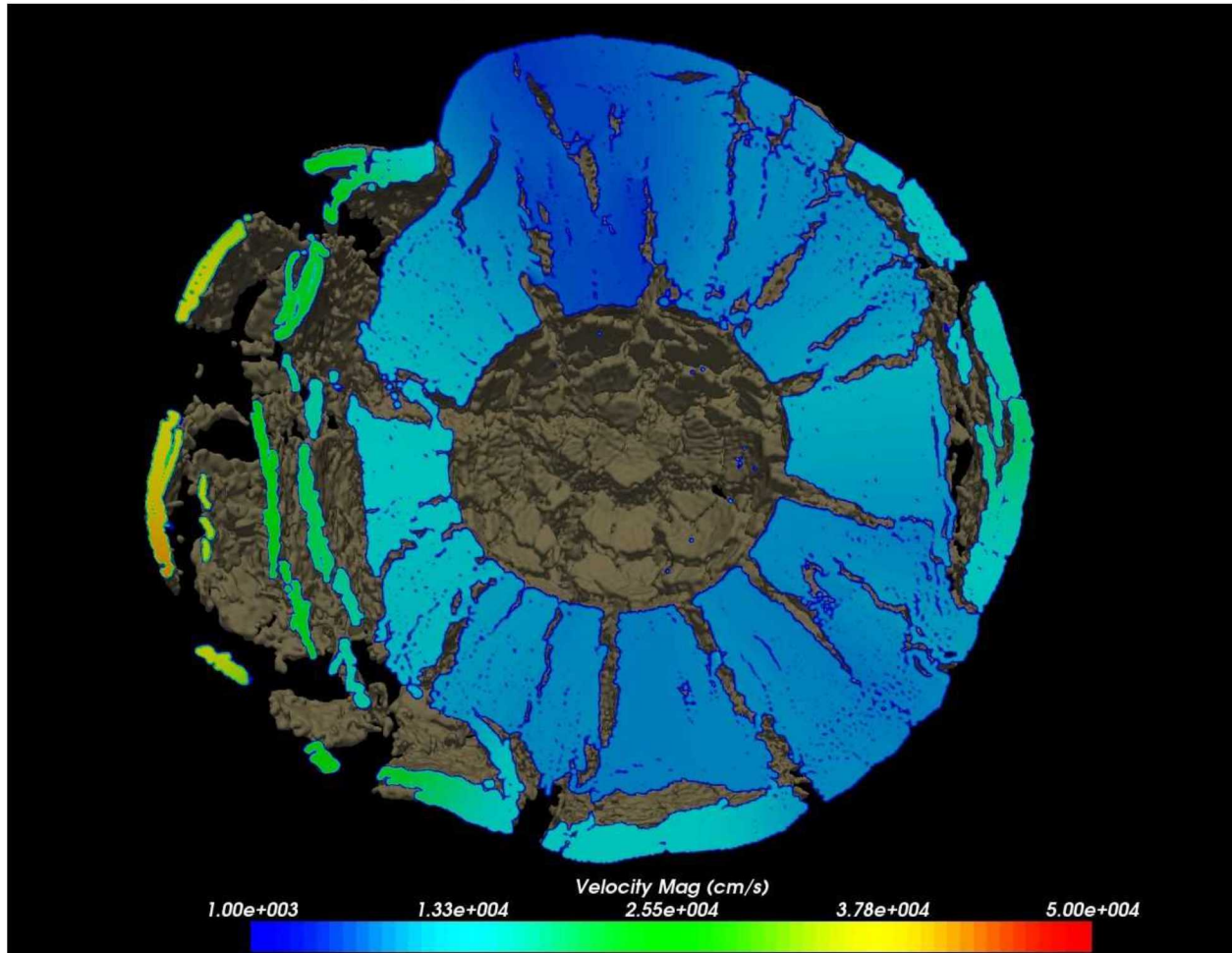
Visualizing Large Models



Golevka Asteroid vs. 10 Megaton Explosion

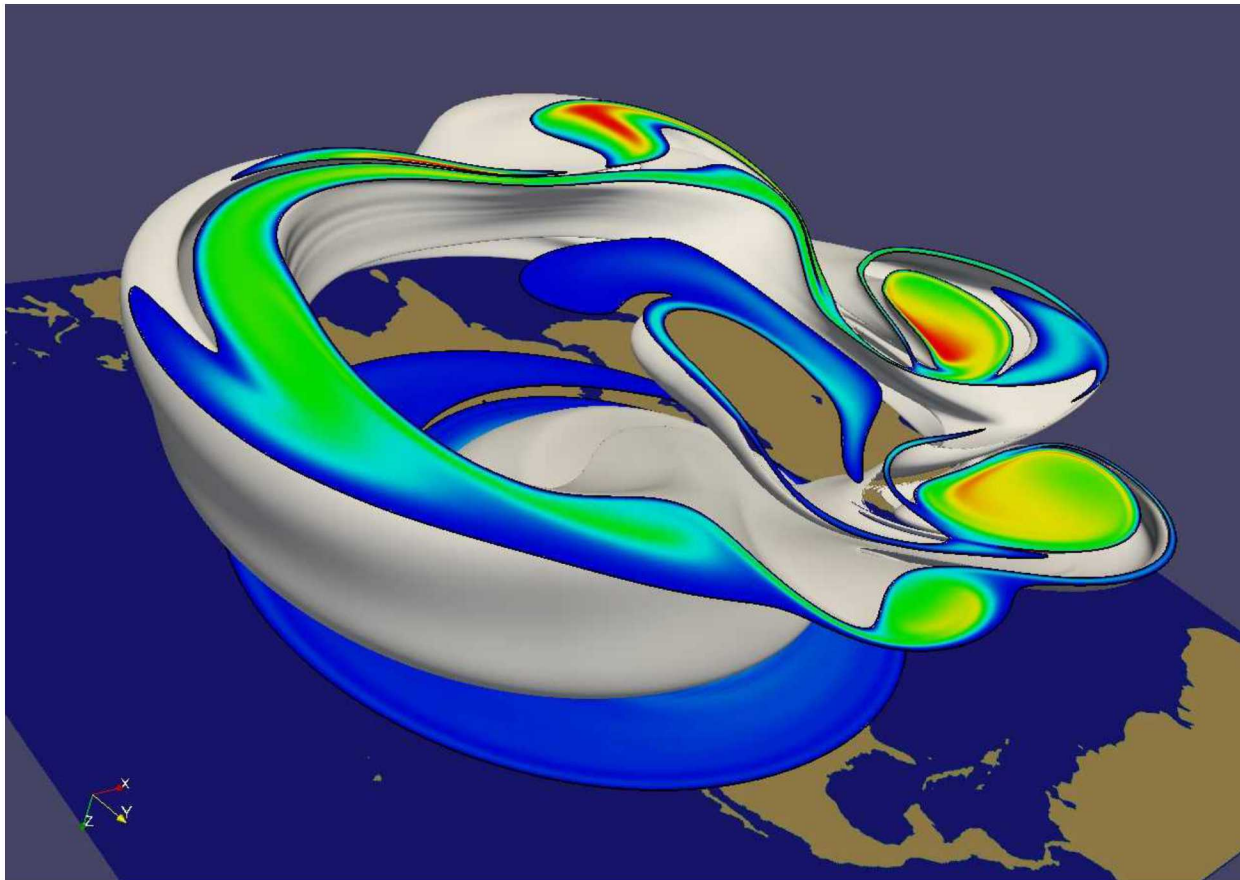


- CTH shock physics, over 1 billion cells



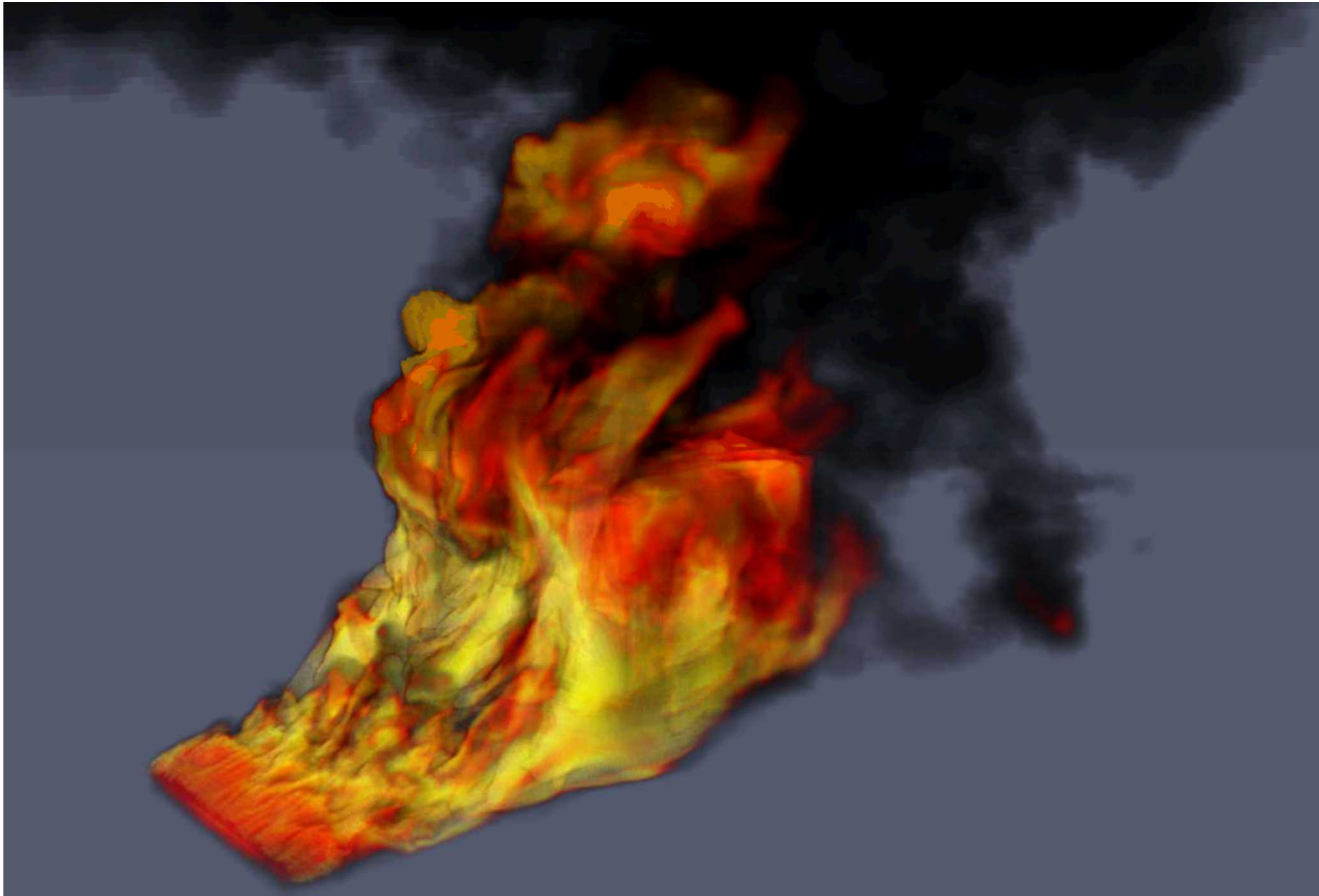
Polar Vortex Breakdown

- SEAM Climate Modeling, 1 billion cells (500 million cells visualized).

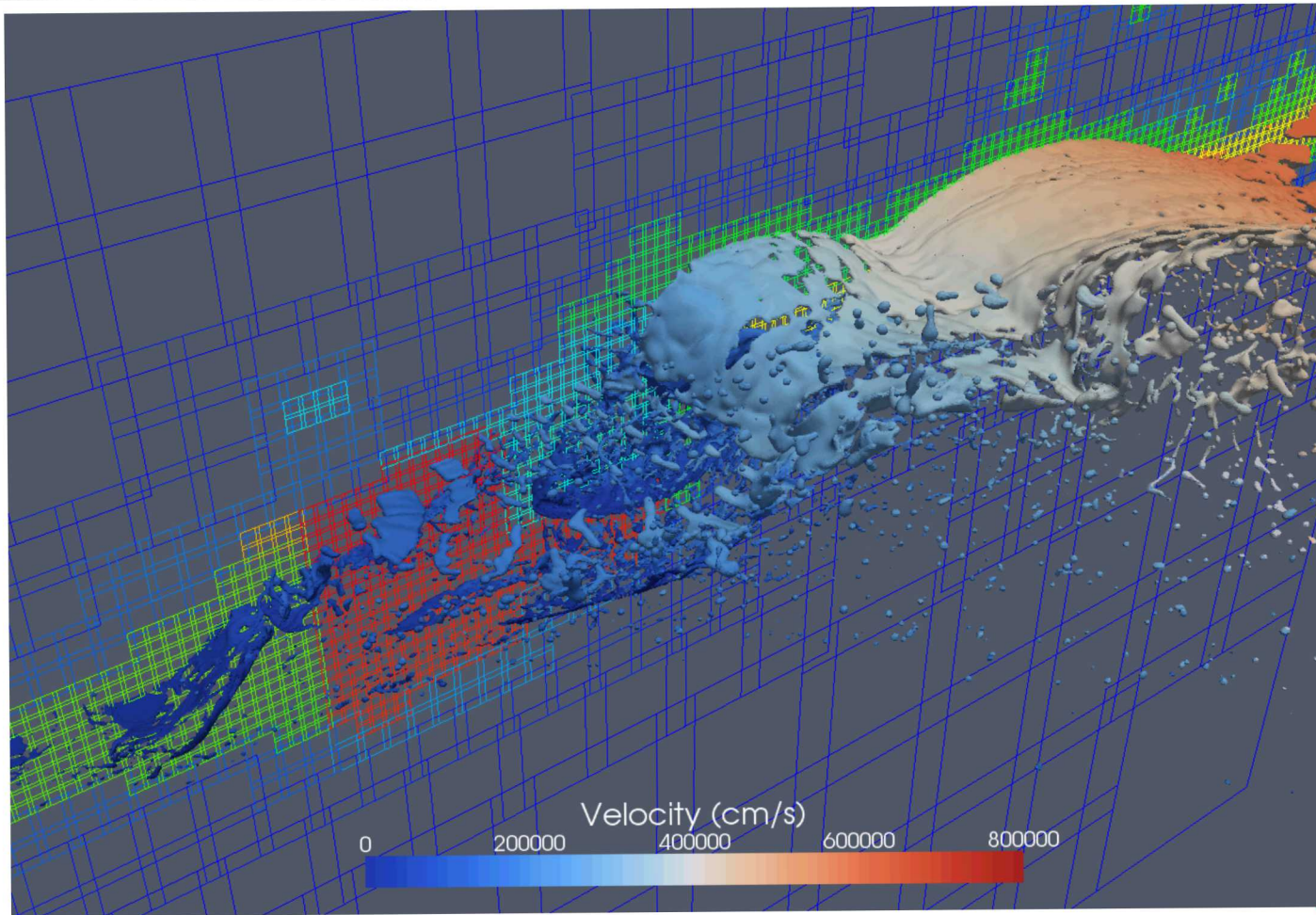


Objects-in-Crosswind Fire

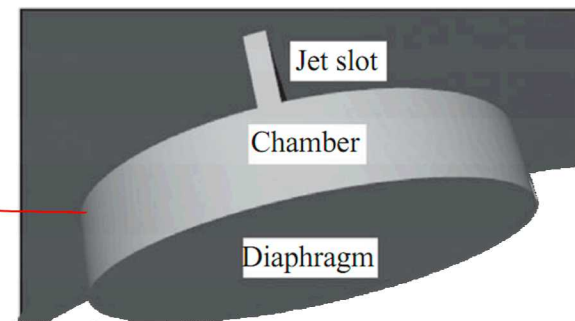
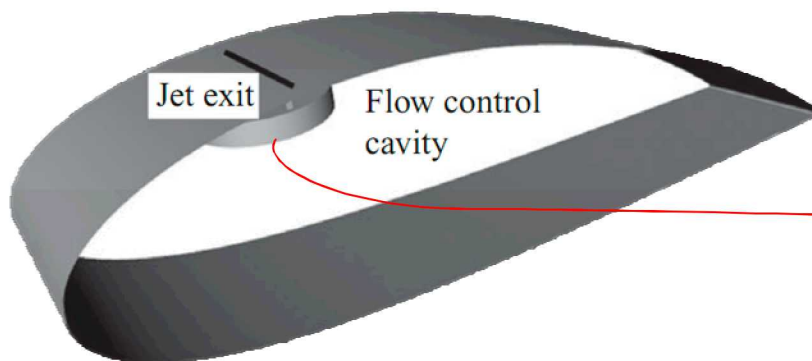
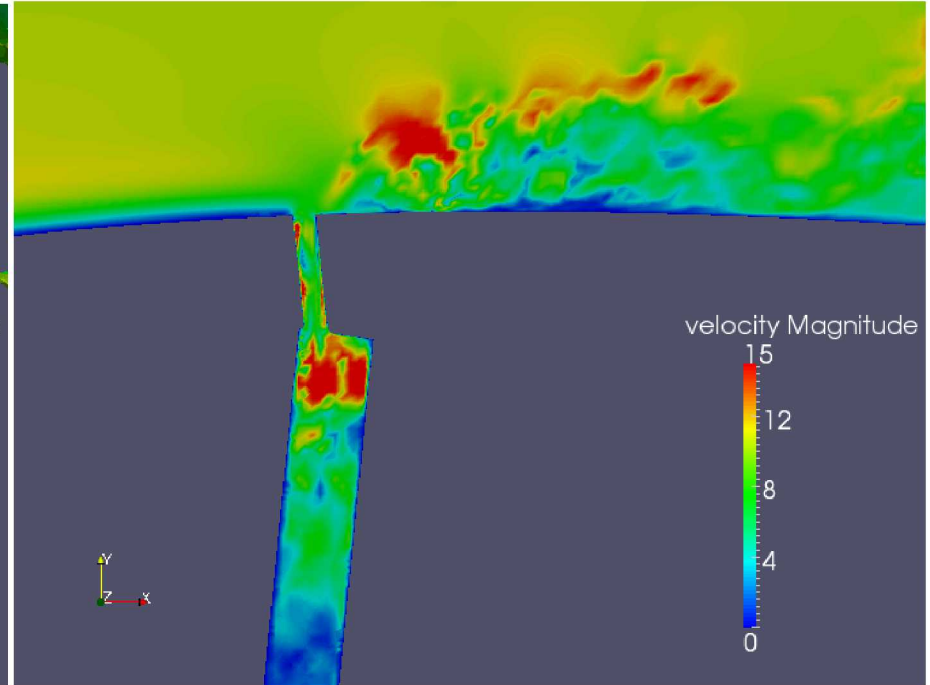
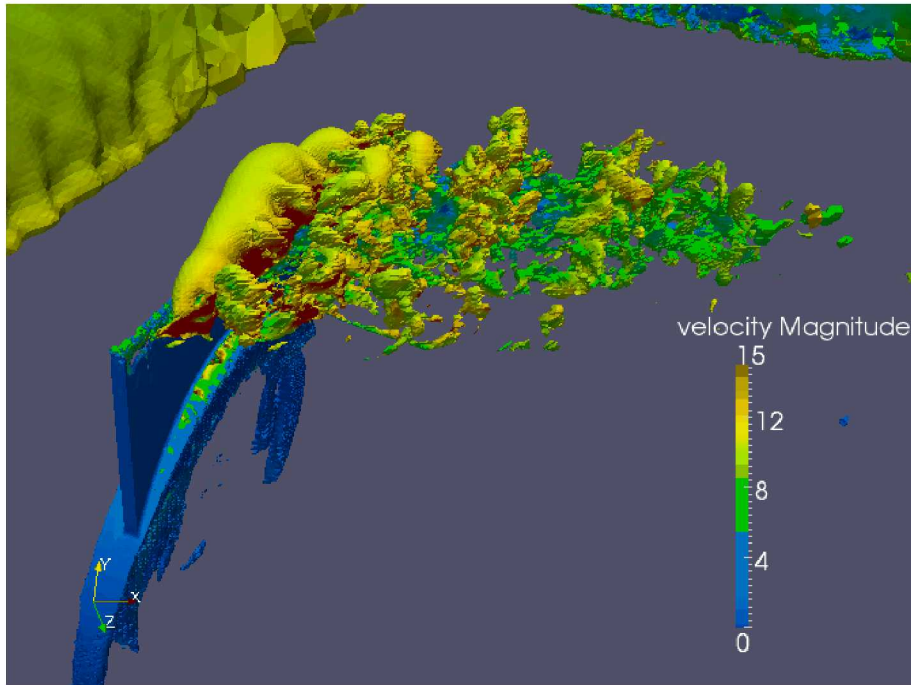
- Coupled SIERRA/Fuego/Syrinx/Calore, 10 million unstructured hexahedra



