Data visualisation
Paraview

Zakaria Meliani

Observatoire de Paris

Outline

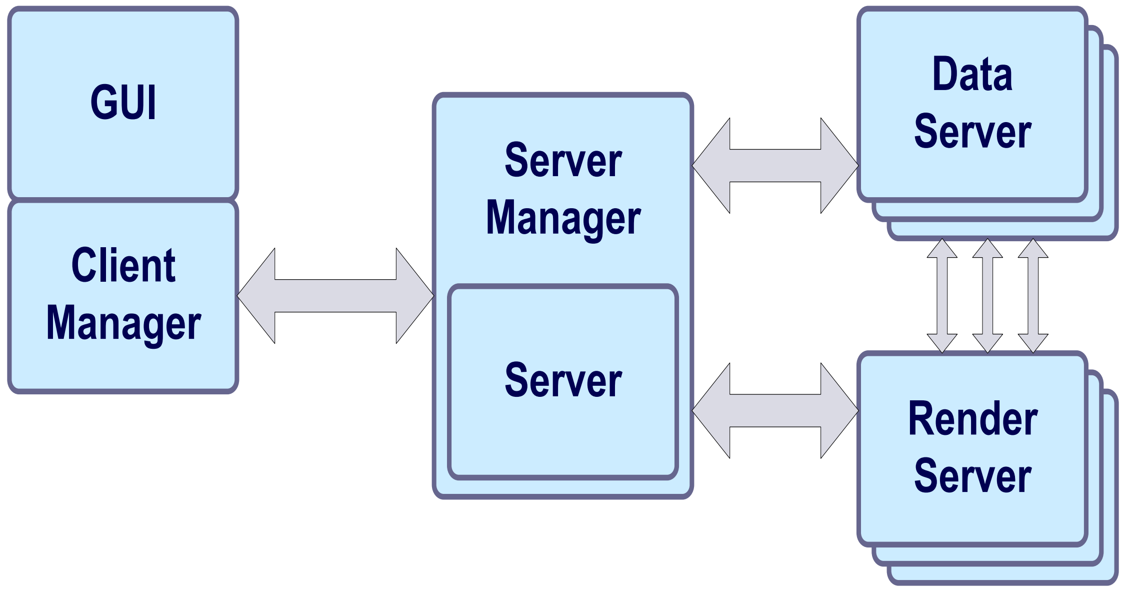
- Paraview
- Basic use
- Save state
- python

Paraview

- http://www.paraview.org/
- Open-source, multi-platform parallel data analysis and visualization application
- Client–server architecture to facilitate remote visualization of datasets, and generates level of detail(LOD) models to maintain interactive frame rates for large datasets.
- designed for data parallelism on shared-memory or distributed-memory multicomputers and clusters.
- feature-rich , flexible and intuitive user interface
- Extensible architecture based on open standards
- Scriptable via Python
- Saves animations
- built on top of the Visualization ToolKit (VTK) library
- Contributors :
    - Kitware, Inc.
    - Sandia National Laboratory
    - Los Alamos National Laboratory
    - Army Research Laboratory

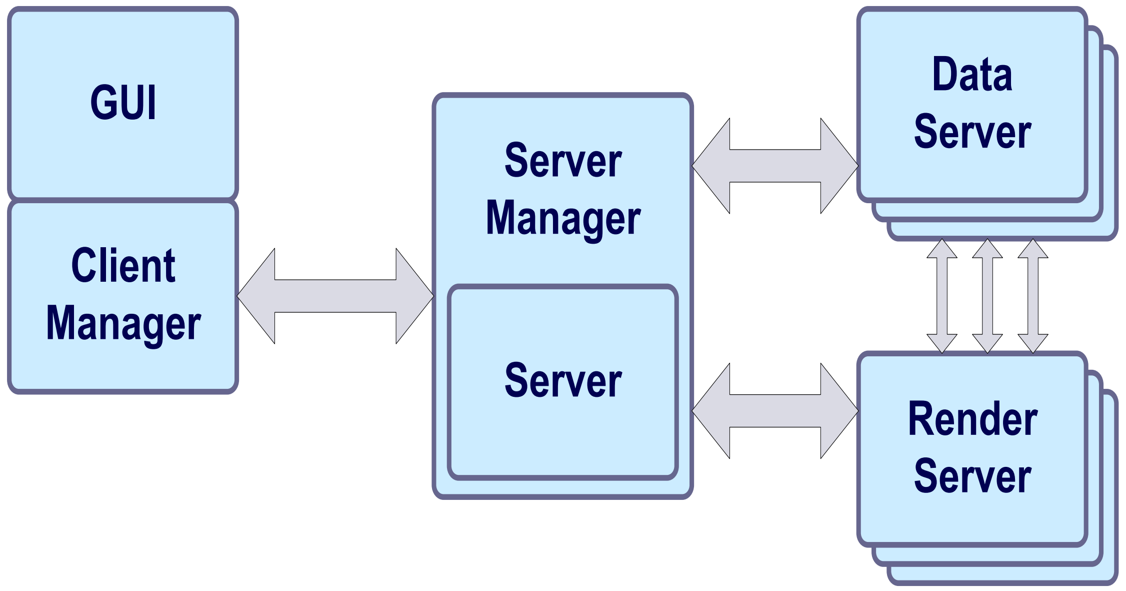
ParaView Architecture

- Parallel IO
- Scalability
- Run as a batch application using the Python interface
- Software Rendering
- Ressource allocation