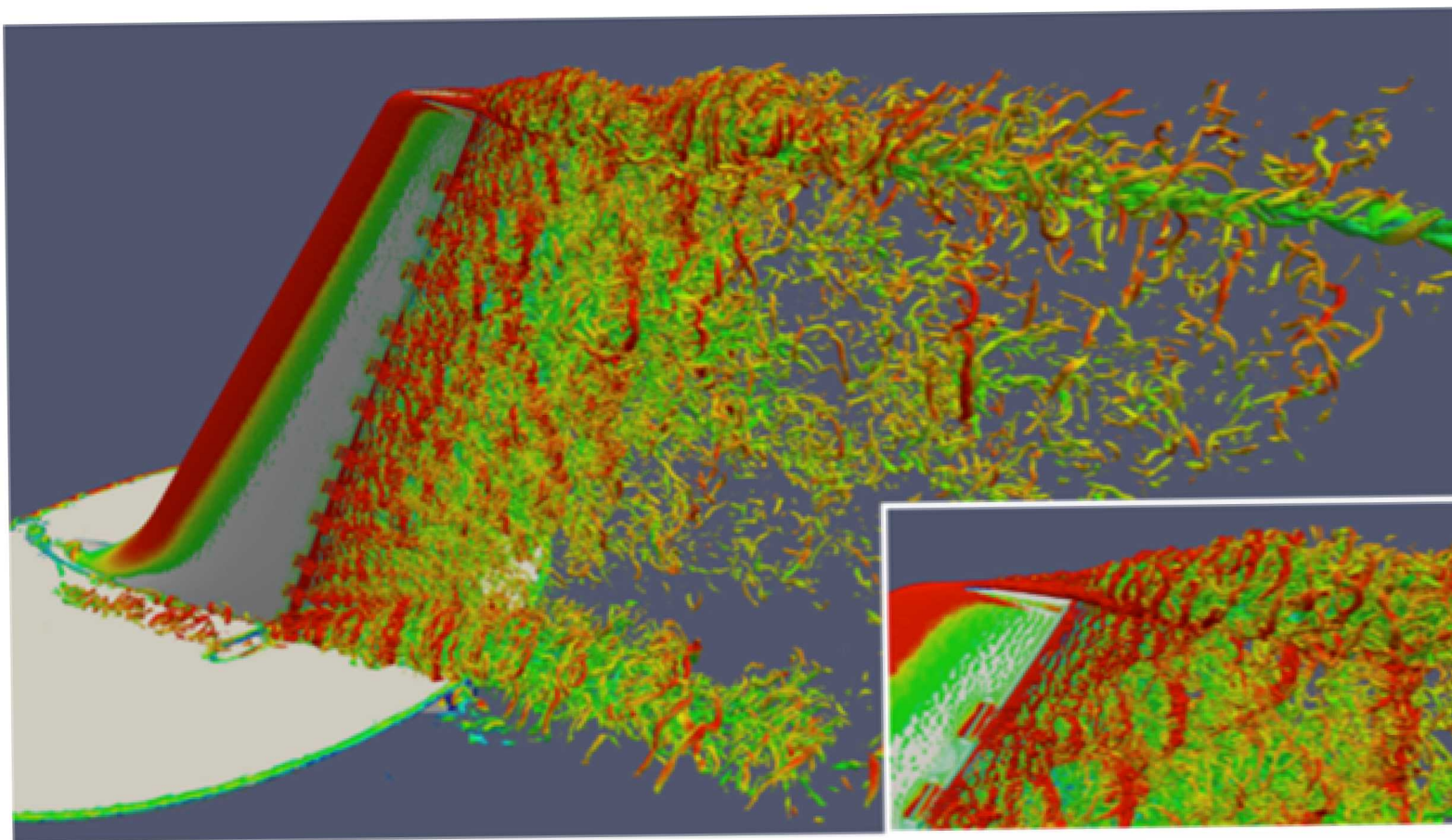
Large Scale AMR



Wing with Unsteady Crossflow from Synthetic Jet (3.3B tet)

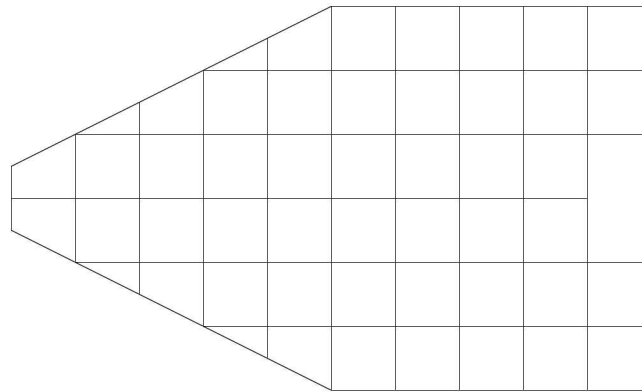


Wake of Deflected Wing Flap



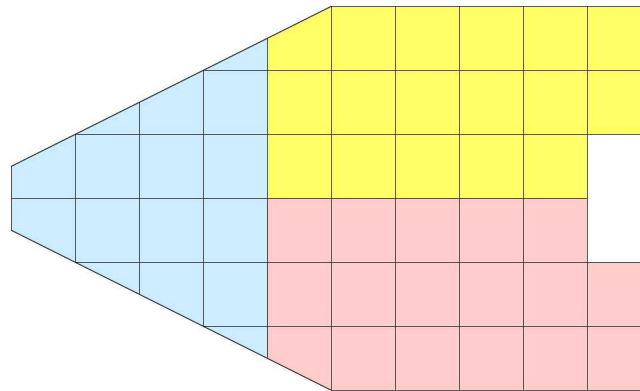
Data Parallel Pipelines

- Duplicate pipelines run independently on different partitions of data.



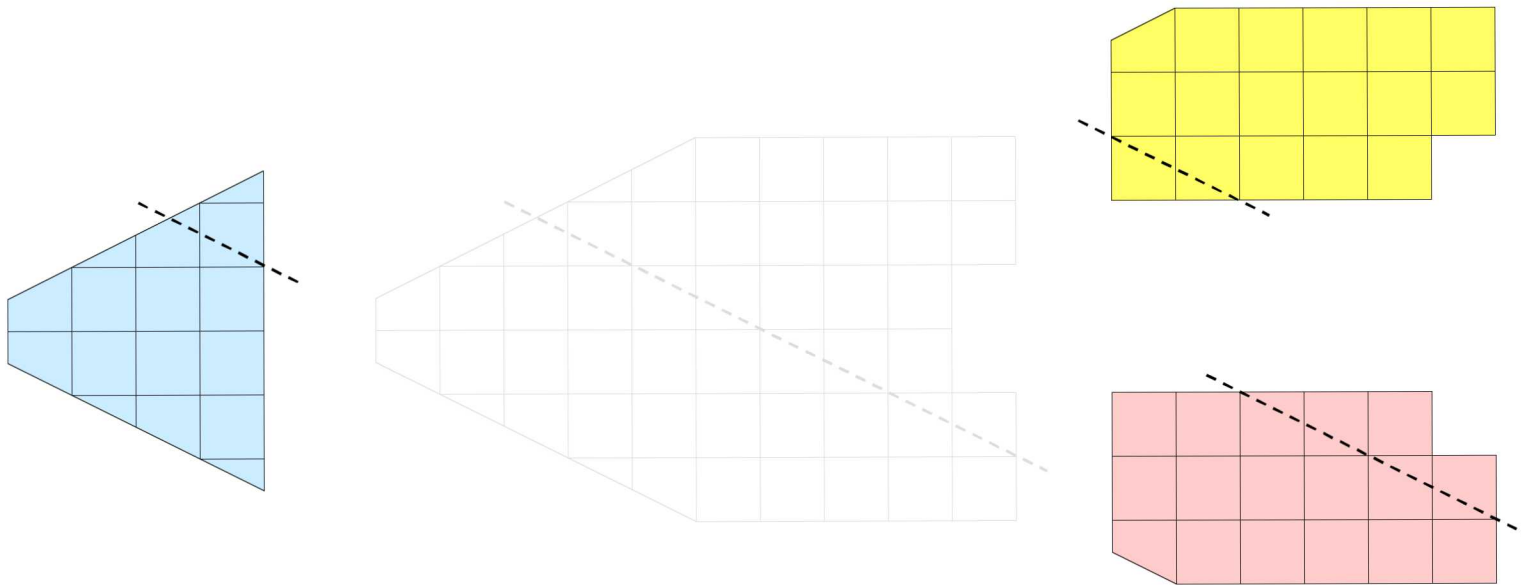
Data Parallel Pipelines

- Duplicate pipelines run independently on different partitions of data.



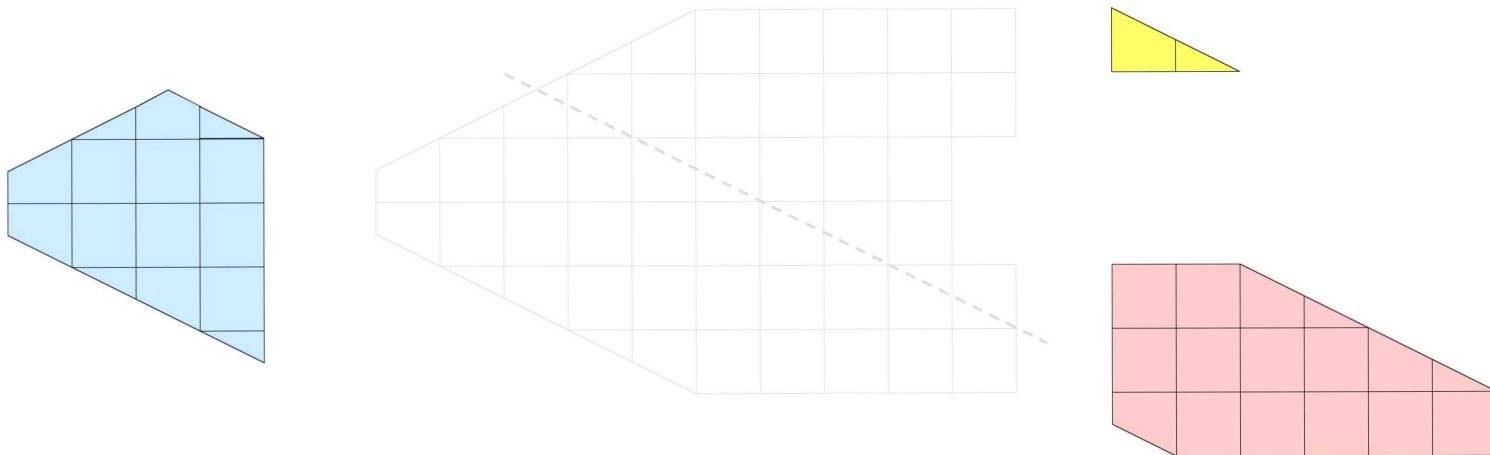
Data Parallel Pipelines

- Many operations will work regardless.
 - Example: Clipping.



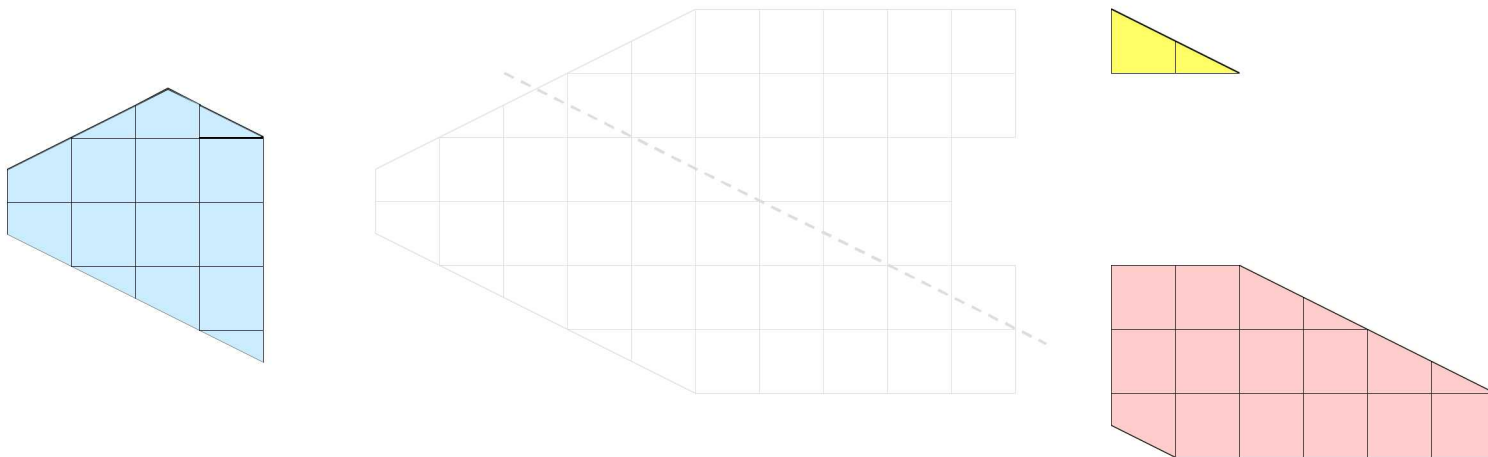
Data Parallel Pipelines

- Many operations will work regardless.
 - Example: Clipping.



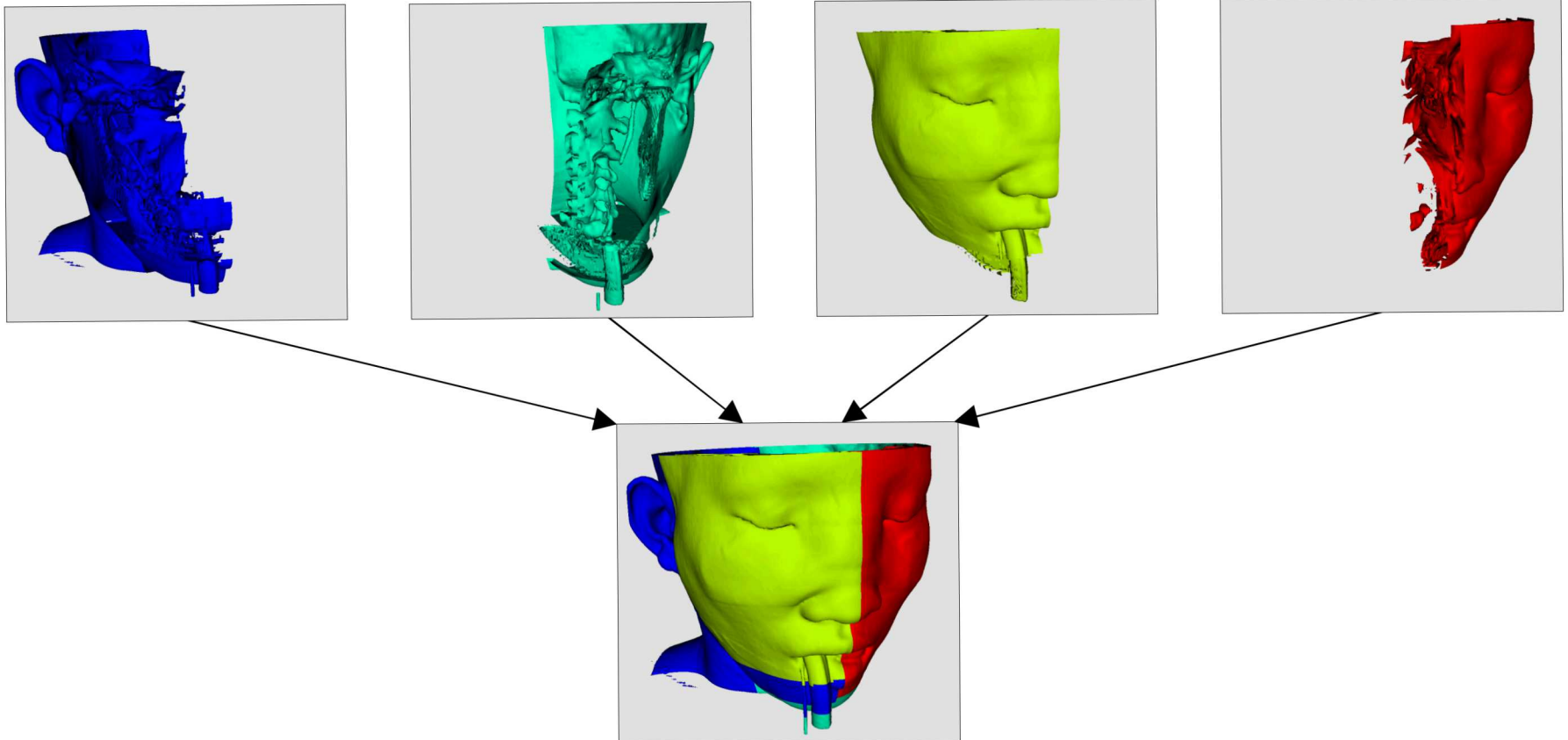
Data Parallel Pipelines

- Many operations will work regardless.
 - Example: Clipping.