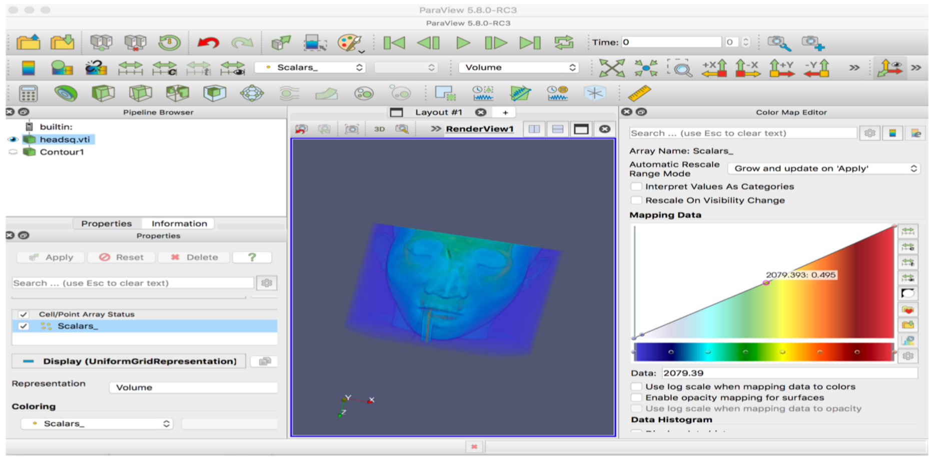
ParaView Sever

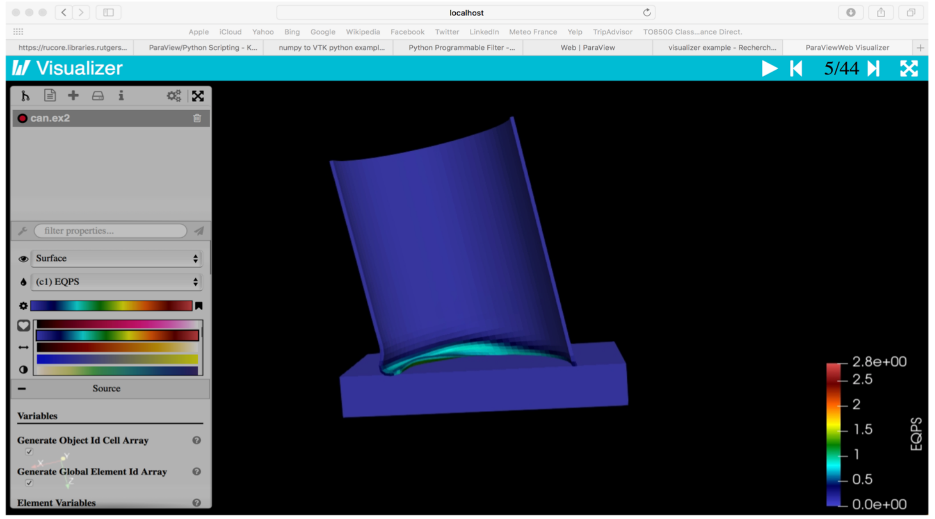
- Data/render server:
  - C++
  - Client/server wrapping
  - No direct access
- Server manager:
  - C++ API
  - XML configuration
  - Tcl, Python or Java scripting (optional)



ParaView Client

Desktop Client:	KWWidgets (Tcl/Tk and C++)
	Tcl scripting of GUI
	XML configuration of GUI
Web Client:	DHTML/JavaScript
	Server code uses Python





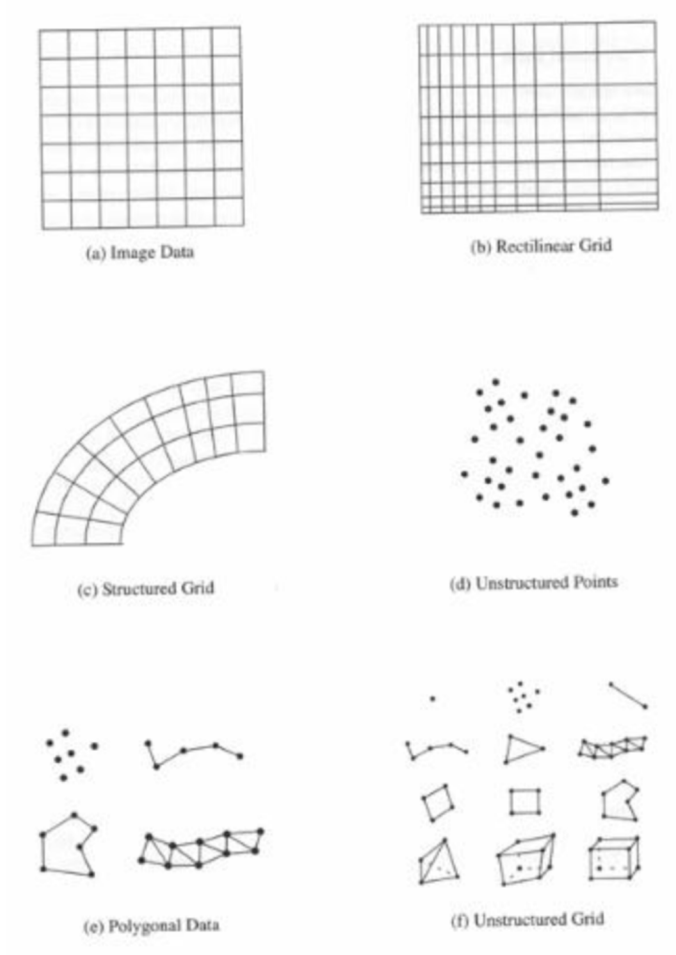
Data import

Paraview input data types

- Supports a wide variety of data types
  - Structured grids
    - uniform rectilinear, non-uniform rectilinear, and curvilinear
  - Unstructured grids
  - Polygonal data
  - Images
  - Multi-block
  - AMR
- Time series support

ParaView Dataset Types

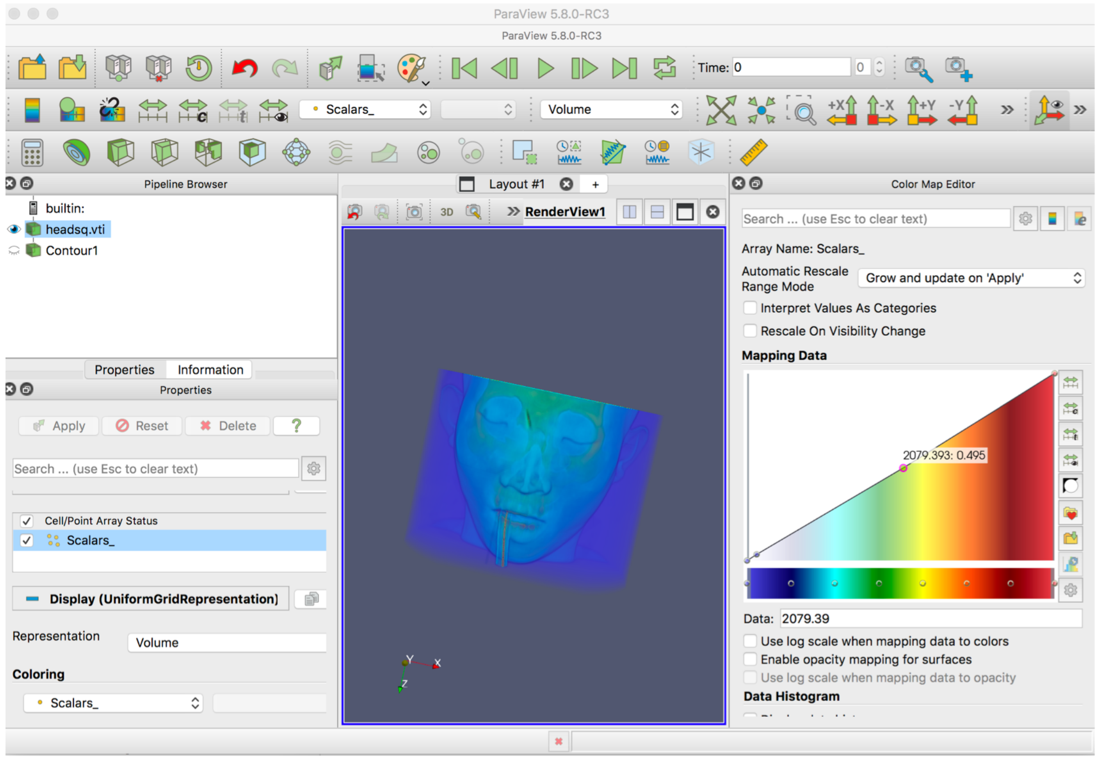
- Image/Volume Data
- Rectilinear Grid
- Structured Grid
- Unstructured Points
- Polygonal Data
- Unstructured Grid



Paraview data

- Scalar
- Vector
- Tensor
- texture coordinate
- Normals

Paraview



Array Name: Scalars_

Automatic Rescale Range Mode: Grow and update on 'Apply'

Interpret Values As Categories

Rescale On Visibility Change

Data: 2079.39

Use log scale when mapping data to colors

Enable opacity mapping for surfaces

Use log scale when mapping data to opacity

Representation: Volume

Coloring: Scalars_

Data format

CSV

Image

VTK (PolyData, UnstructuredGrid, StructuredGrid, ImageData, RectilinearGrid, Multiblock, Heirarchical etc.)

Ensight (case files, ASCII/Binary data)

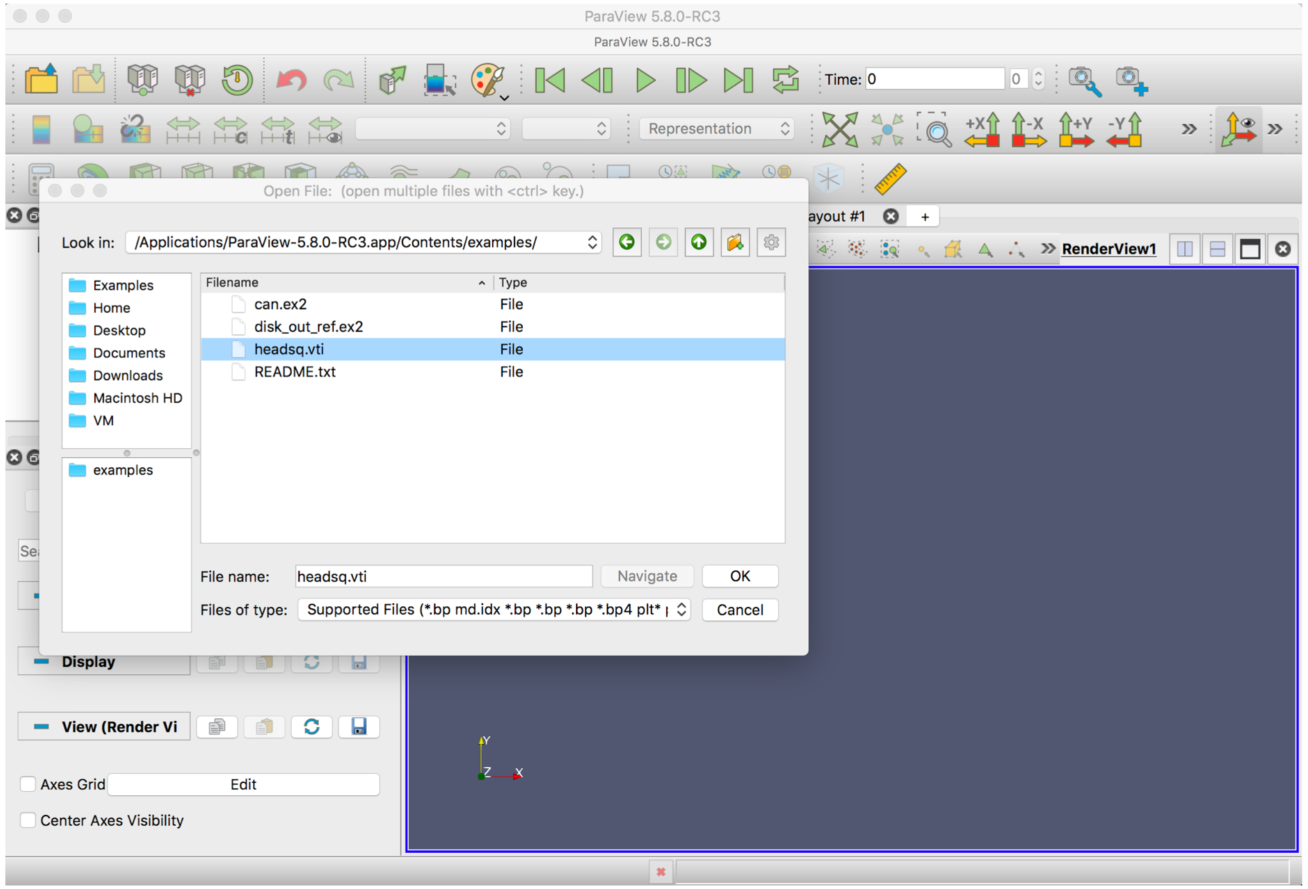
Plot3D

XYZ file

Xdmf

netCDF, HDF5

Usr can add reader plugin to extend support to other formats.