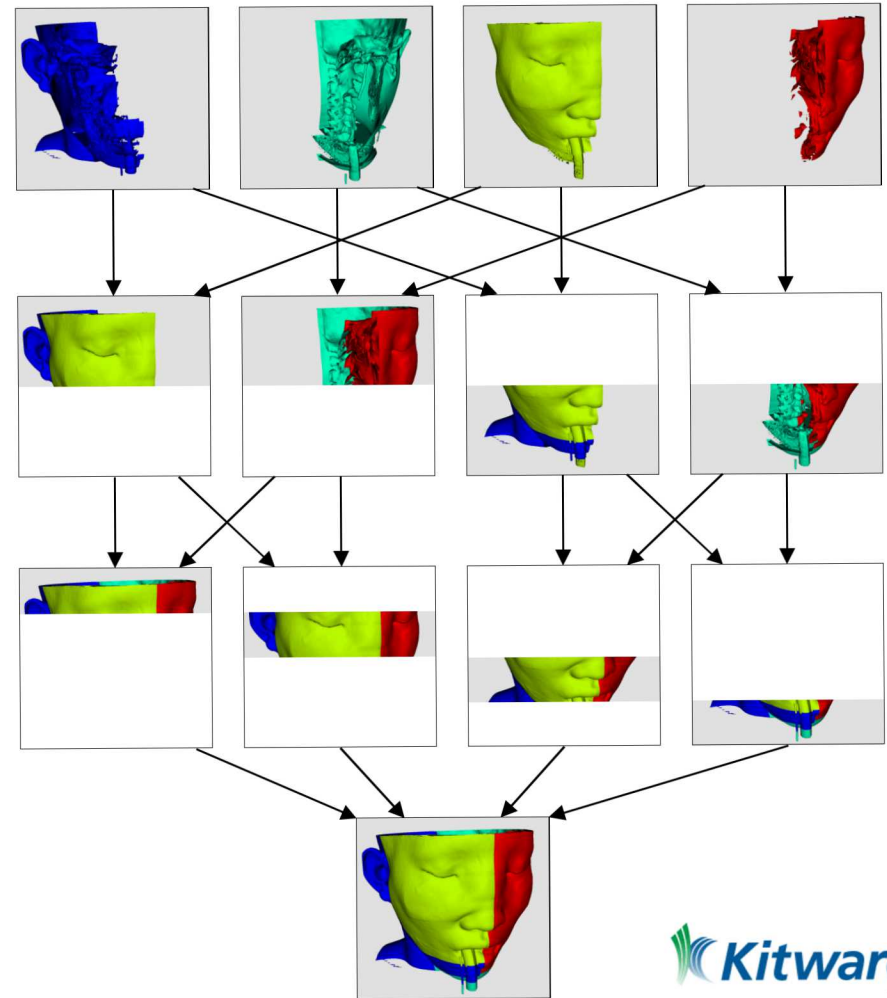
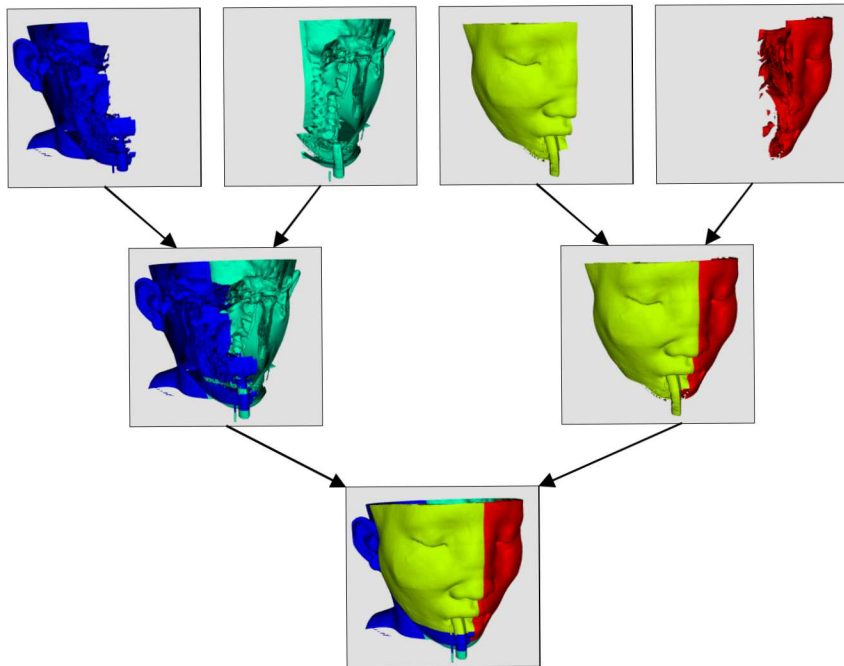


- Will discuss those that don't later

Parallel Rendering



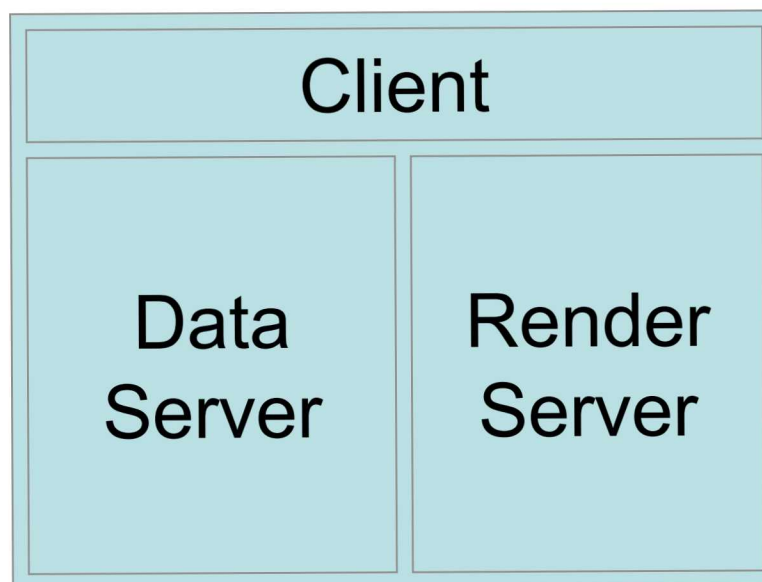
Parallel Rendering



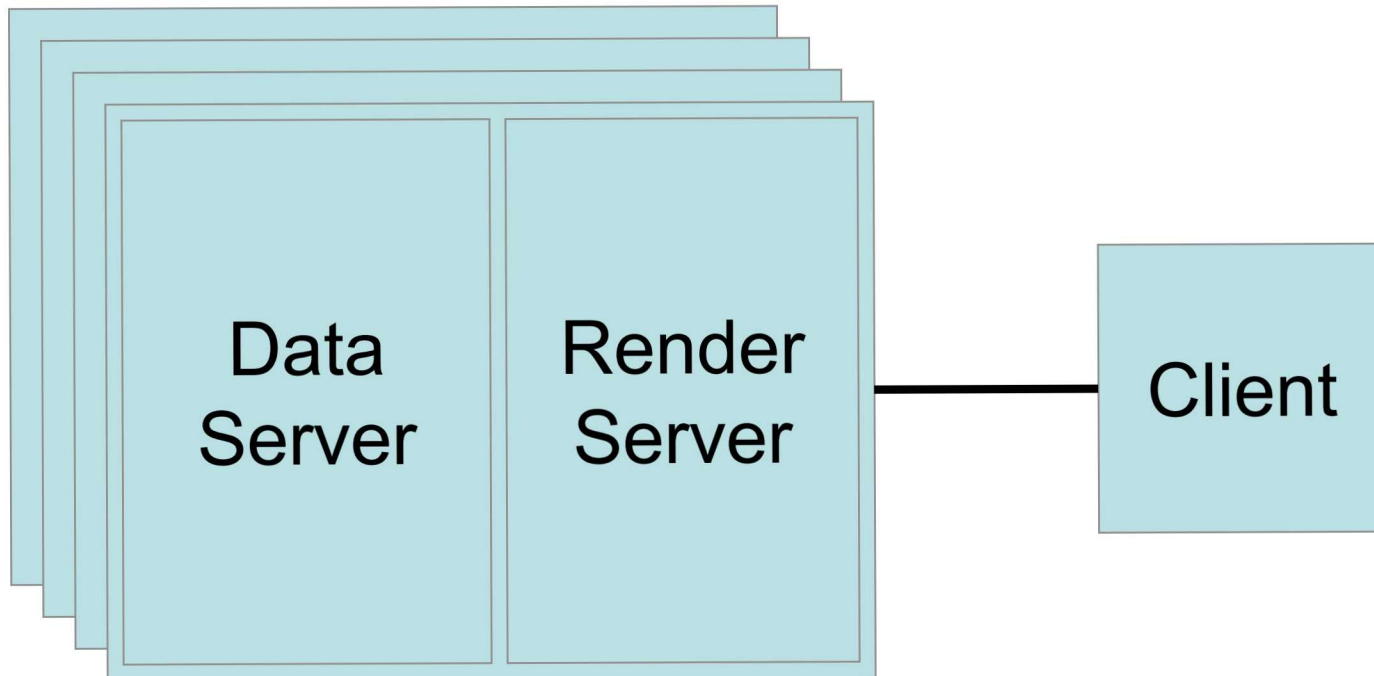
ParaView Architecture

- Three tier
 - Data Server
 - Render Server
 - Client

Standalone

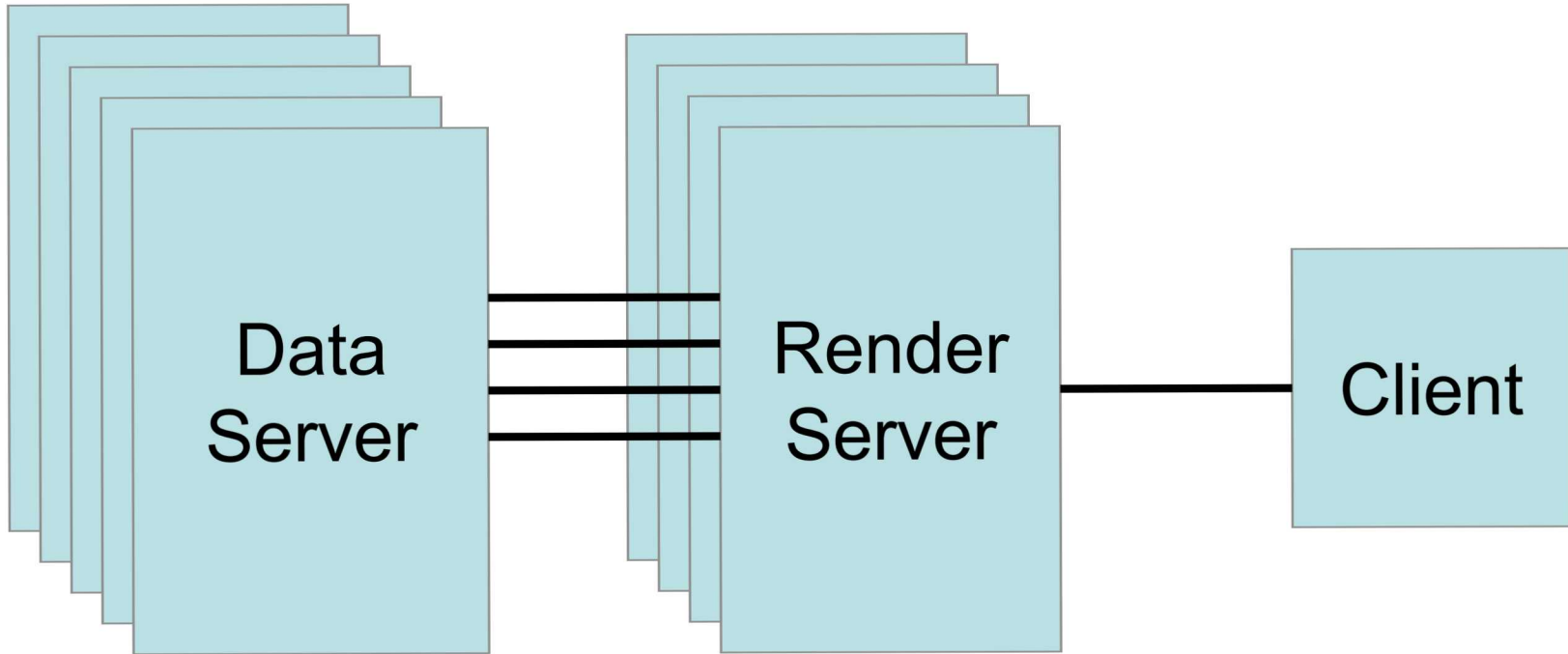


Client-Server





Client-Render Server-Data Server



How to Visualize Large Data?

- Run server on large machine
 - Aggregate memory
 - MPI for communication

Kitware binaries have MPI

but

not tuned to any particular machine

Microsoft MPI (free) not packaged

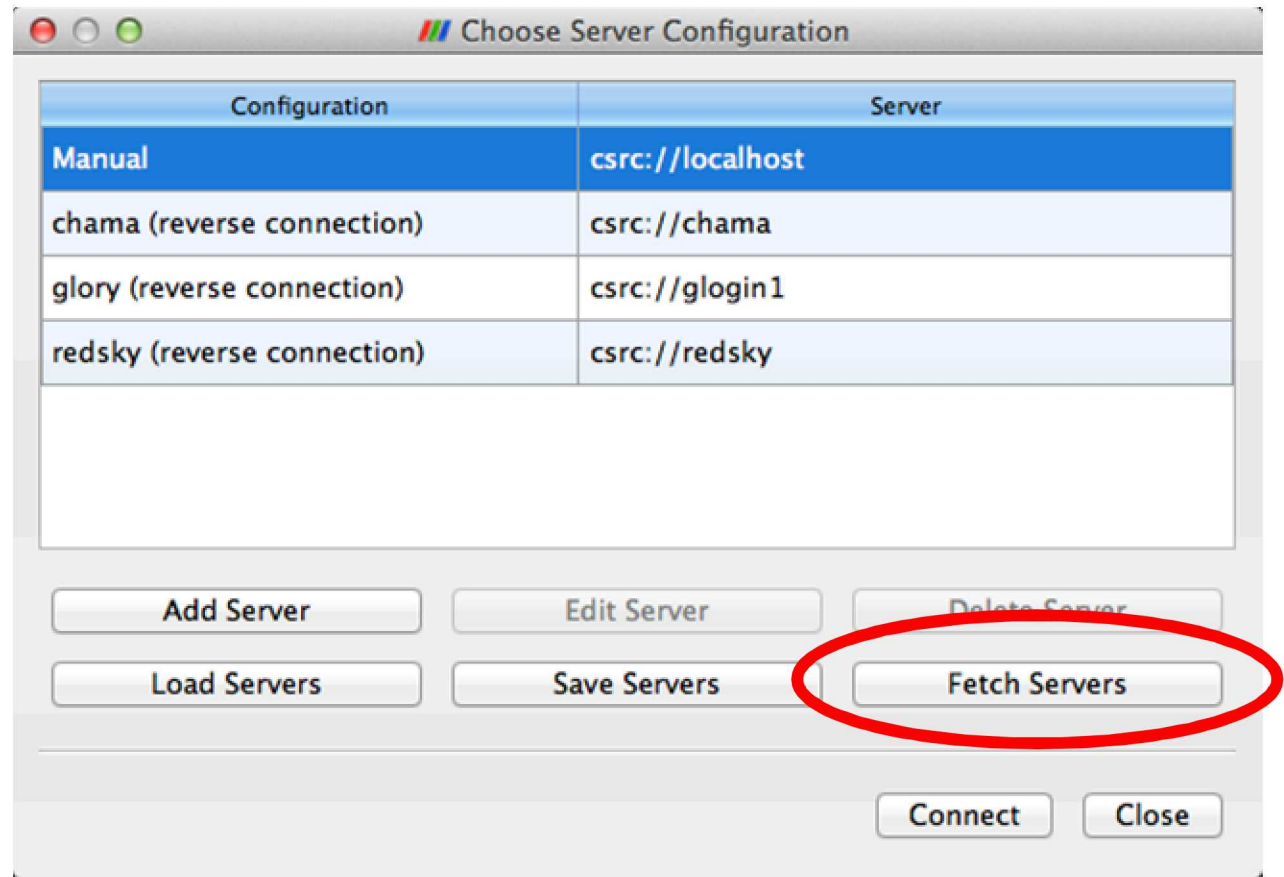
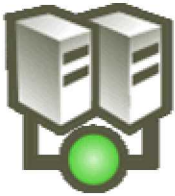


Requirements for Installing ParaView Server



- C++
- CMake (www.cmake.org)
- MPI
- OpenGL (or Mesa3D www.mesa3d.org)
- Python + NumPy + Matplotlib (optional)
- Qt >= 4.7 (optional)
- http://www.paraview.org/Wiki/Setting_up_a_ParaView_Server#Compiling

Connecting to a ParaView Server



http://www.paraview.org/Wiki/Setting_up_a_ParaView_Server#Running_the_Server

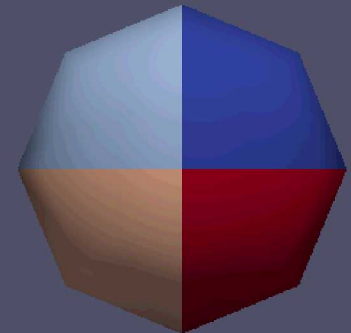
Batch Parallel Processing

- `parasphere.py` paraview script

```
from paraview.simple import *
sphere = Sphere()
rep = Show()
ColorBy(rep, ("POINTS", "vtkProcessId"))
Render()
rep.RescaleTransferFunctionToDataRange(True)
Render()
WriteImage("parasphere.png")
```

- **Run it in parallel**

```
<path to>mpiexec -np 4 \  
  <path to>pvbatch parasphere.py
```



Batch Parallel Processing

- More generally

```
qsub -A <project to charge compute time to> \  
-N <number of nodes> \  
-n <number of processors on each node> \  
mpirun -np <N*n> \  
    pvbatch <arguments for pvbatch> \  
    script.py <arguments for Python script>
```

- Consult system administrators for syntax

Interactive Parallel Processing

- Run pvserver instead, but still in parallel

```
<path to>mpiexec -np 4 \  
<path to>pvserver &
```

```
$ /Applications/paraview.app/Contents/MacOS/mpiexec -np 4 \  
/Applications/paraview.app/Contents/bin/pvserver &  
Waiting for client...
```

```
Connection URL: cs://kaltuel.local:11111
```

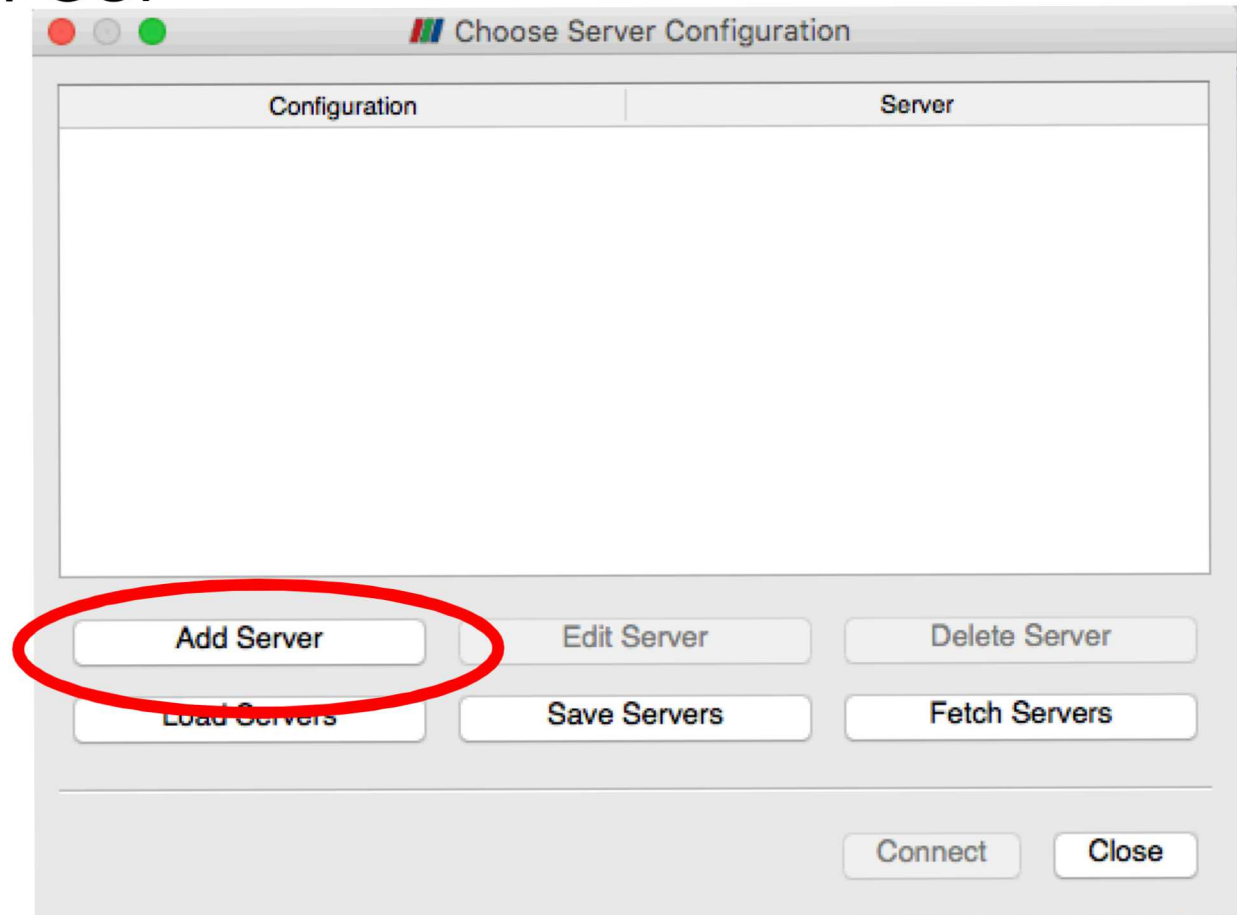
```
Accepting connection(s): kaltuel.local:11111
```

Interactive Parallel Processing

- Start ParaView GUI

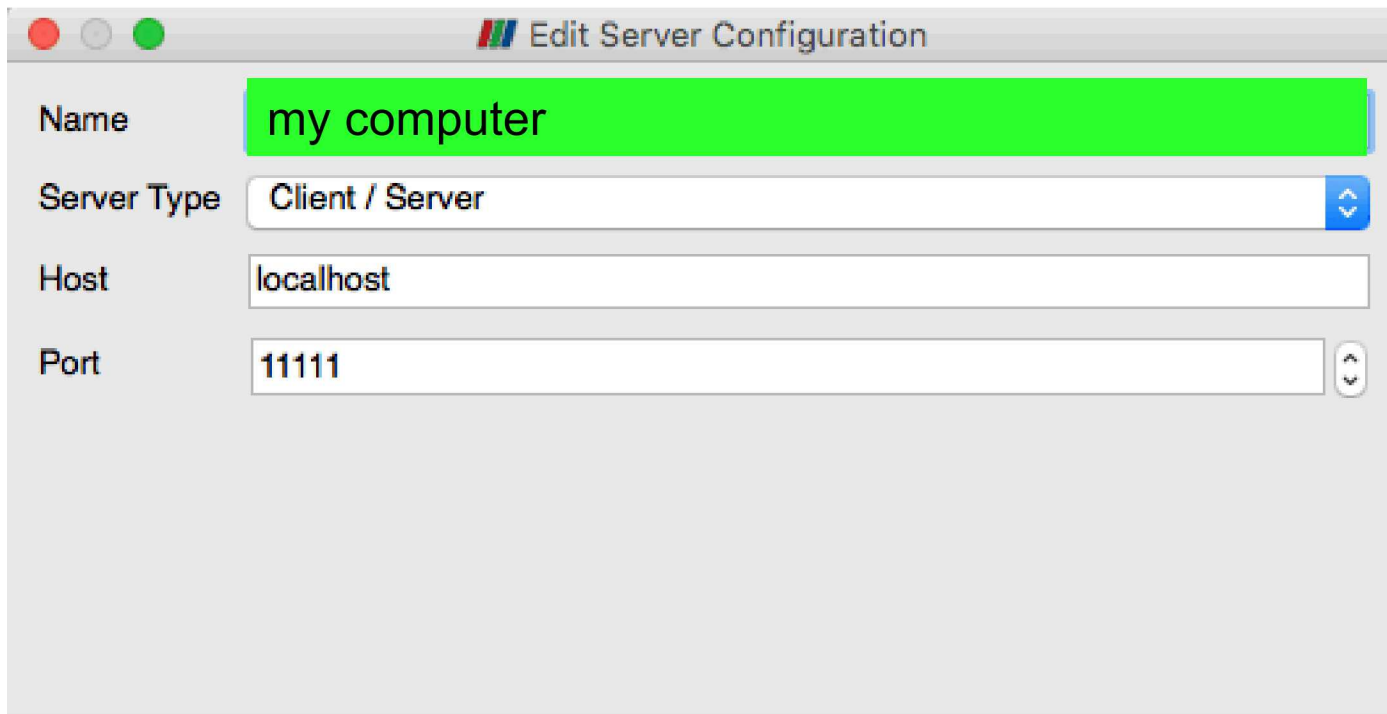
-  Connect

- Add Server



Interactive Parallel Processing

- Specify an TCP/IP path
- Just change nick name to “my computer” in this case

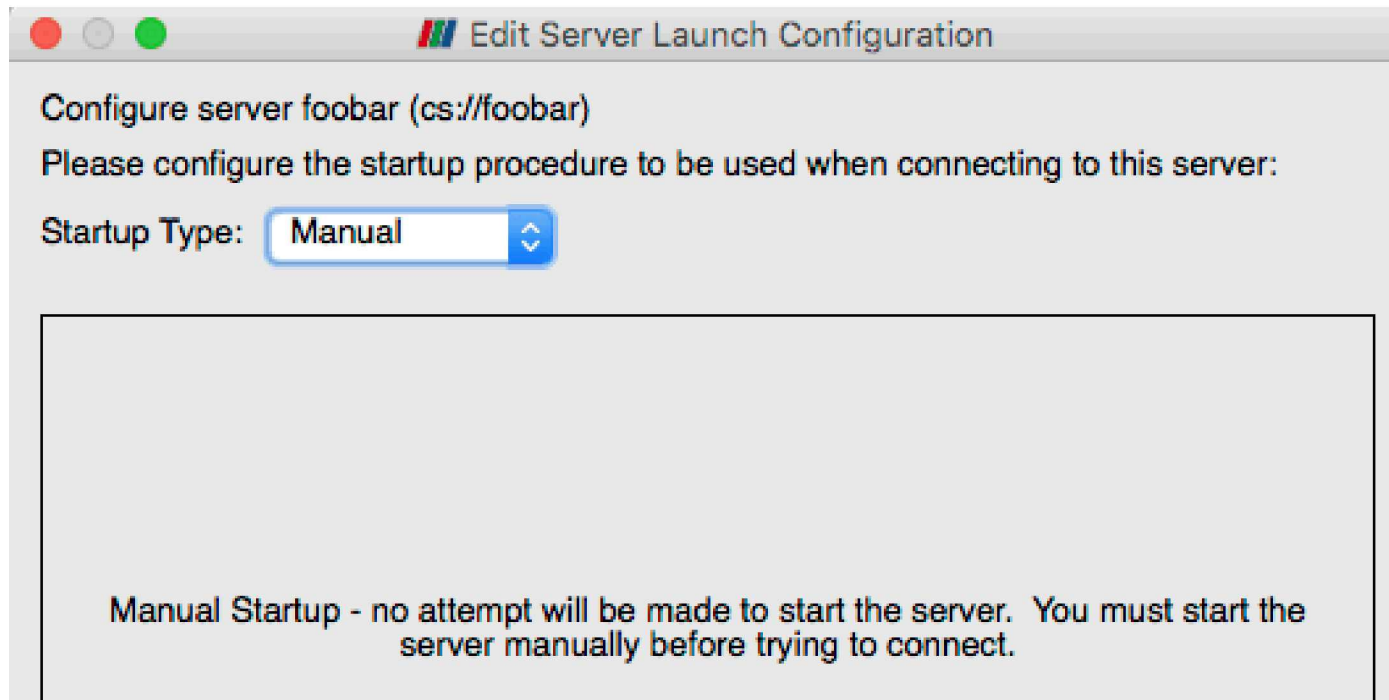


Edit Server Configuration

Name	my computer
Server Type	Client / Server
Host	localhost
Port	11111

Interactive Parallel Processing

- pvserver already waiting, so keep `Manual` connection

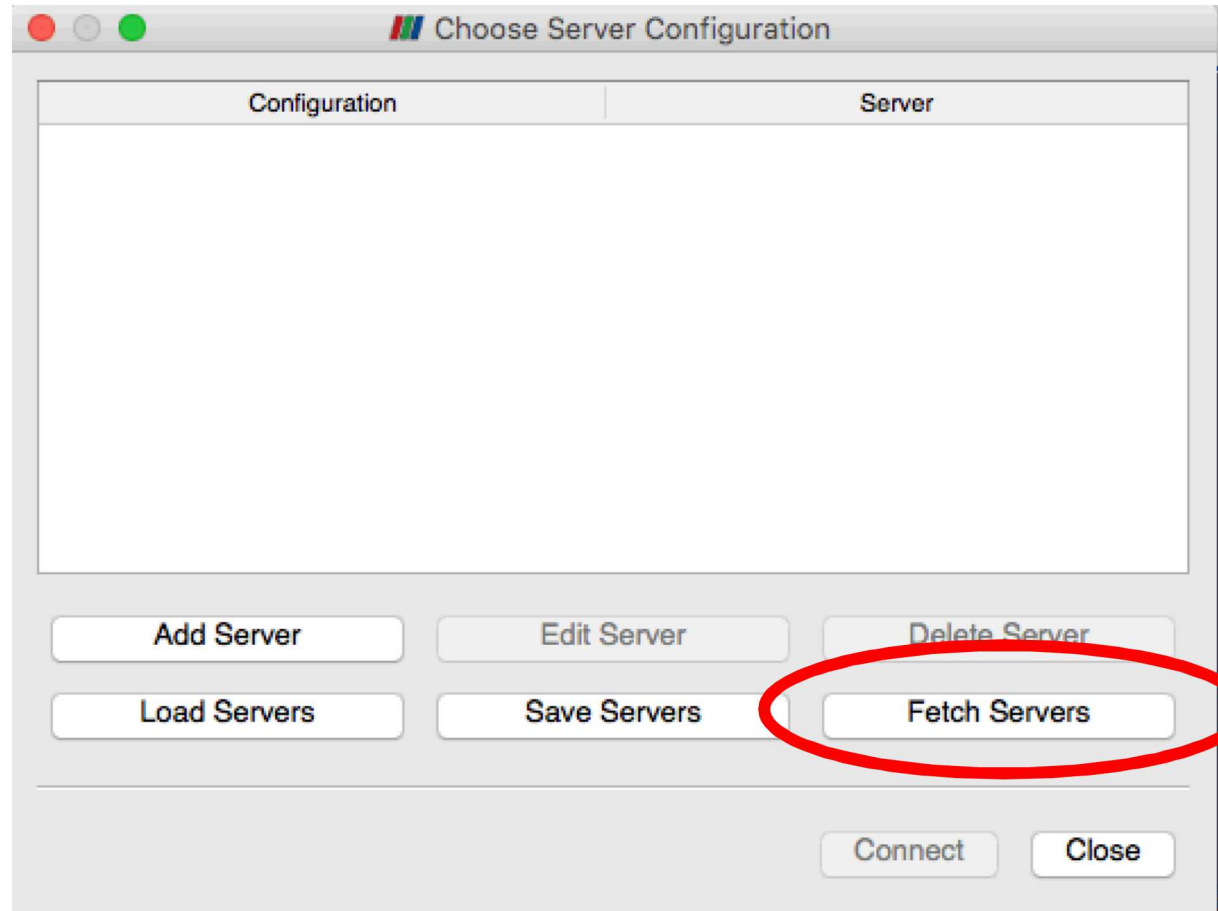


Try a Remote Server Configuration

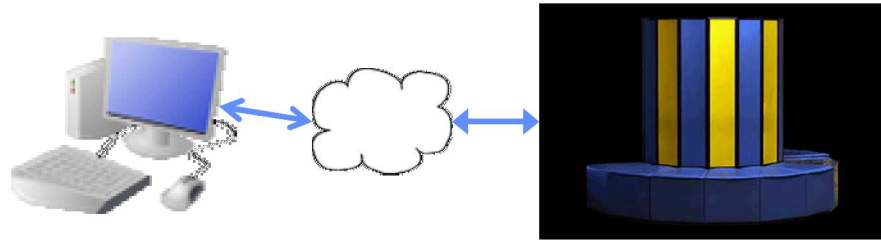
- Start ParaView GUI

-  Connect

- Fetch Servers



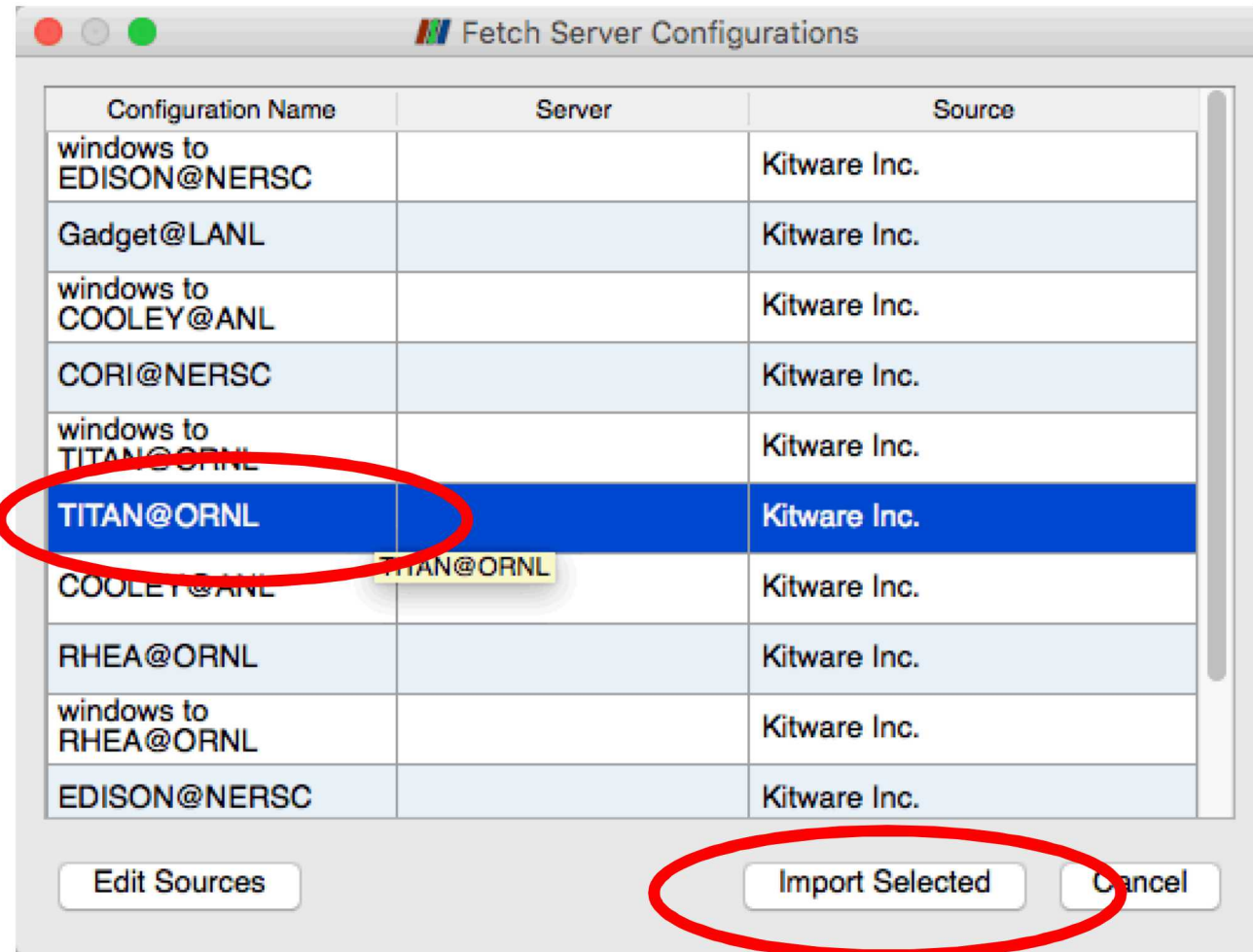
Try a Remote Server Configuration



- Prerequisites on local machine:
 - Windows - need Putty (i.e. ssh) for secure connection
 - Mac - need XQuartz (i.e. X11) for login terminal
 - Linux - typically OK but do know 'which xterm'

Try a Remote Server Configuration

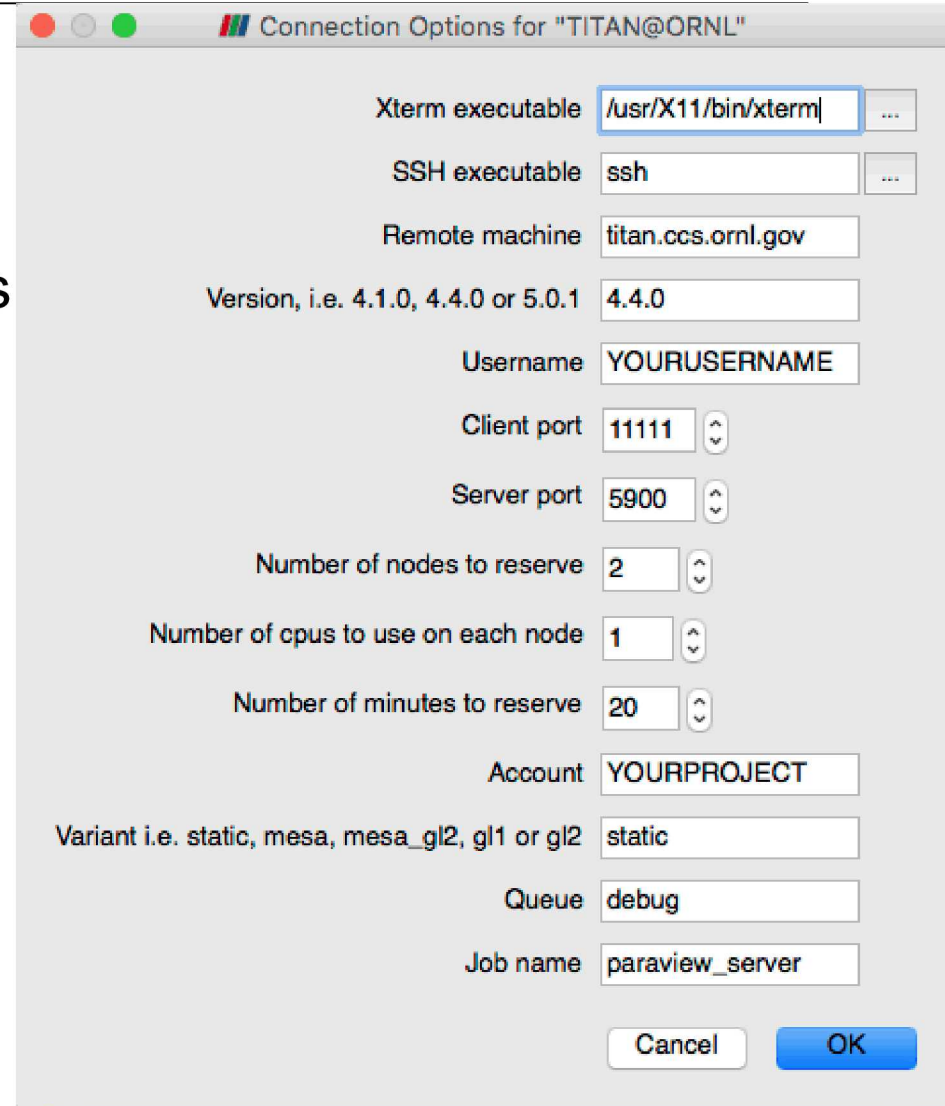
- Select one



- Import it

Try a Remote Server Configuration

- Click `Connect` then
- Configure session
 - Local terminal and ssh exes
 - ParaView version
 - Remote username
 - Number of nodes
 - Amount of time
 - Remote project
 - Site specific options
- Click `OK`



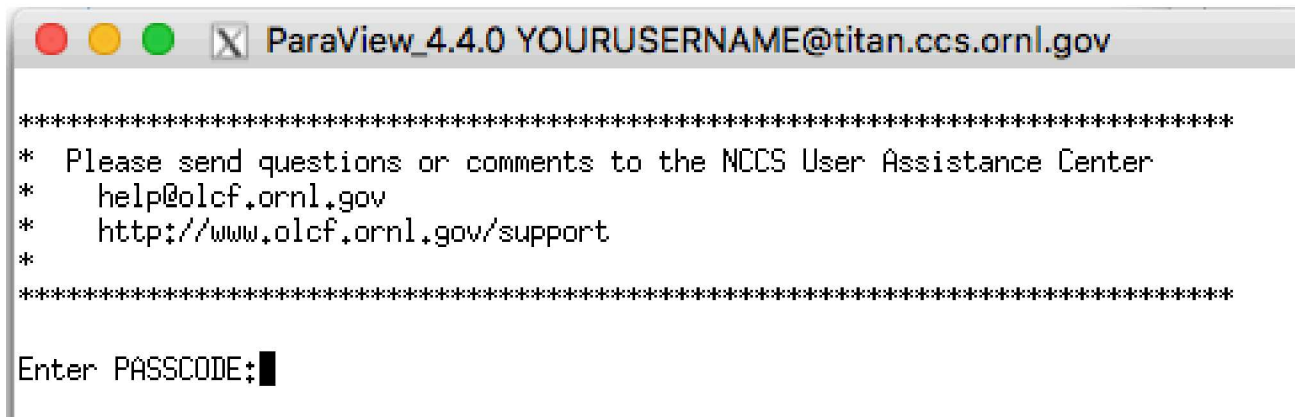
Connection Options for "TITAN@ORNL"

Xterm executable	<input type="text" value="/usr/X11/bin/xterm "/>	...
SSH executable	<input type="text" value="ssh"/>	...
Remote machine	<input type="text" value="titan.ccs.ornl.gov"/>	
Version, i.e. 4.1.0, 4.4.0 or 5.0.1	<input type="text" value="4.4.0"/>	
Username	<input type="text" value="YOURUSERNAME"/>	
Client port	<input type="text" value="11111"/>	⬆ ⬇ ⬆
Server port	<input type="text" value="5900"/>	⬆ ⬇ ⬆
Number of nodes to reserve	<input type="text" value="2"/>	⬆ ⬇ ⬆
Number of cpus to use on each node	<input type="text" value="1"/>	⬆ ⬇ ⬆
Number of minutes to reserve	<input type="text" value="20"/>	⬆ ⬇ ⬆
Account	<input type="text" value="YOURPROJECT"/>	
Variant i.e. static, mesa, mesa_gl2, gl1 or gl2	<input type="text" value="static"/>	
Queue	<input type="text" value="debug"/>	
Job name	<input type="text" value="paraview_server"/>	