Look in: /Applications/ParaView-5.8.0-RC3.app/Contents/examples/

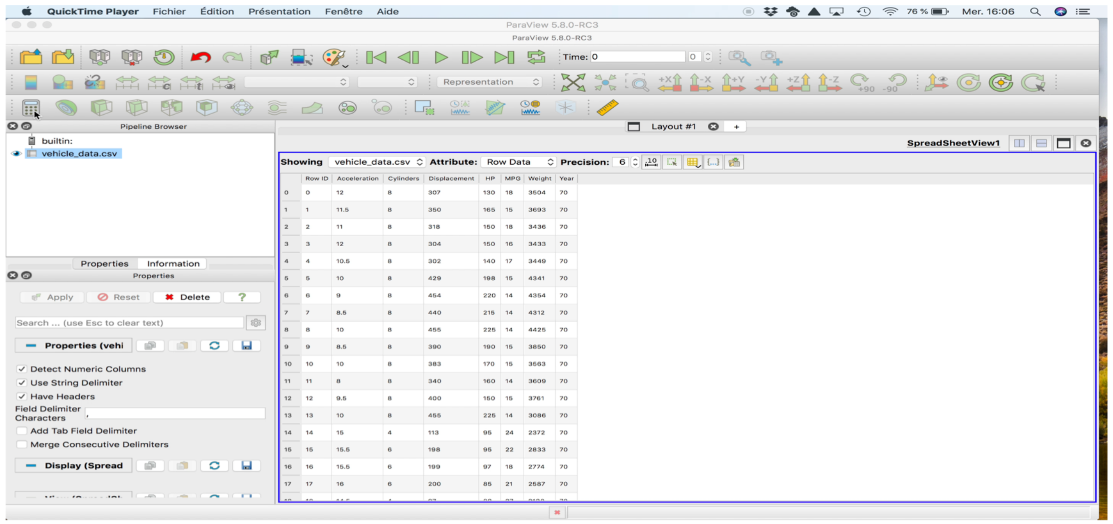
Filename	Type
can.ex2	File
disk_out_ref.ex2	File
headsq.vti	File
README.txt	File

File name: headsq.vti

Files of type: Supported Files (*.bp md.idx *.bp *.bp *.bp *.bp4 plt*

Getting start

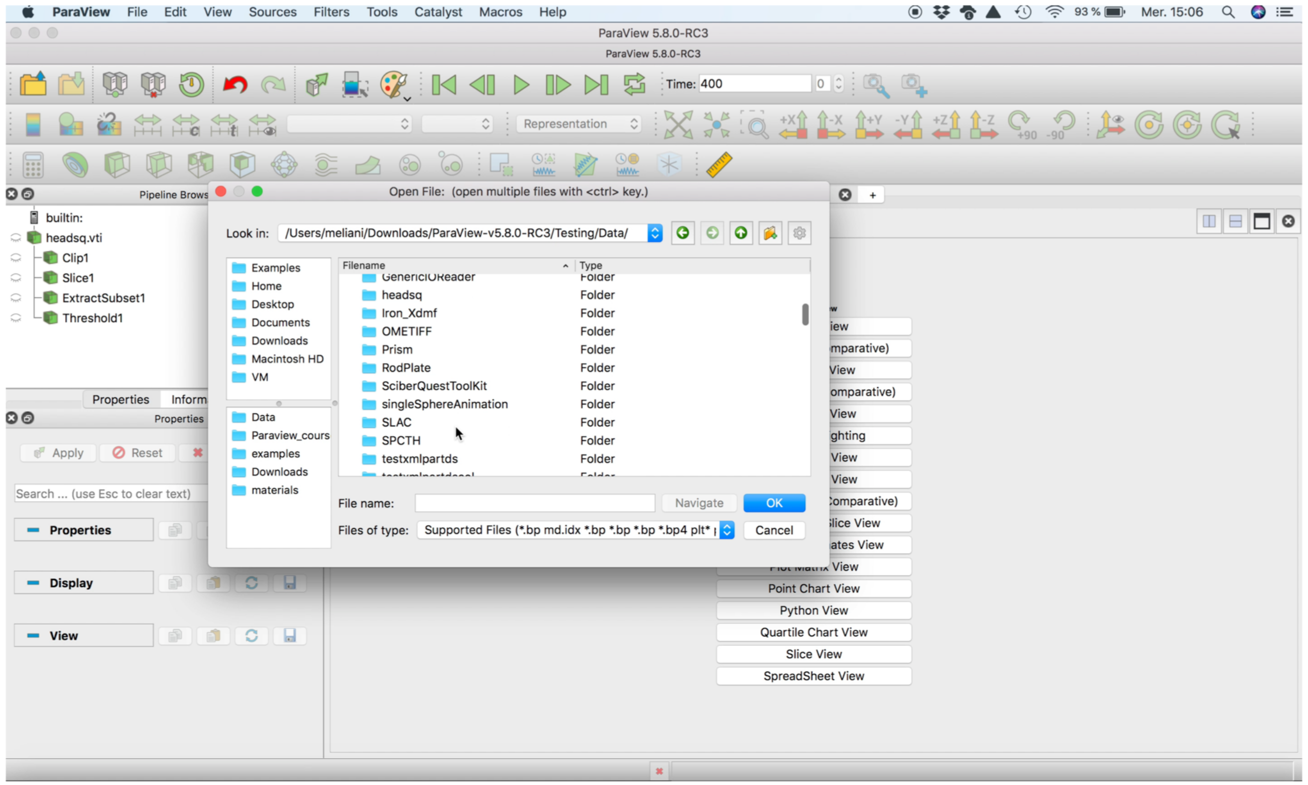
- If is possible even to use **CSV** file
- Use calcutor
-



Showing vehicle_data.csv Attribute: Row Data Precision: 6

	Row ID	Acceleration	Cylinders	Displacement	HP	MPG	Weight	Year
0	0	12	8	307	130	18	3504	70
1	1	11.5	8	350	165	15	3693	70
2	2	11	8	318	150	18	3436	70
3	3	12	8	304	150	16	3433	70
4	4	10.5	8	302	140	17	3449	70
5	5	10	8	429	198	15	4341	70
6	6	9	8	454	220	14	4354	70
7	7	8.5	8	440	215	14	4312	70
8	8	10	8	455	225	14	4425	70
9	9	8.5	8	390	190	15	3850	70
10	10	10	8	383	170	15	3563	70
11	11	8	8	340	160	14	3609	70
12	12	9.5	8	400	150	15	3761	70
13	13	10	8	455	225	14	3086	70
14	14	15	4	113	95	24	2372	70
15	15	15.5	6	198	95	22	2833	70
16	16	15.5	6	199	97	18	2774	70
17	17	16	6	200	85	21	2587	70

Properties (vehicle_data.csv):
- ✓ Detect Numeric Columns
- ✓ Use String Delimiter
- ✓ Have Headers
- Field Delimiter Characters ,
- Add Tab Field Delimiter
- Merge Consecutive Delimiters



Open File: (open multiple files with <ctrl> key.)