Cancel OK

Try a Remote Server Configuration

- Enter credentials



```

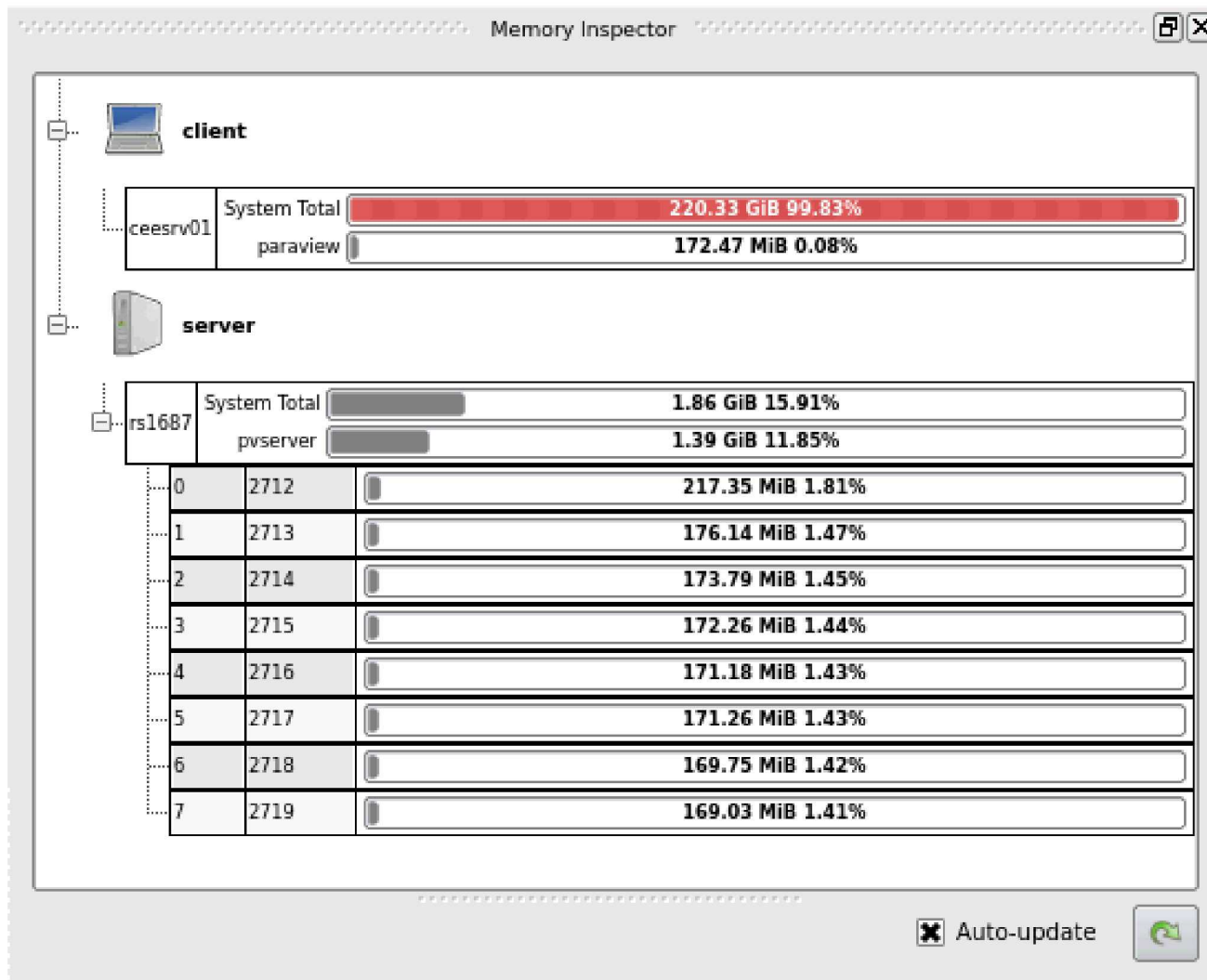
ParaView_4.4.0 YOURUSERNAME@titan.ccs.ornl.gov

*****
* Please send questions or comments to the NCCS User Assistance Center
*   help@olcf.ornl.gov
*   http://www.olcf.ornl.gov/support
*
*****

Enter PASSCODE:█
  
```

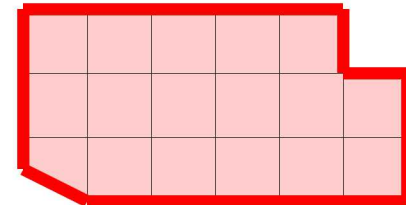
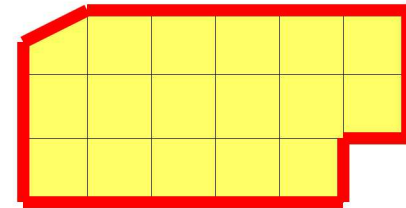
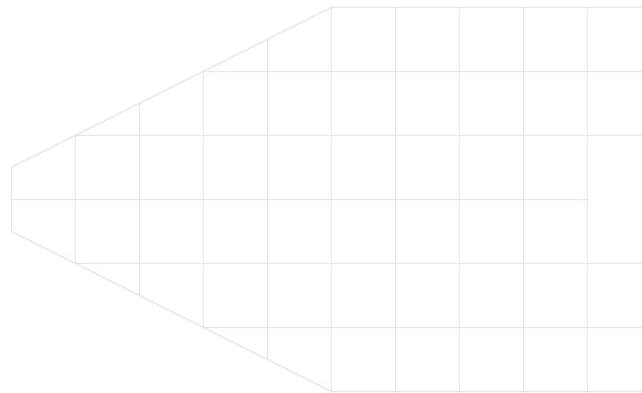
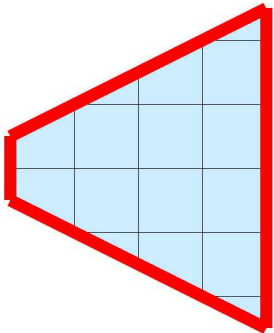
- Once logged in, wait for job to reach top of queue. 3D View will return then.
- File -> Open, is remote system's file system, otherwise about the same as normal

Memory Inspector



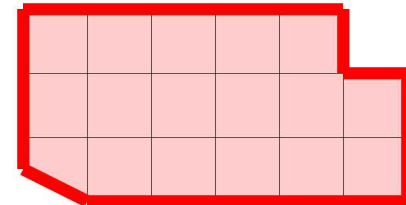
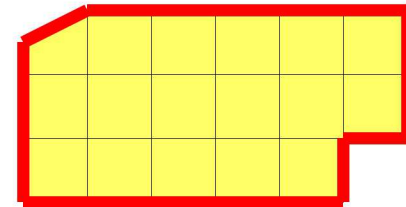
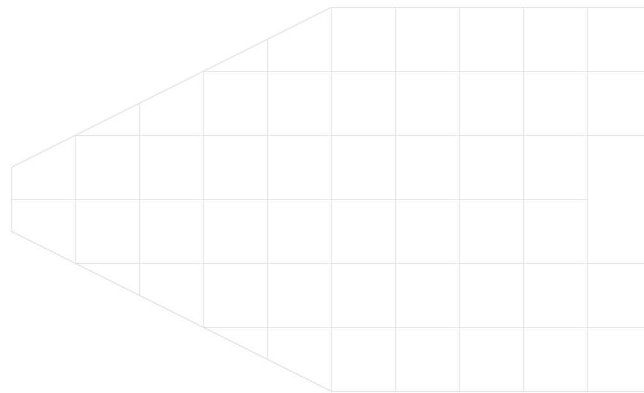
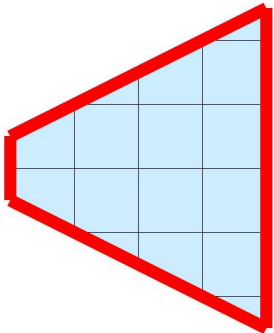
Data Parallel Pipelines

- Some operations will have problems.
 - Example: External Faces



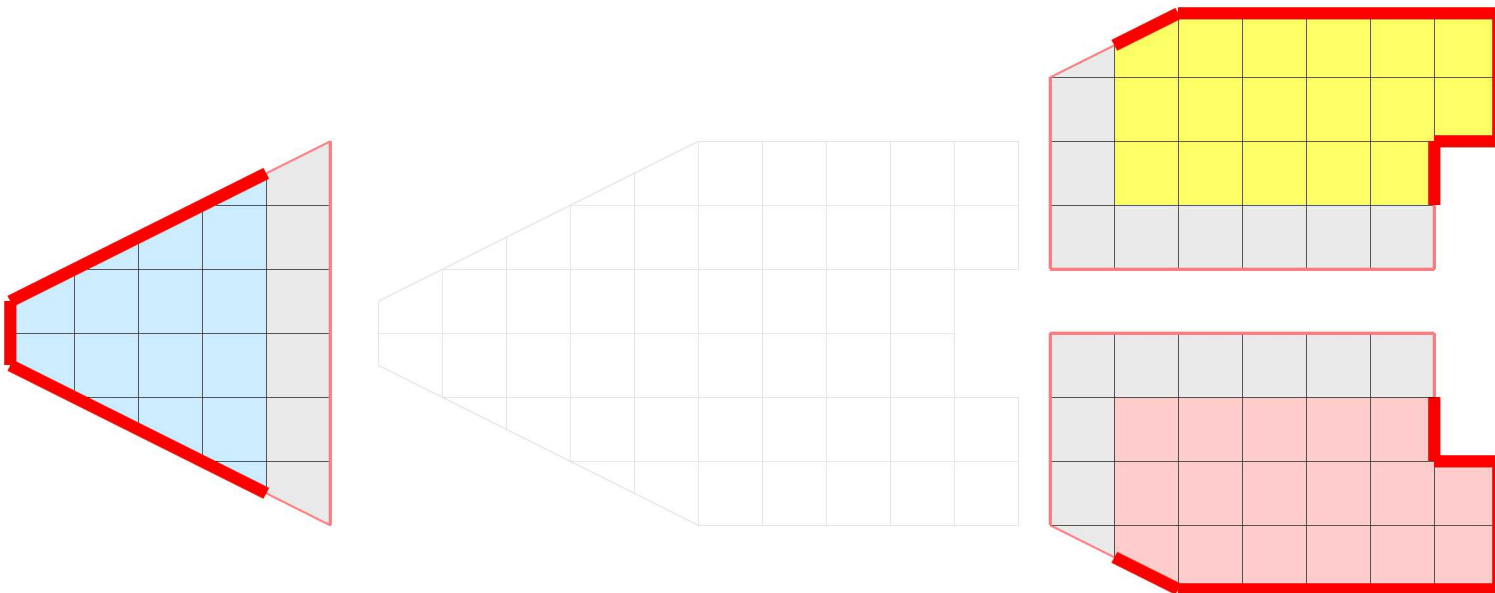
Advanced Data Parallel Pipelines

- Some operations will have problems.
 - Example: External Faces



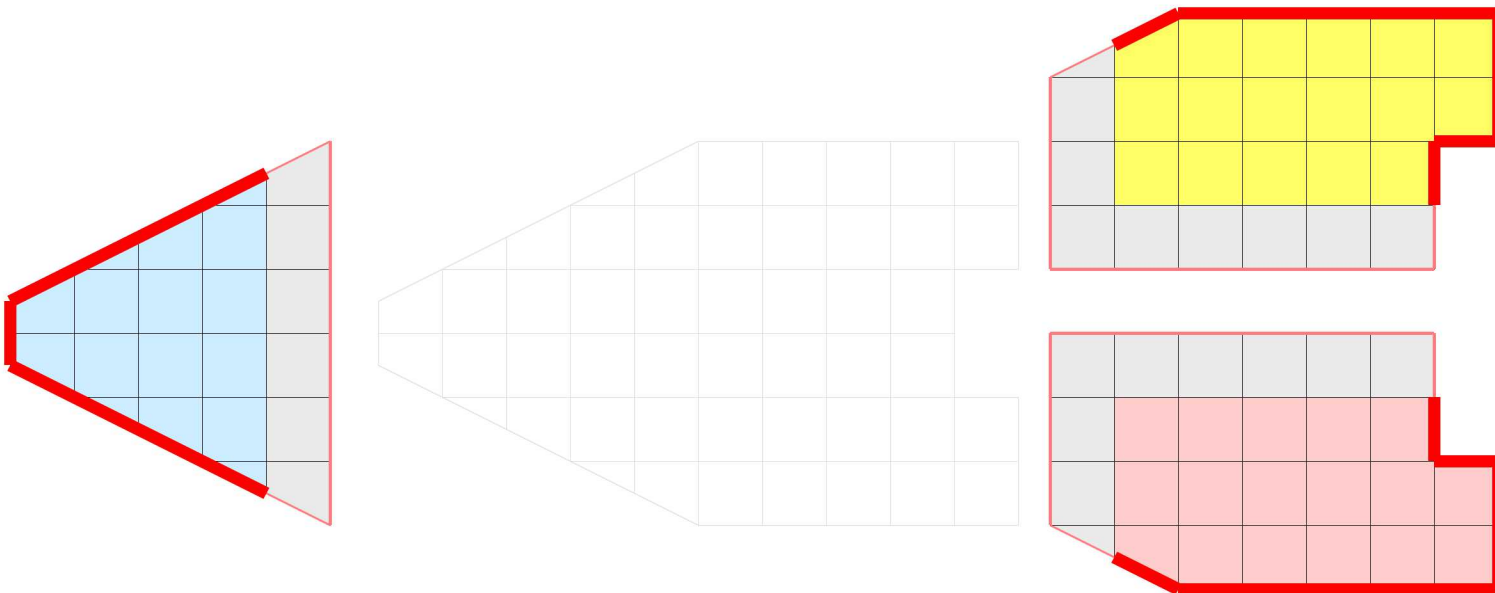
Advanced Data Parallel Pipelines

- Ghost cells can solve most of these problems.



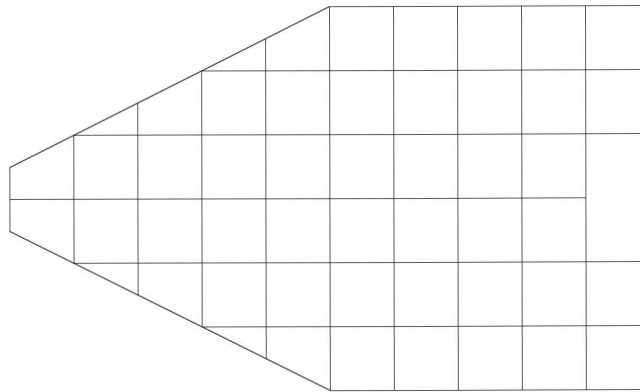
Advanced Data Parallel Pipelines

- Ghost cells can solve most of these problems.



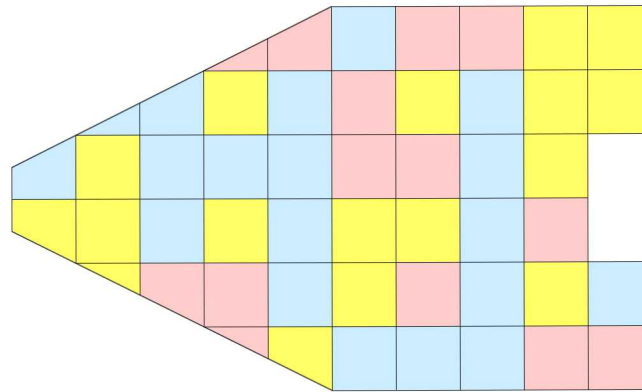
Data Partitioning

- Partitions should be load balanced and spatially coherent.



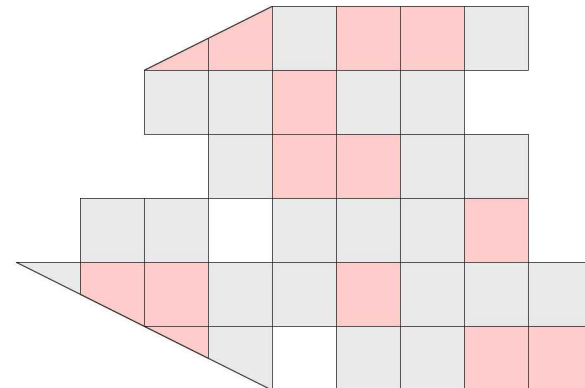
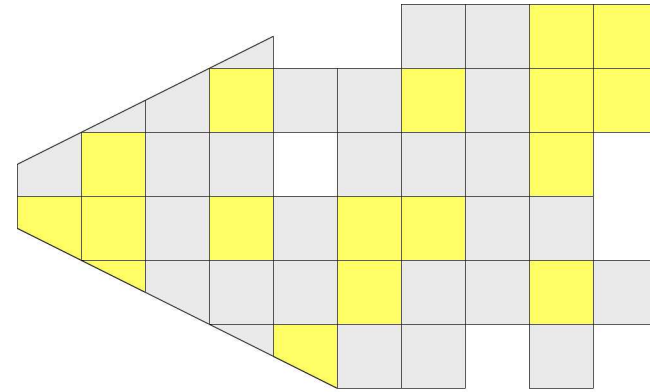
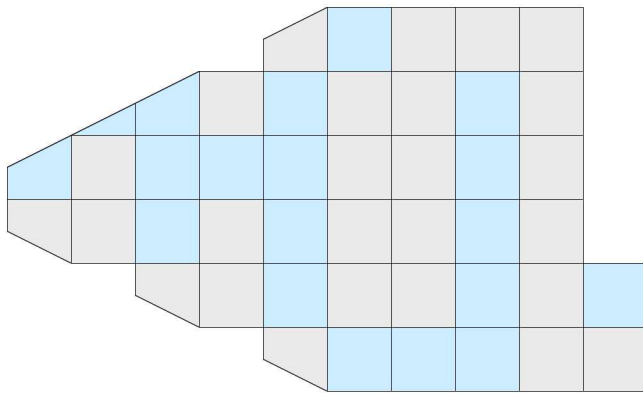
Data Partitioning

- Partitions should be load balanced and spatially coherent.



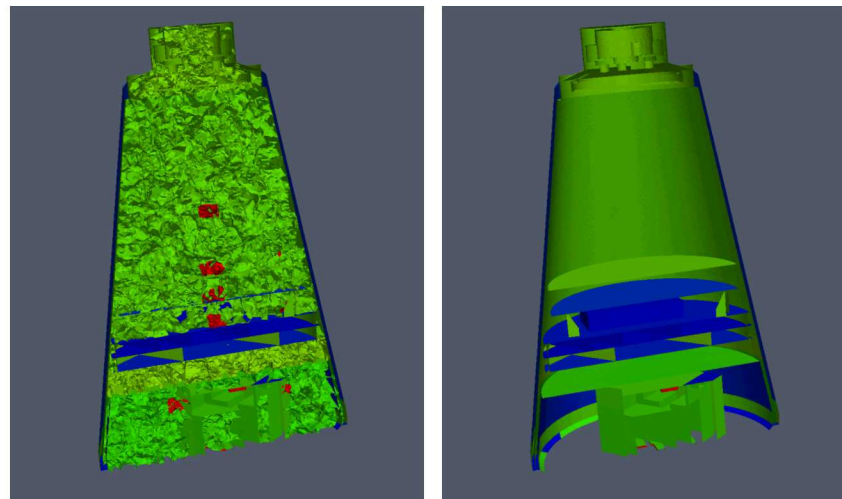
Data Partitioning

- Partitions should be load balanced and spatially coherent.



Load Balancing/Ghost Cells

- Automatic for Structured Meshes.
- Partitioning/ghost cells for unstructured is “manual.”
- Use Ghost Level Generator to create
- Legacy option: D3. Also repartitions



Job Size Rules of Thumb

- Structured Data
 - Try for max 20 M cell/processor.
 - Shoot for 5 – 10 M cell/processor.
- Unstructured Data
 - Try for max 1 M cell/processor.
 - Shoot for 250 – 500 K cell/processor.

Avoiding Data Explosion

- Pipeline may cause data to be copied, created, converted.
- This advice **only for dealing with very large amounts of data.**
 - Remaining available memory is low.

Topology Changing, No Reduction

- Append Datasets
- Append Geometry
- Clean
- Clean to Grid
- Connectivity
- D3
- Delaunay 2D/3D
- Extract Edges
- Linear Extrusion
- Loop Subdivision
- Reflect
- Rotational Extrusion
- Shrink
- Smooth
- Subdivide
- Tessellate
- Tetrahedralize
- Triangle Strips
- Triangulate

Topology Changing, Moderate Reduction



- Clip 

- Decimate

- Extract Cells by
Region

- Extract Selec 

- Quadric Clustering




- Threshold 

Similar: Extract Subset 





Topology Changing, Dimension Reduction










- Cell Centers
- Contour A circular icon with concentric rings in shades of green and blue, representing a contour plot.
- Extract CTH Parts
- Extract Surface
- Feature Edges
- Mask Points
- Outline (curvilinear)
- Slice A 3D wireframe cube with a green semi-transparent slice through it, representing a slice operation.
- Stream Tracer An icon showing a green sphere with three wavy lines extending from it, representing a stream tracer.

Adds Field Data

- Block Scalars
- Calculator 
- Cell Data to Point Data
- Compute Derivatives
- Curvature
- Elevation
- Generate Ids
- Gen. Surface Normals
- Gradient
- Level Scalars
- Median
- Mesh Quality
- Octree Depth Limit
- Octree Depth Scalars
- Point Data to Cell Data
- Process Id Scalars
- Random Vectors
- Resample with dataset
- Surface Flow
- Surface Vectors
- Texture Map to...
- Transform
- Warp (scalar)
- Warp (vector) 

Total Shallow Copy or Output Independent of Input




- Annotate Time
- Append Attributes
- Extract Block
- Extract Datasets
- Extract Level 
- Glyph 
- Group Datasets 
- Histogram 
- Integrate Variables
- Normal Glyphs
- Outline
- Outline Corners
- Plot Global Variables Over Time
- Plot Over Line 
- Plot Selection Over Time 
- Probe Location 
- Temporal Shift Scale
- Temporal Snap-to-Time-Steps
- Temporal Statistics



Special Cases

- Temporal Filters
 - Temporal Interpolator
 - Particle Tracer
 - Temporal Cache
- Programmable Filter {...}

Culling Data

- Reduce dimensionality early. 
 - Contour and slice “see” inside volumes.
- Prefer data reduction over extraction.
 - Slice instead of Clip.
 - Contour instead of Threshold.
- Only extract when reducing an order of magnitude or more.
 - Can still run into troubles.

Culling Data

- Experiment with subsampled data.
 - Extract Subset
- Use caution.
 - Subsampled data may be lacking.
 - Use full data to draw final conclusions.

Resample To Image

- Structured Grid representations are compact
- Efficiently copies values onto regular grid
- Can downsample at same time

Resample To Image

DIY2 handles
block data
transfer

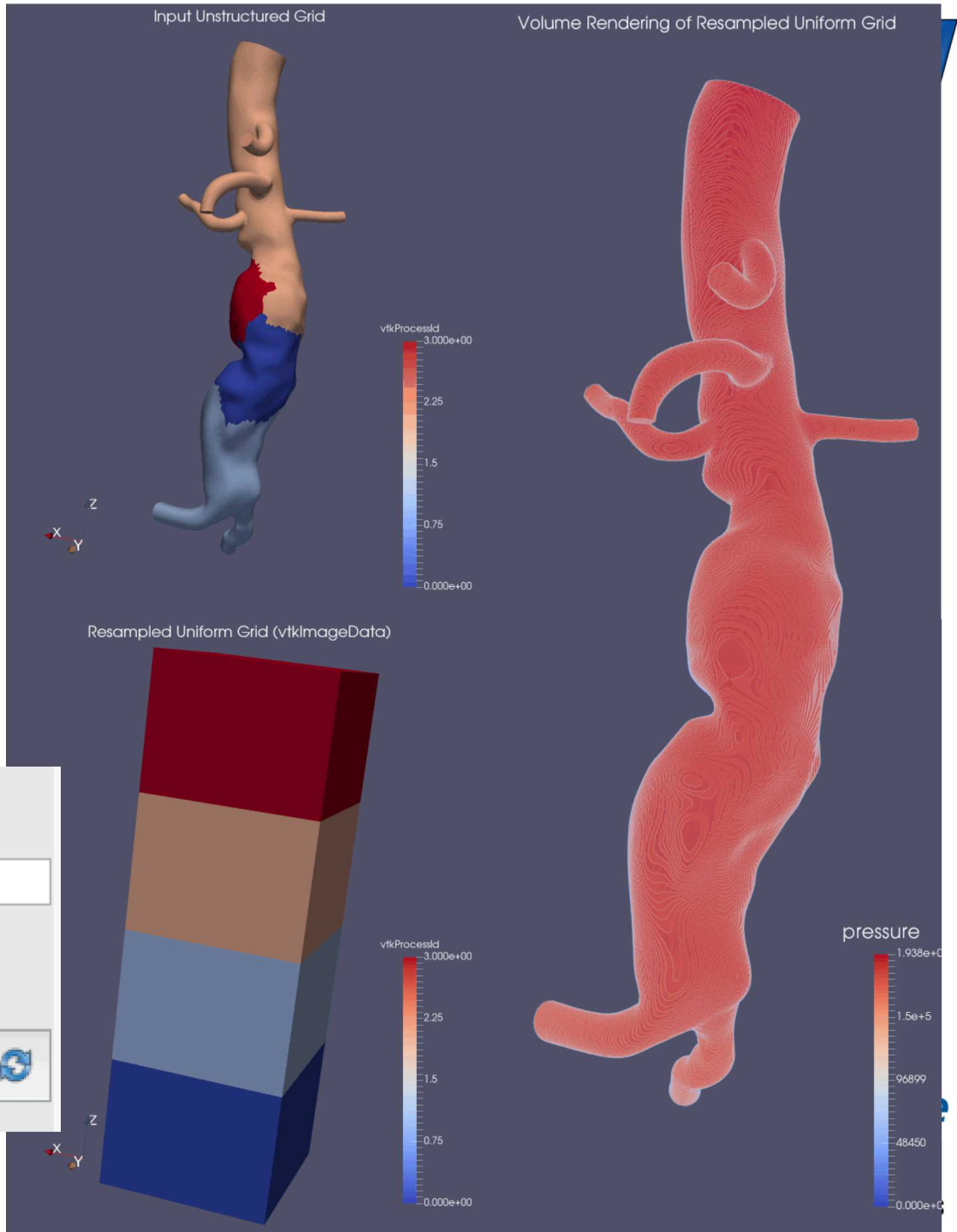
Use Input Bounds

Sampling Dimensions

256	256	1024
-----	-----	------

Sampling Bounds

-6.64	1.04
-2.01	6.25
-9.94	13.42

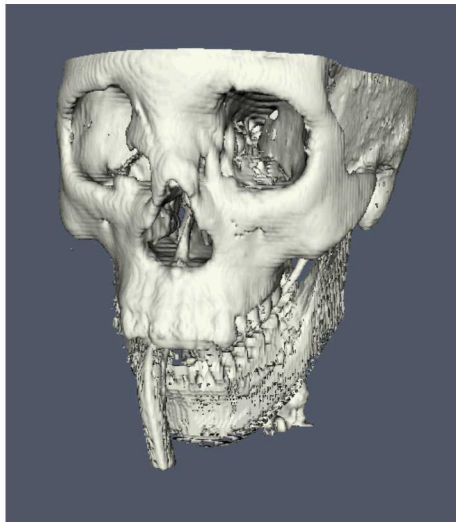


Rendering Modes

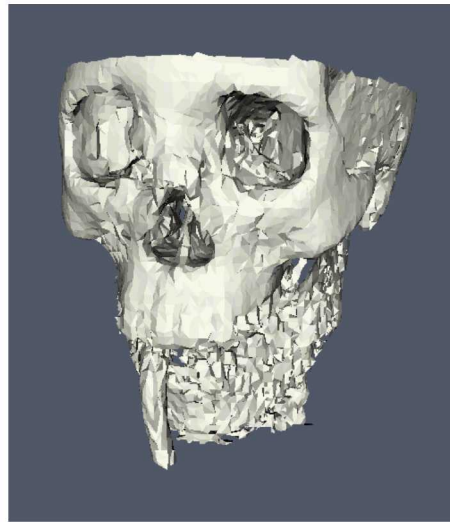
- Still Render
 - Full detail render.
- Interactive Render
 - Sacrifices detail for speed.
 - Provides quick rendering rate.
 - Used when interacting with 3D view.

Level of Detail (LOD)

- Geometric decimation
- Used only with Interactive Render



Original Data



Default Decimation



Maximum Decimation

3D Rendering Parameters

Edit → Settings, Render View

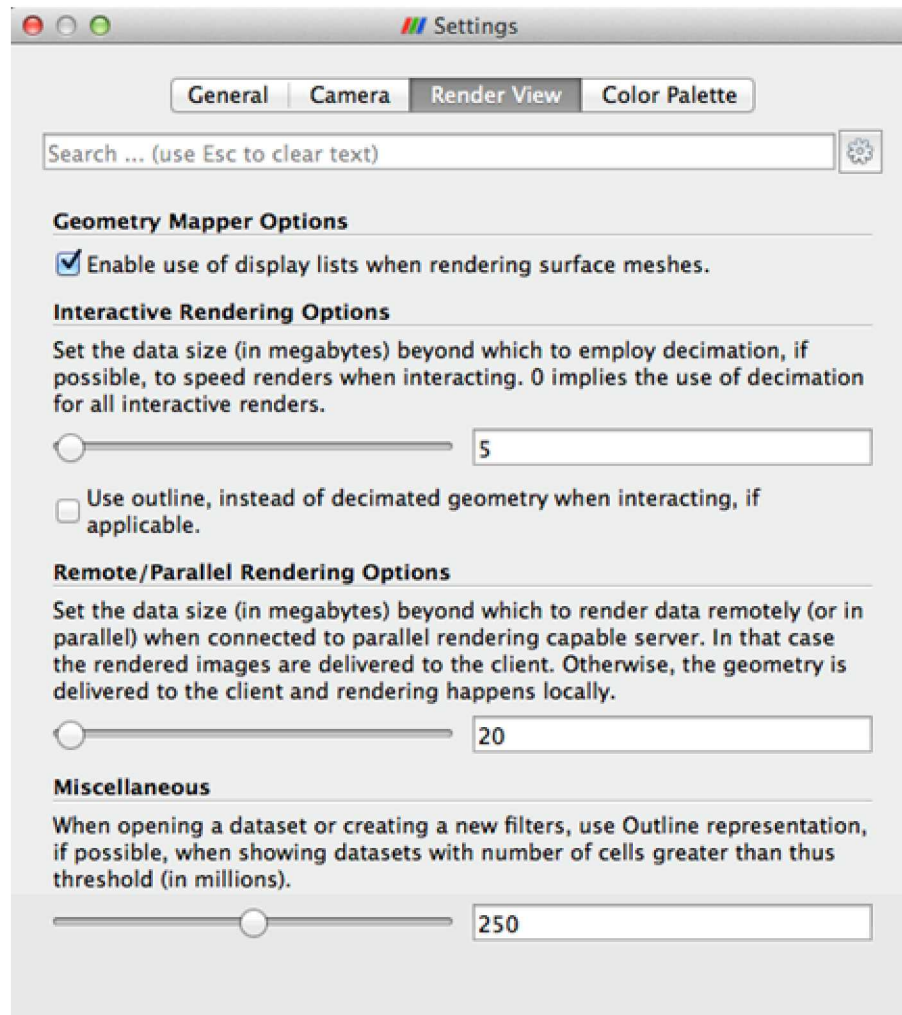
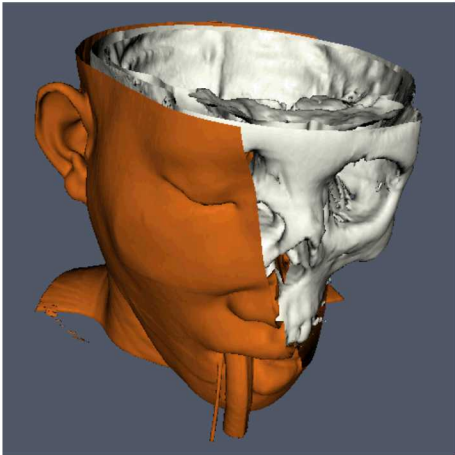
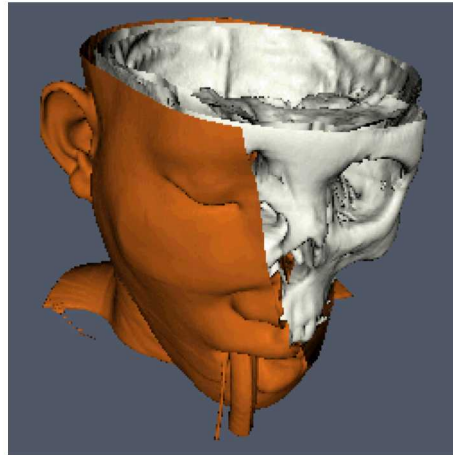


Image Size LOD

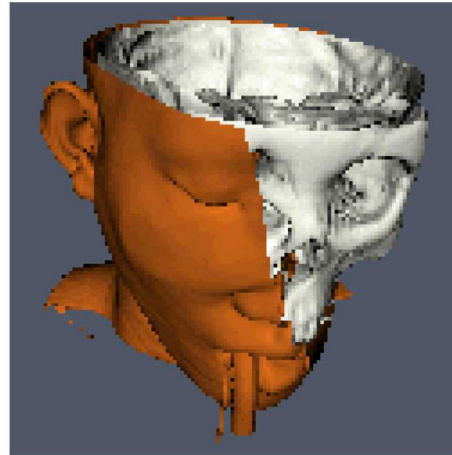
- ParaView's parallel rendering overhead proportional to image size.
- Can use smaller images for interactive rendering.



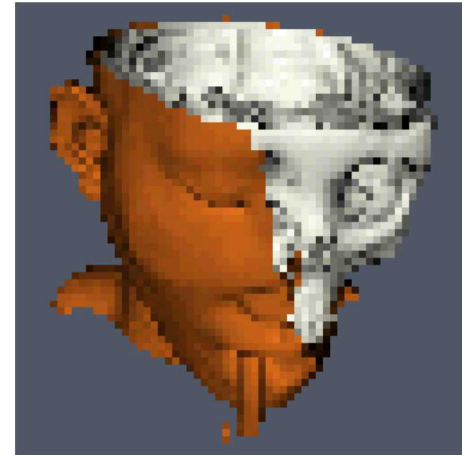
Original Data



Subsample Rate: 2 pixels



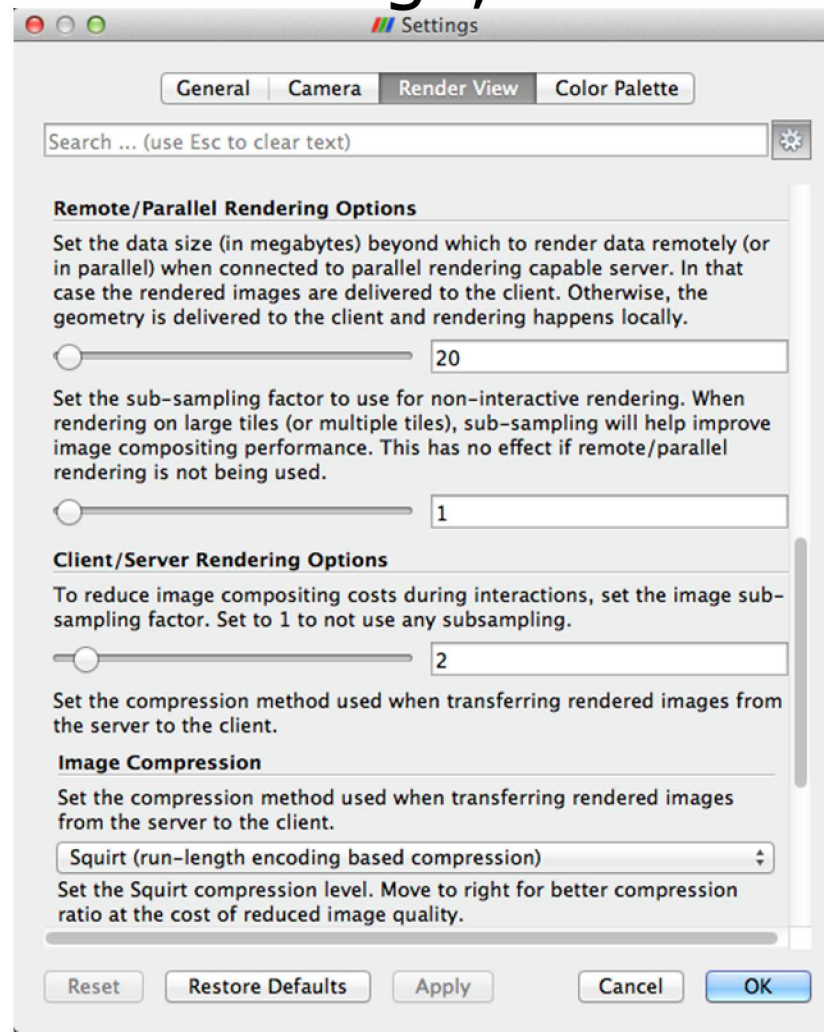
Subsample Rate: 4 pixels



Subsample Rate: 8 pixels

Parallel Rendering Parameters

Edit → Settings, Render View



Parameters for Large Data

- Use display lists for GPU, off for CPU.
- Try outline instead of decimated geometry.
 - Also try lowering decimation factor.
- Avoid shipping large data back to client.
- Turn on subsampling.
- Image Compression on.

A decorative graphic in the top-left corner featuring a white star on a red background, with a blue and white striped pattern extending from the top-left towards the center.

Parameters for Low Bandwidth

- Try larger subsampling rates.
- Try Zlib compression and fewer bits.

Parameters for High Latency

- Turn up Remote Render Threshold.
 - Allow more data to go to client.
- Play with the LOD Threshold and LOD Resolution to control geometry sent to client.
- Turn on Use outline for LOD rendering if decimated geometry too big for client.
- Try turning on interactive render delay.



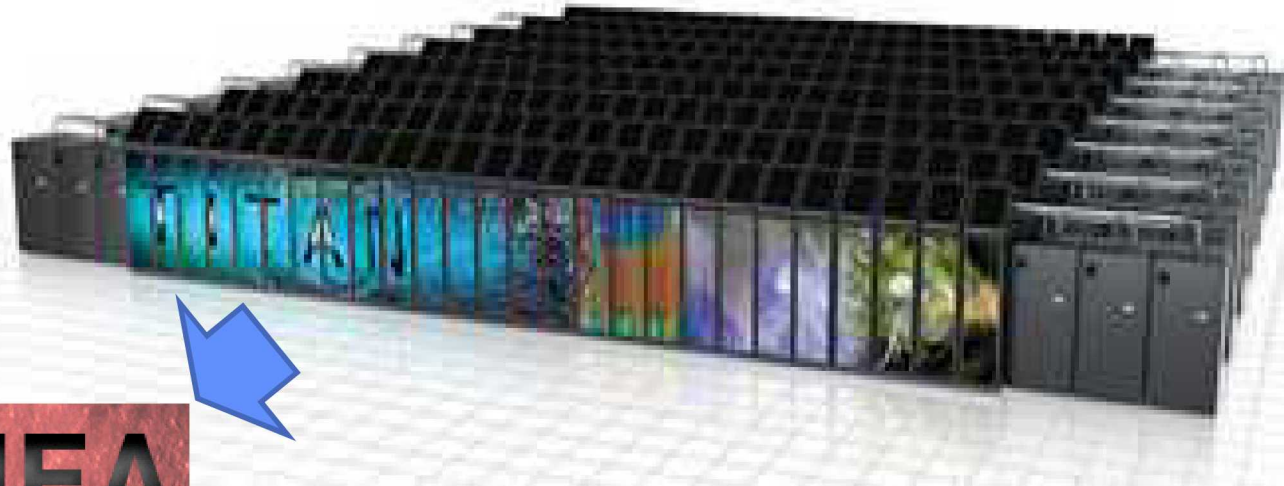
ParaView
Catalyst

Why *In Situ*?