Look in: /Users/meliani/Downloads/ParaView-v5.8.0-RC3/Testing/Data/

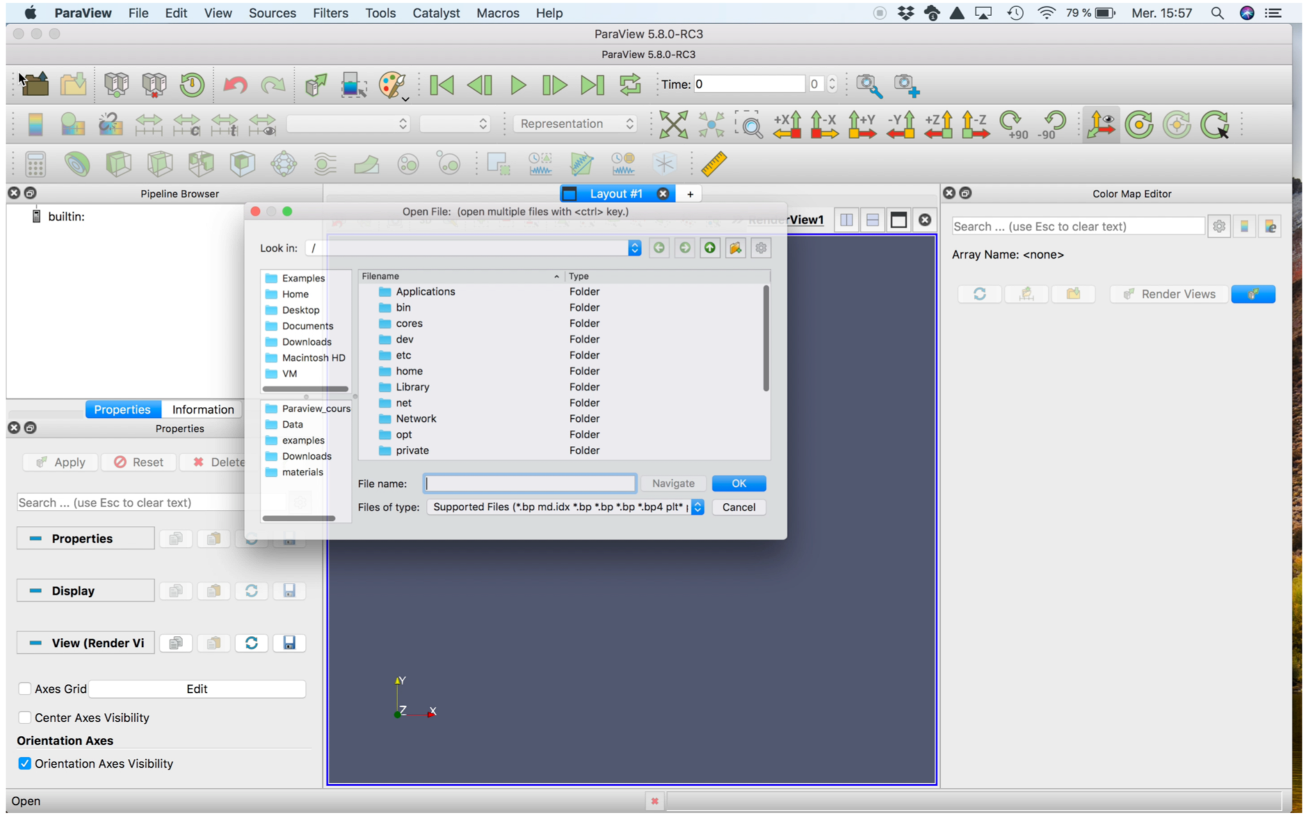
Filename	Type
GenericIOReader	Folder
headsq	Folder
Iron_Xdmf	Folder
OMETIFF	Folder
Prism	Folder
RodPlate	Folder
SciberQuestToolKit	Folder
singleSphereAnimation	Folder
SLAC	Folder
SPCTH	Folder
testxmlpartds	Folder

File name:
Files of type: Supported Files (*.bp md.idx *.bp *.bp *.bp *.bp4 plt*

Use image (jpeg)

It is possible even to use **jpeg** file

use texture map with various geometries



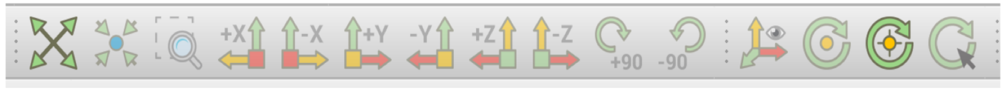
Visualization Algorithms

Supports a wide variety of visualization algorithms

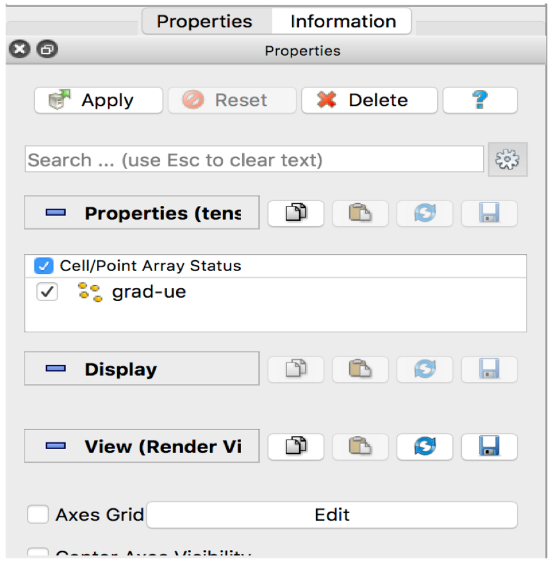
- Isosurfaces
- Cutting planes
- Streamlines
- Glyphs
- Volume rendering
- Clipping
- 1D cut
- reflect
- texture
- interpolation

Paraview: basics

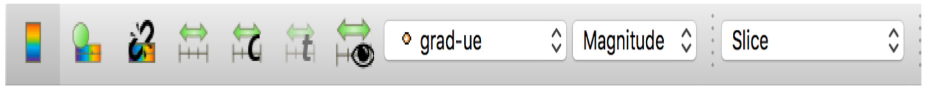
- Simple Camera Manipulation



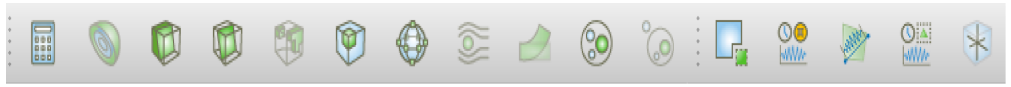
- Display Properties

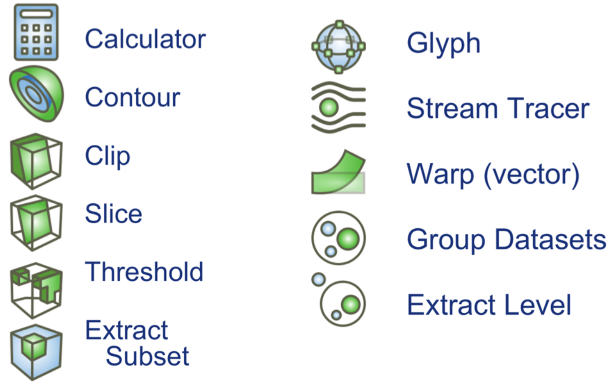


- Representation/variables/Range/colorbar

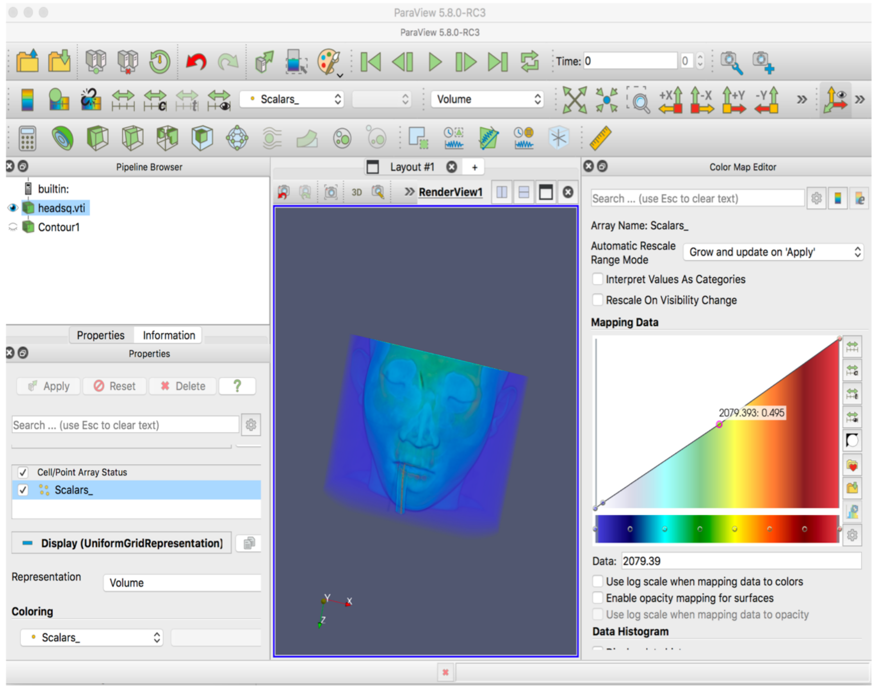


- Common filters





Basic use: 3D rendering/Opacity/Contour/Extract subset

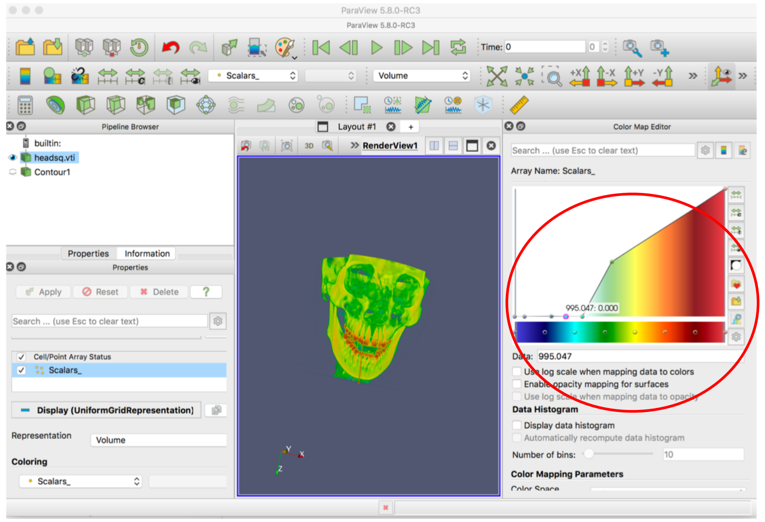

ParaView 5.8.0-RC3

Pipeline Browser: builtin: / headsq.vti / Contour1

Properties — Cell/Point Array Status: Scalars_

Display (UniformGridRepresentation)
Representation: Volume
Coloring: Scalars_

Color Map Editor
Array Name: Scalars_
Automatic Rescale Range Mode: Grow and update on 'Apply'
Interpret Values As Categories
Rescale On Visibility Change
Mapping Data
2079.393: 0.495
Data: 2079.39
Use log scale when mapping data to colors
Enable opacity mapping for surfaces
Use log scale when mapping data to opacity
Data Histogram

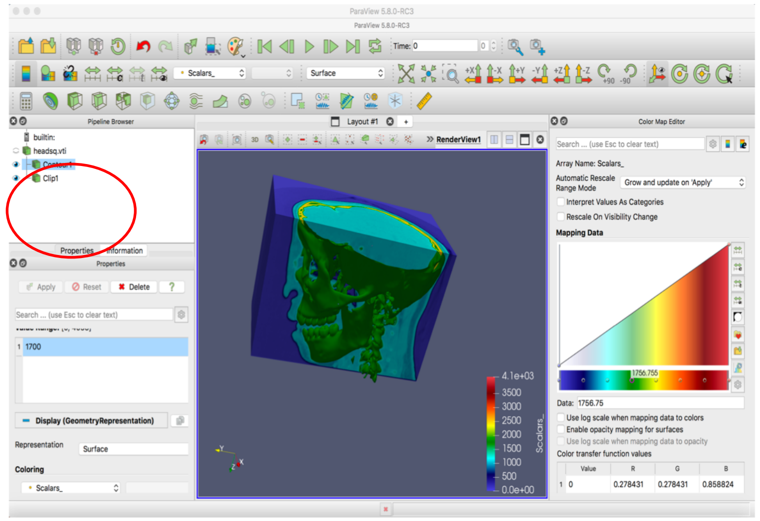

ParaView 5.8.0-RC3

Pipeline Browser: builtin: / headsq.vti / Contour1

Display (UniformGridRepresentation)
Representation: Volume
Coloring: Scalars_

Color Map Editor
Array Name: Scalars_
995.047: 0.000
Data: 995.047
Use log scale when mapping data to colors
Enable opacity mapping for surfaces
Use log scale when mapping data to opacity
Data Histogram
Display data histogram
Automatically recompute data histogram
Number of bins: 10
Color Mapping Parameters