System Parameter	2011	“2018”		Factor Change
System peak	2 PF	1 EF		500
Power	6 MW	≤20 MW		3
System Memory	0.3 PB	32-64 PB		33
Node Performance	0.125 Tf/s	1 TF	10 TF	8-80
Node Concurrency	12	1,000	10,000	83-830
Network BW	1.5 GB/s	100 GB/s	1,000 GB/s	66-660
System Size (nodes)	18,700	1M	100k	50
Total Concurrency	225 K	10 B	100 B	40k-400k
Storage Capacity	15 PB	300-1,000 PB		20-67
I/O BW	0.2 TB/s	20-60 TB/s		100-300

Why *In Situ*?

Need a supercomputer to analyze results from a hero run



RHEA

2 orders of magnitude difference
between each level

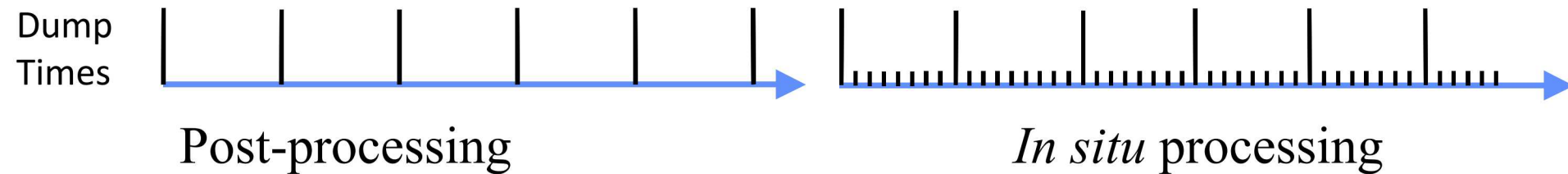
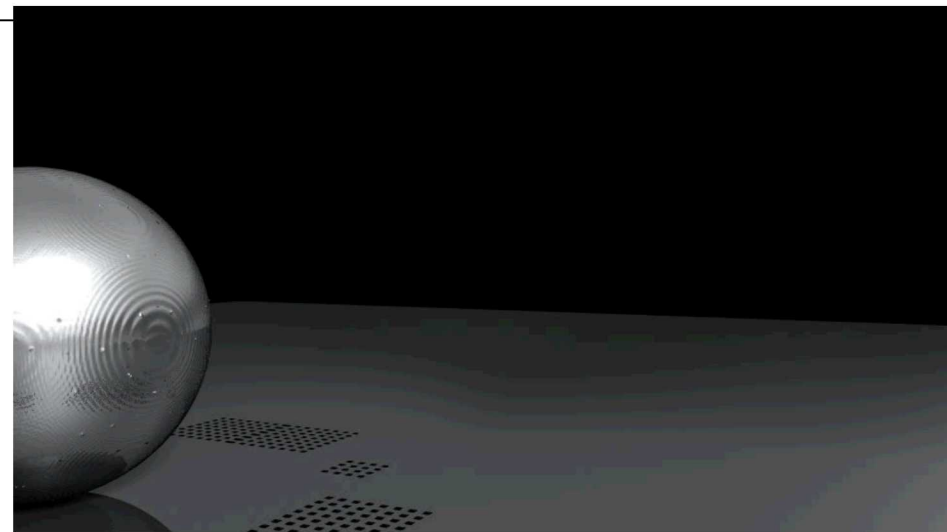
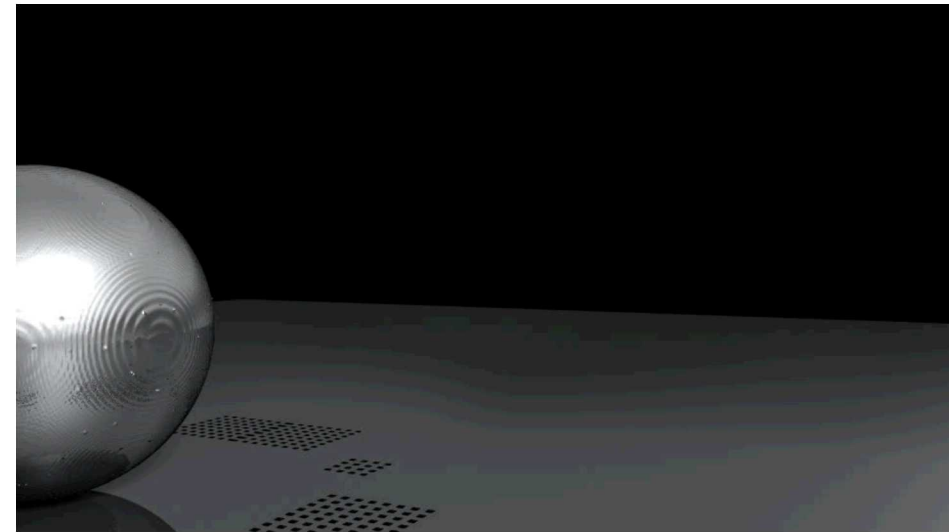


Reduced File IO Costs

Time of Processing	Type of File	Size per File	Size per 1000 time steps	Time per File to Write at Simulation
Post	Restart	1,300 MB	1,300,000 MB	1-20 seconds
Post	Ensign Dump	200 MB	200,000 MB	> 10 seconds
<i>In Situ</i>	PNG	.25 MB	250 MB	< 1 second

XRAGE (LANL) simulation

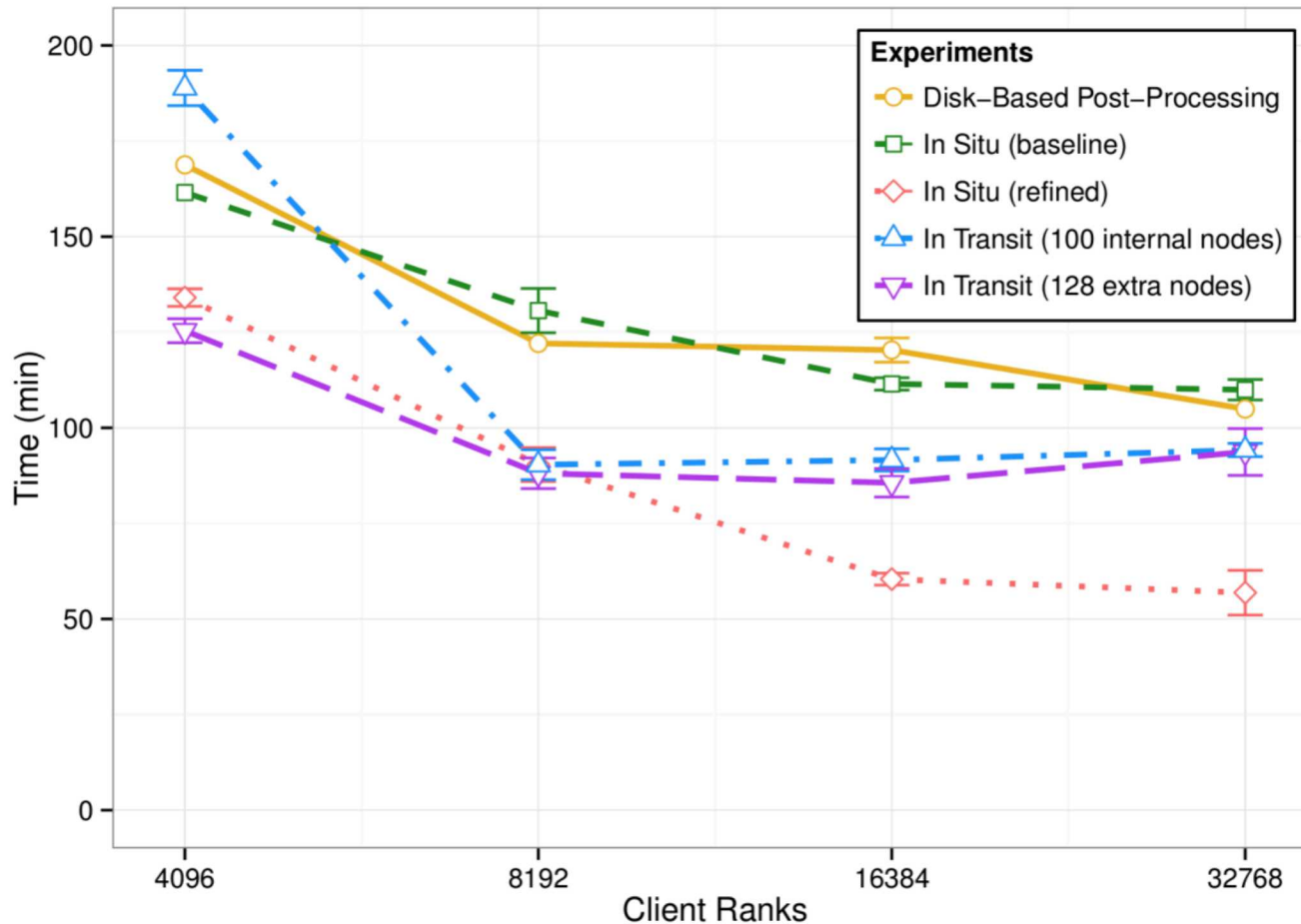
Access to More Data



CTH (Sandia) simulation with roughly equal data stored at simulation time

Reflections and shadows added in post-processing for both examples

Faster Time to Solution

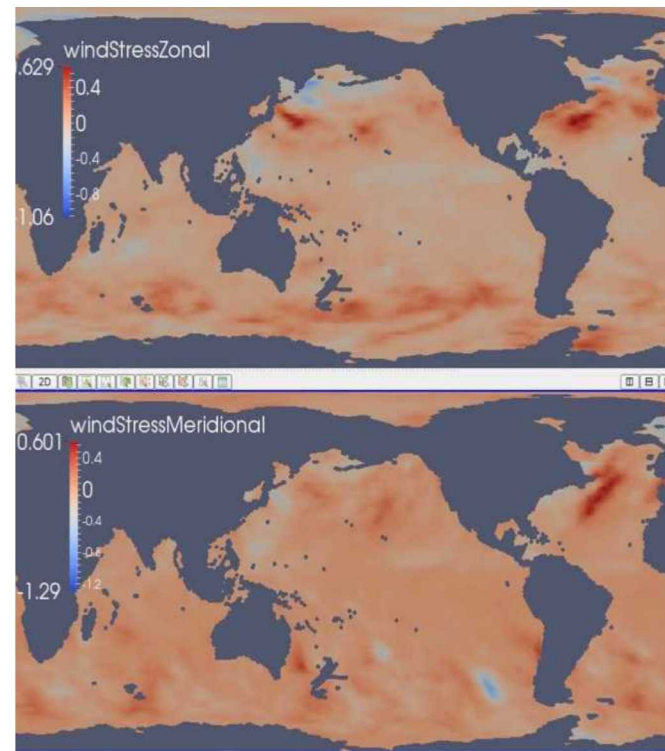
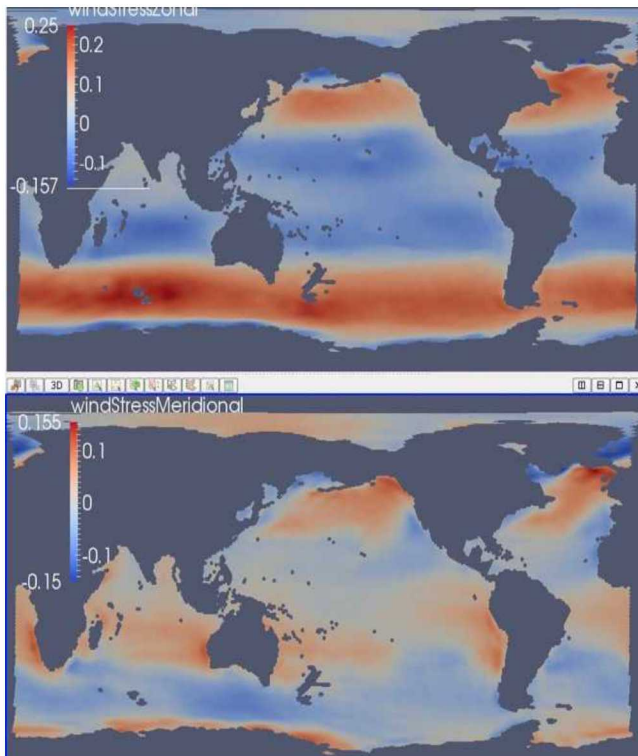


CTH (Sandia) simulations comparing different workflows

Quick and Easy Run-Time Checks

Expected wind stress field at the surface of the ocean

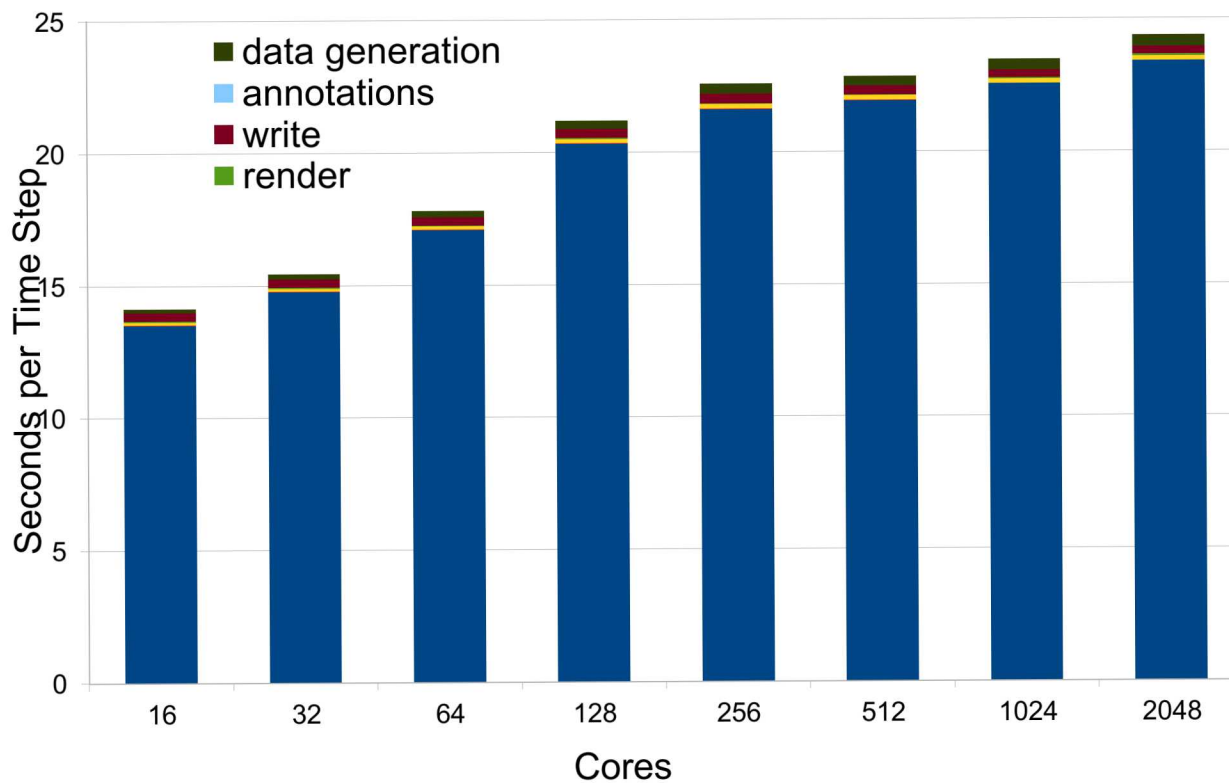
Wind stress in new run, quick glance indicates using wrong wind stress



MPAS-O (LANL) simulation

Kitware

Small Run-Time Overhead



XRAGE (LANL) simulation

High Level View

Simulation Users

- Knowledge of ParaView as a post-processing/analysis tool
 - Basic interaction with GUI Catalyst script generator plugin
 - Incremental knowledge increase to use the *in situ* tools from basic ParaView use
- Programming knowledge can be useful to extend the tools

Simulation Developers

- Pass necessary simulation data to ParaView Catalyst
- Need sufficient knowledge of both codes
 - VTK for grids and field data
 - ParaView Catalyst libraries
- Transparent to simulation users
- Extensible

In Situ Exercise

- \$PVSOURCE/Examples/Catalyst/PythonFullExample
 - fedriver.py - simulation main loop
 - fedatastructures.py - toy simulation kernel on regular grid
 - coprocessing.py - generic adaptor that runs catalyst scripts
 - gridwriter.py - Catalyst script with simple pipeline to output full grid
 - cpscript.py – Catalyst script with simple pipeline to output slices

- Run it in parallel

```
<path to>mpiexec -np 2 \  
  <path to>pvbatch --symmetric \  
  fedriver.py gridwriter.py
```

- Examine output with ParaView

```
<path to>paraview full_grid_0.pvti
```

ParaView GUI Plugin

- Creates Catalyst scripts (e.g. gridwriter.py) to insert into simulation runs
- Mostly just use ParaView as normal
 - Setup desired pipelines
 - Ideally, start with a representative data set from the simulation
- Plugin lets you add extra pipeline information to tell what to take in and output during simulation run
 - What will simulation produce?
 - Add data extract writers
 - Create screenshots to output

In Situ Exercise – Load Plugin

View Sources Filters **Tools** Catalyst Macros Help

- Create Custom Filter...
- Add Camera Link...
- Manage Custom Filters...
- Manage Links...
- Manage Plugins...**
- Record Test...
- Play Test...
- Lock View Size
- Lock View Size Custom...
- Timer Log
- Output Window
- Python Shell
- Start Trace

are automatically searched for in /media/ssddrive/BUILDS/ParaView/debug/bin/plugins.

Local Plugins

Name	Property
NonOrthogonalSource	Not Loaded
SciberQuestToolKit	Not Loaded
QuadView	Not Loaded
pvNektarReader	Not Loaded
GMVReader	Not Loaded
AnalyzeNifTIO	Not Loaded
H5PartReader	Not Loaded
EyeDomeLightingView	Not Loaded
MobileRemoteControl	Not Loaded
UncertaintyRendering	Not Loaded
PacMan	Not Loaded
CatalystScriptGeneratorPlugin	Not Loaded
Version	
Location	/media/ssddrive/BUILDS/P
Required Plugins	
Status	Not Loaded
Auto Load	<input checked="" type="checkbox"/>
Moments	Not Loaded
StreamingParticles	Not Loaded
ArrowGlyph	Not Loaded
SurfaceLIC	Not Loaded
SierraPlotTools	Not Loaded
RGBZView	Loaded
SLACTools	Not Loaded
PointSprite_Plugin	Not Loaded
vtkPVInitializerPlugin	Loaded

Load New ... Load Selected Remove

Load New ... Load Selected Remove

Close

In Situ Exercise – New Menus

CoProcessor Writers

Export State

Parallel Hierarchical Box Data Writer

Parallel MultiBlockDataSet Writer

Parallel Image Data Writer

Parallel PolyData Writer

Parallel Rectilinear Grid Writer

Parallel Structured Grid Writer

Parallel UnstructuredGrid Writer

A decorative graphic in the top left corner features a white star with a red and blue trail, suggesting motion or a simulation path.