Change opacity


ParaView 5.8.0-RC3

Pipeline Browser: builtin: / headsq.vti / Contour1 / Clip1

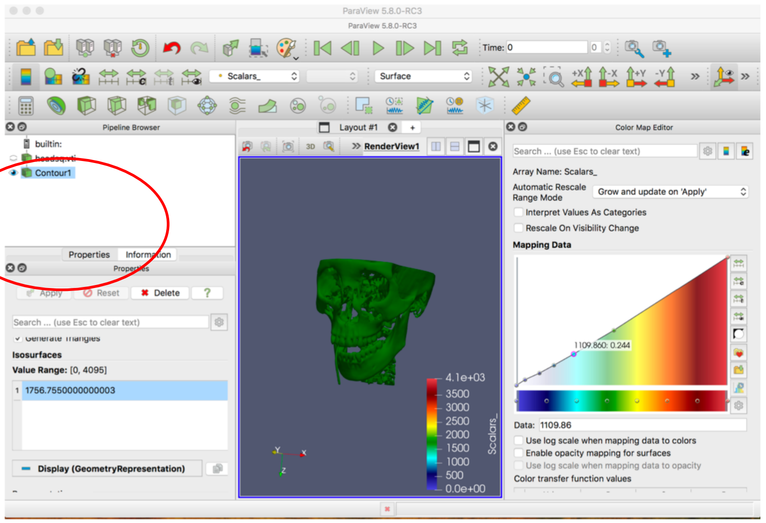
Properties: 1 1700

Display (GeometryRepresentation)
Representation: Surface
Coloring: Scalars_

Color Map Editor
Array Name: Scalars_
Automatic Rescale Range Mode: Grow and update on 'Apply'
1756.755
Data: 1756.75
Color transfer function values

	Value	R	G	B
1	0	0.278431	0.278431	0.858824

Clip


ParaView 5.8.0-RC3

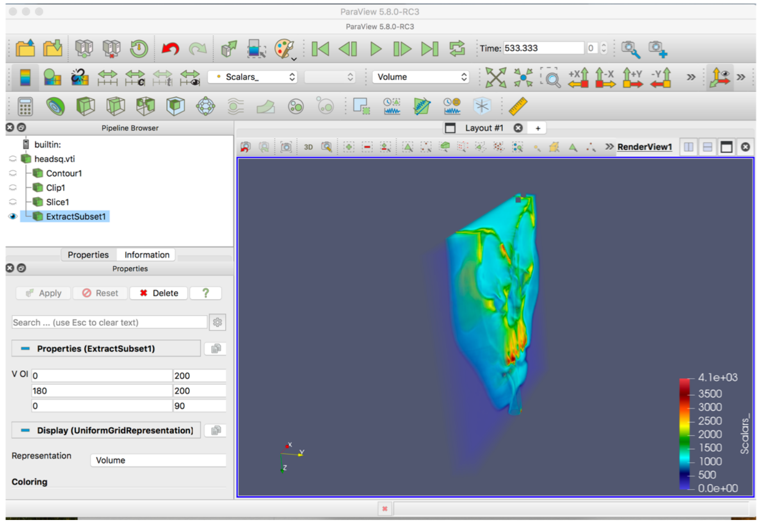
Pipeline Browser: builtin: / headsq.vti / Contour1

Properties
Generate Triangles
Isosurfaces
Value Range: [0, 4095]
1 1756.7550000000003

Display (GeometryRepresentation)

Color Map Editor
Array Name: Scalars_
Automatic Rescale Range Mode: Grow and update on 'Apply'
Interpret Values As Categories
Rescale On Visibility Change
Mapping Data
1109.860: 0.244
Data: 1109.86
Use log scale when mapping data to colors
Enable opacity mapping for surfaces
Use log scale when mapping data to opacity
Color transfer function values

Contour


ParaView 5.8.0-RC3

Pipeline Browser: builtin: / headsq.vti / Contour1 / Clip1 / Slice1 / ExtractSubset1

Properties (ExtractSubset1)