In Situ Exercise

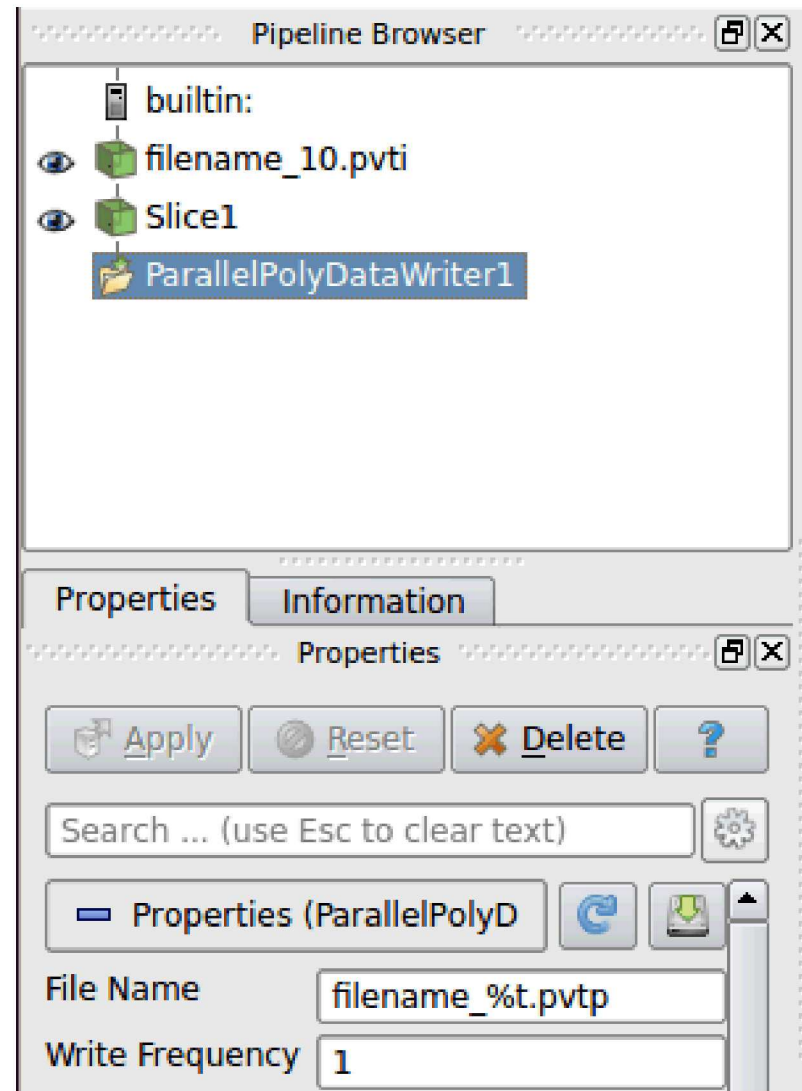
- Bootstrap with some similar data
fedriver.py made fullgrid_*.pvti files
open one
- Customize the visualization pipeline for simulation
Produce different results
Almost anything ParaView can make is fair game
 - Insert an Extract Subset Filter
 - Insert a writer to have the simulation save its output
 - Export View
 - Save Script

In Situ Exercise – Adding in Writers

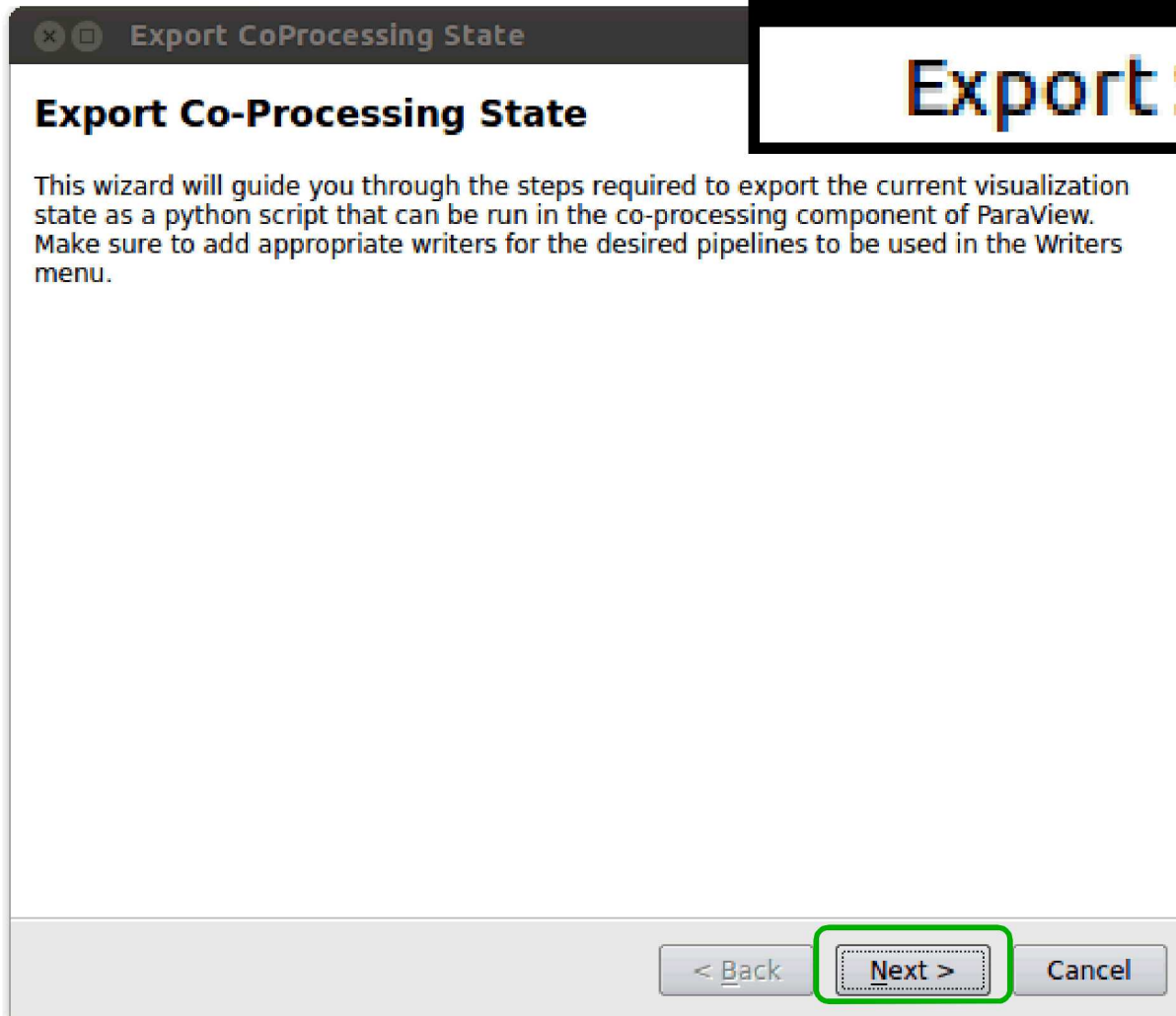


- Only valid writers available in Writers menu
- Parameters:
 - File Name – %t gets replaced with time step
 - Write Frequency

Parallel Hierarchical Box Data Writer
Parallel MultiBlockDataSet Writer
Parallel Image Data Writer
Parallel PolyData Writer
Parallel Rectilinear Grid Writer
Parallel Structured Grid Writer
Parallel UnstructuredGrid Writer



In Situ Exercise – Exporting the Script

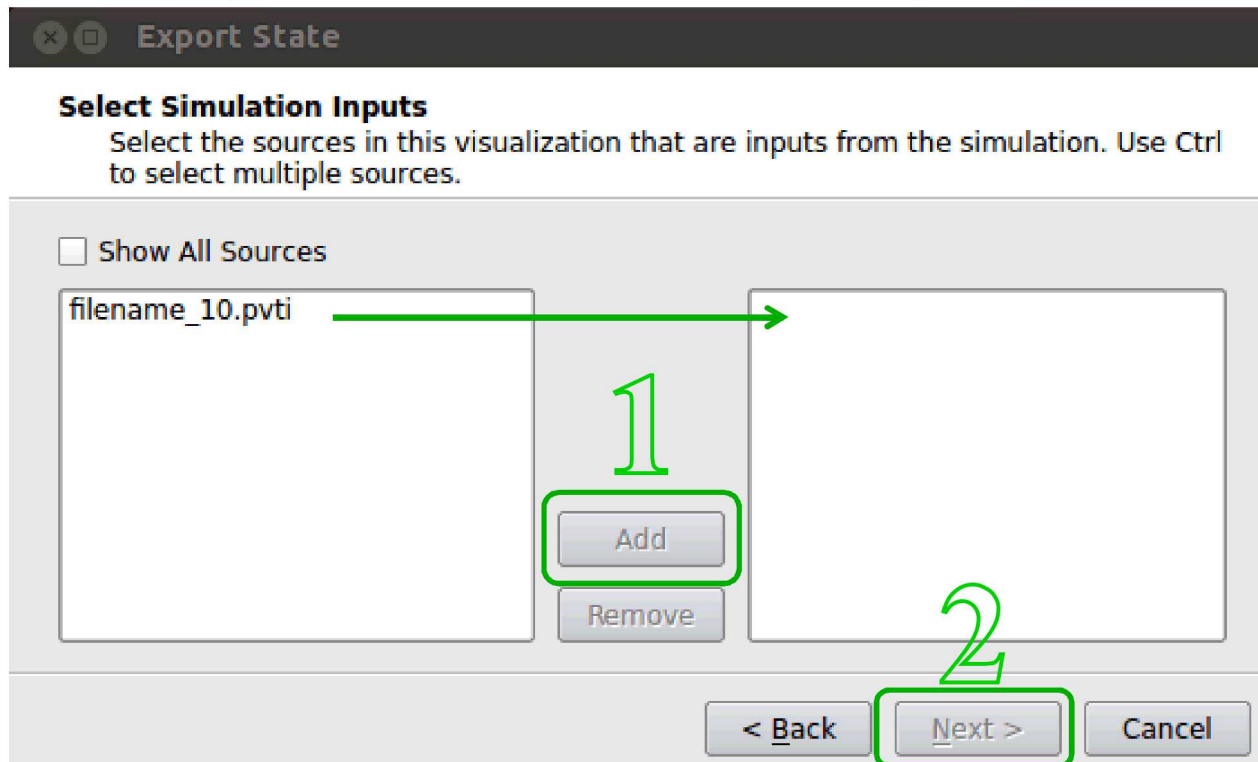


CoProcessor

Export State

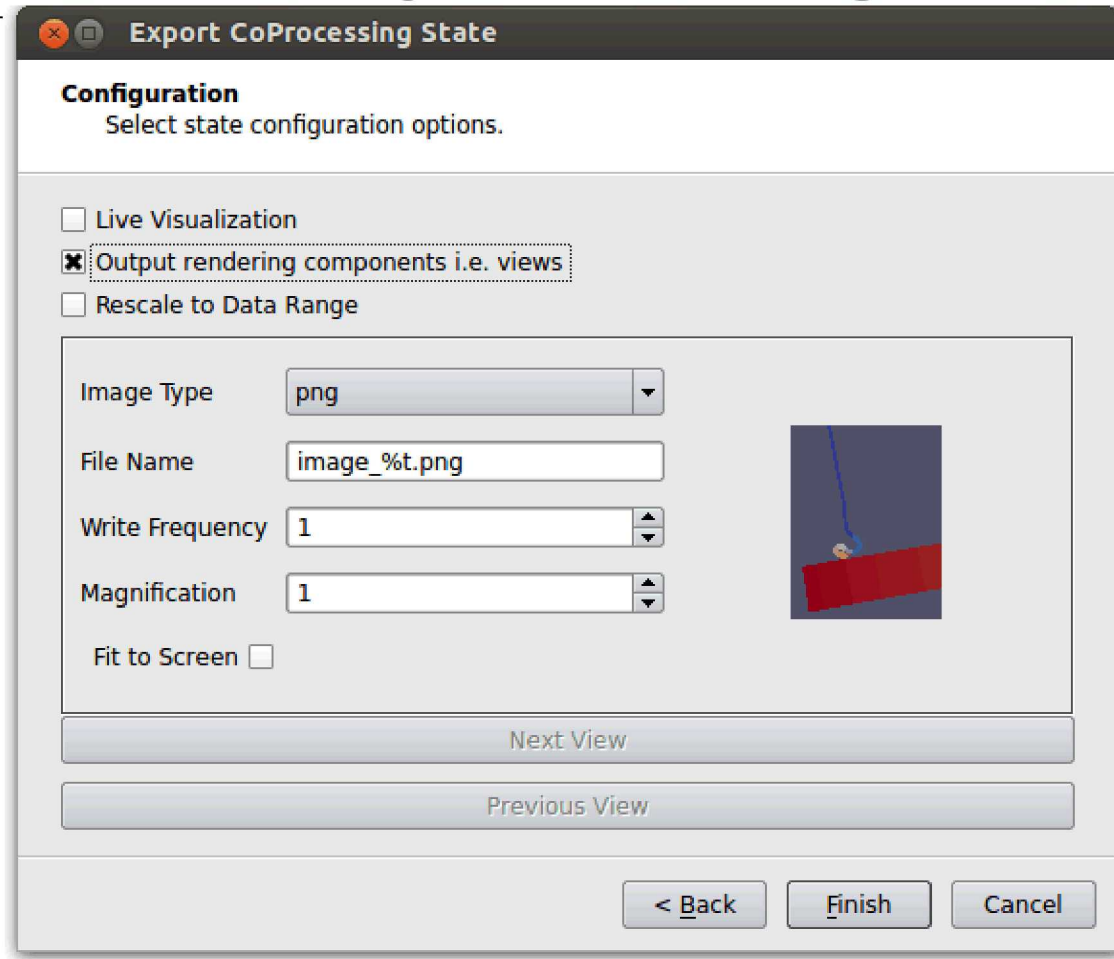
In Situ Exercise – Select Inputs

- Usually only a single input but can have multiple inputs
- Each pipeline source is a potential input



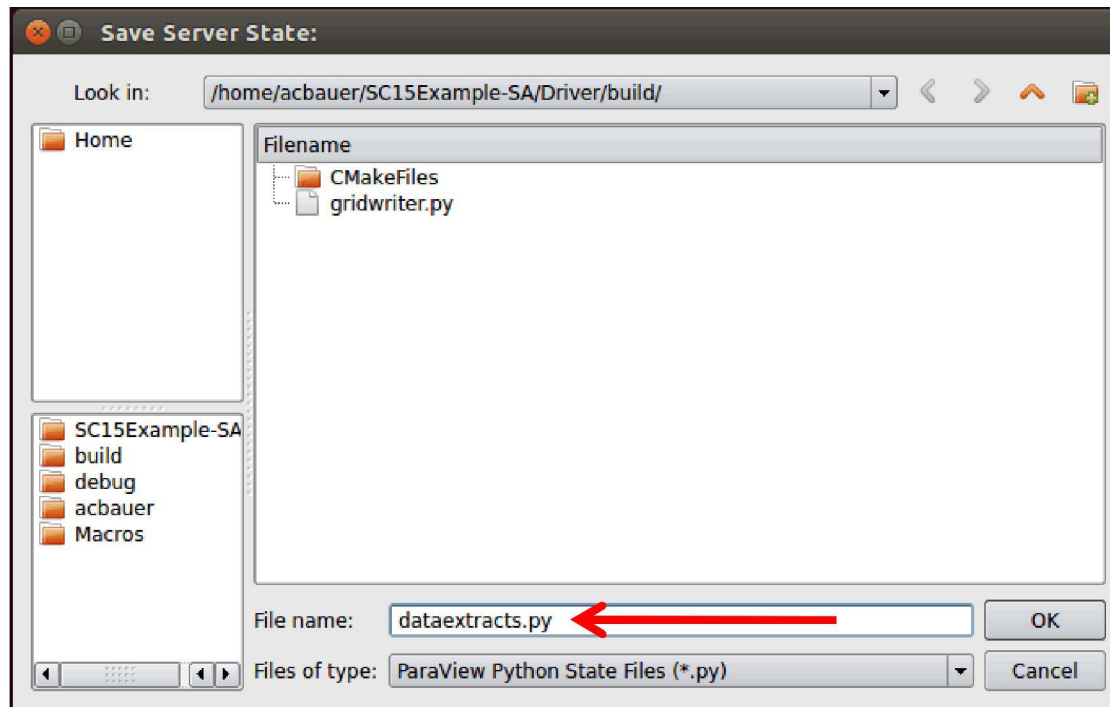
In Situ Exercise – Specify Renderings

- Parameters/Options:
 - Live visualization
 - Rescale to Data Range (all images)
 - Individual images
 - Image Type
 - File Name
 - Write Frequency
 - Magnification
 - Fit to Screen
- %t gets replaced with time step



In Situ Exercise – Save the Script

- Click on **Finish** and save script



A decorative graphic in the top left corner features a white star with a red and blue trail, suggesting motion or a path.

In Situ Exercise

- Run the simulation again

```
<path to>mpiexec -np 2 \  
  <path to>pvbatch --symmetric \  
  fedriver.py dataextracts.py
```

- It will produce requested data subsets and rendered images

ParaView Catalyst Possibilities

- Just scratched the surface
 - Don't need ParaView GUI
 - Don't need Python
 - Don't need full ParaView library

 - Can connect to running simulations
 - Output Cinema databases for image based deferred rendering interactive and queryable visualization.

Want to Learn More?

- Catalyst User's Guide:
 - http://www.paraview.org/files/catalyst/docs/ParaViewCatalystUsersGuide_v2.pdf
- Website:
 - <http://catalyst.paraview.org>
- Examples:
 - In source code under the Examples/Catalyst subdirectory.
 - <https://gitlab.kitware.com/paraview/paraview/tree/master/Examples/Catalyst>

Further Reading

- ParaView's documentation page.
 - <http://www.paraview.org/documentation>
- *The ParaView User's Guide*
- <http://www.paraview.org/Wiki/ParaView>
- http://www.paraview.org/Wiki/Setting_up_a_ParaView_Server

 A decorative graphic in the top-left corner featuring a white star and red and blue stripes.

Further Reading

Visualization and Customization

- Will Schroeder, Ken Martin, and Bill Lorensen. *The Visualization Toolkit*. Kitware, Inc., fourth edition, 2006.
- Kitware Inc. *The VTK User's Guide*. Kitware, Inc., 2006.
- Jasmin Blanchette and Mark Summerfield. *C++ GUI Programming with Qt 4*. Prentice Hall, 2006.



Further Reading

Parallel VTK Topics

- Kenneth Moreland. “A Survey of Visualization Pipelines.” *IEEE Transactions on Visualization and Computer Graphics*, 19(3), March 2013. DOI 10.1109/TVCG.2012.133.
- James Ahrens, Charles Law, Will Schroeder, Ken Martin, and Michael Papka. “A Parallel Approach for Efficiently Visualizing Extremely Large, Time-Varying Datasets.” Technical Report #LAUR-00-1620, Los Alamos National Laboratory, 2000.
- James Ahrens, Kristi Brislawn, Ken Martin, Berk Geveci, C. Charles Law, and Michael Papka. “Large-Scale Data Visualization Using Parallel Data Streaming.” *IEEE Computer Graphics and Applications*, 21(4): 34–41, July/August 2001.

Further Reading

Advanced Pipeline Execution

- Andy Cedilnik, Berk Geveci, Kenneth Moreland, James Ahrens, and Jean Farve. “Remote Large Data Visualization in the ParaView Framework.” *Eurographics Parallel Graphics and Visualization 2006*, pg. 163–170, May 2006.
- James P. Ahrens, Nehal Desai, Patrick S. McCormic, Ken Martin, and Jonathan Woodring. “A Modular, Extensible Visualization System Architecture for Culled, Prioritized Data Streaming.” *Visualization and Data Analysis 2007, Proceedings of SPIE-IS&T Electronic Imaging*, pg 64950I-1–12, January 2007.
- John Biddiscombe, Berk Geveci, Ken Martin, Kenneth Moreland, and David Thompson. “Time Dependent Processing in a Parallel Pipeline Architecture.” *IEEE Visualization 2007*. October 2007.



Further Reading

Parallel Rendering



- Kenneth Moreland, Brian Wylie, and Constantine Pavlakos. “Sort-Last Parallel Rendering for Viewing Extremely Large Data Sets on Tile Displays.” *Proceedings of IEEE 2001 Symposium on Parallel and Large-Data Visualization and Graphics*, pg. 85–92, October 2001.
- Kenneth Moreland and David Thompson. “From Cluster to Wall with VTK.” *Proceedings of IEEE 2003 Symposium on Parallel and Large-Data Visualization and Graphics*, pg. 25–31, October 2003.
- Kenneth Moreland, Lisa Avila, and Lee Ann Fisk. “Parallel Unstructured Volume Rendering in ParaView.” *Visualization and Data Analysis 2007, Proceedings of SPIE-IS&T Electronic Imaging*, pg. 64950F-1–12, January 2007.