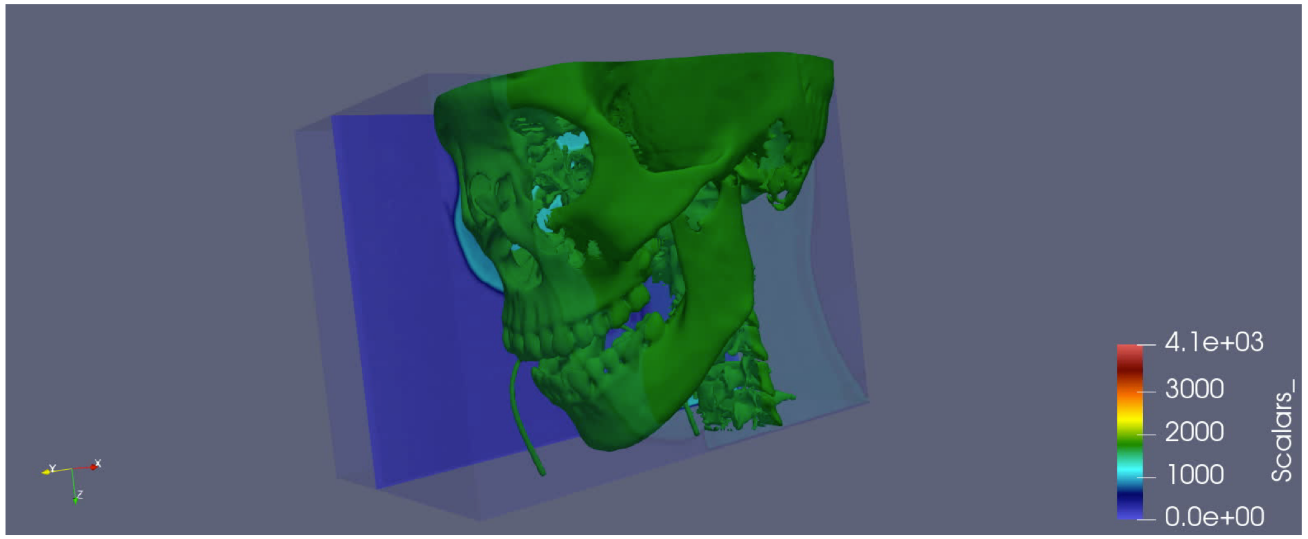
V OI	0	200
	180	200
	0	90

Display (UniformGridRepresentation)
Representation: Volume
Coloring

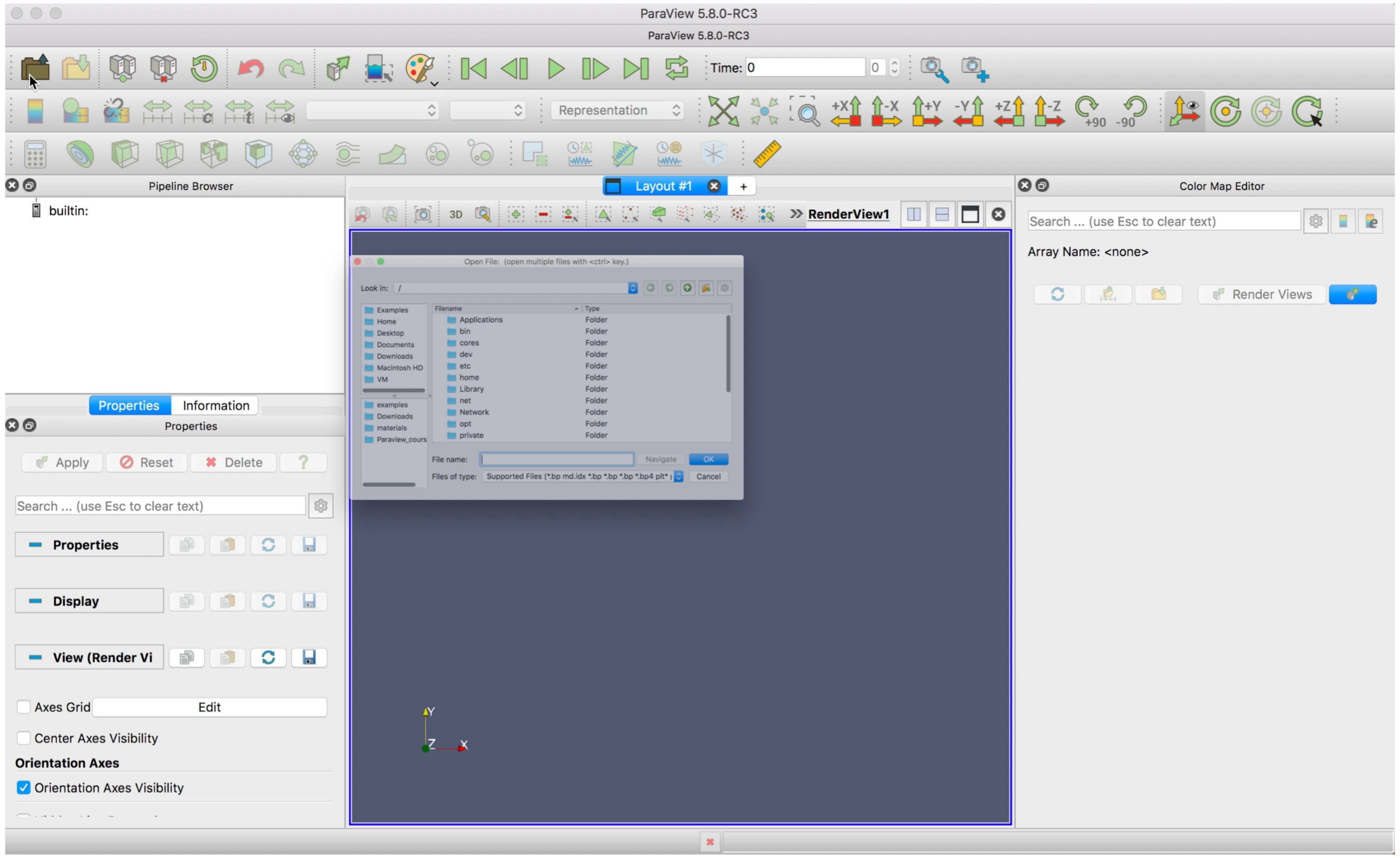
Extract subset

Camera Animation

Make animation using Animation View tools



All the steps with scalar variable



Open File: (open multiple files with <ctrl> key.)

Look in: /

Examples, Home, Desktop, Documents, Downloads, Macintosh HD, VM

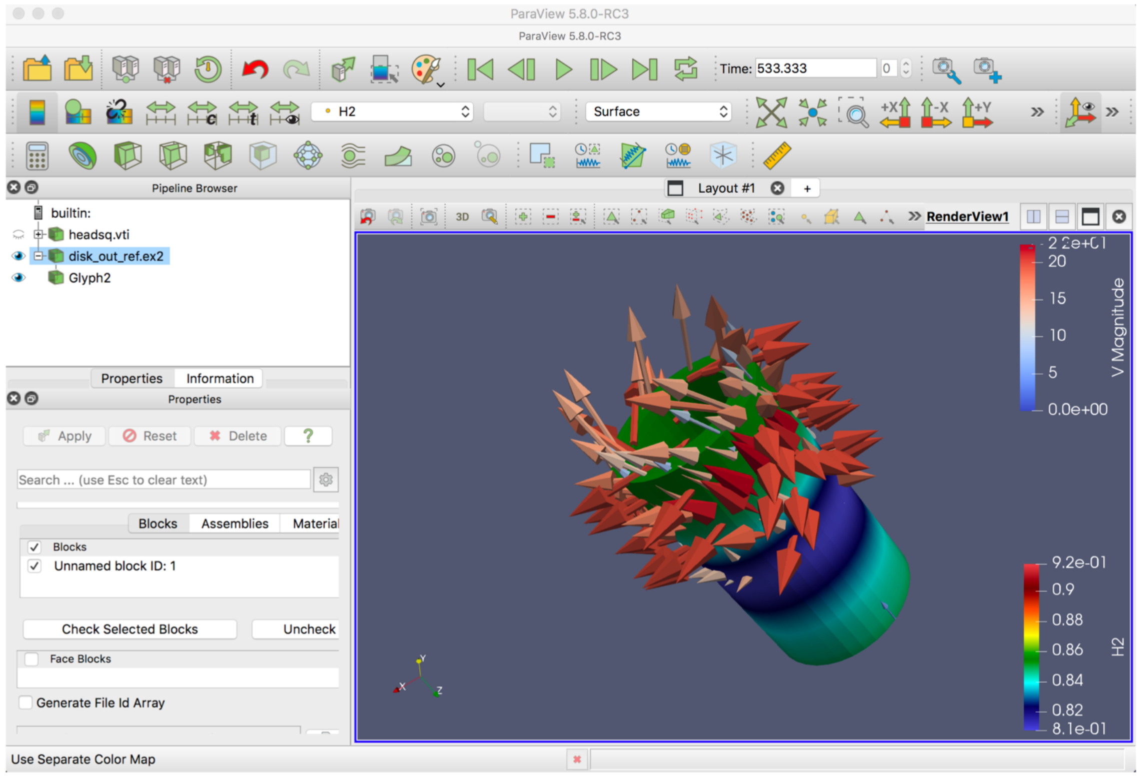
examples, Downloads, materials, Paraview_cours

Filename	Type
Applications	Folder
bin	Folder
cores	Folder
dev	Folder
etc	Folder
home	Folder
Library	Folder
net	Folder
Network	Folder
opt	Folder
private	Folder

File name: Navigate OK

Files of type: Supported Files (*.bp md.idx *.bp *.bp *.bp *.bp4 plt* Cancel

Vector: Glyphs/streamlines



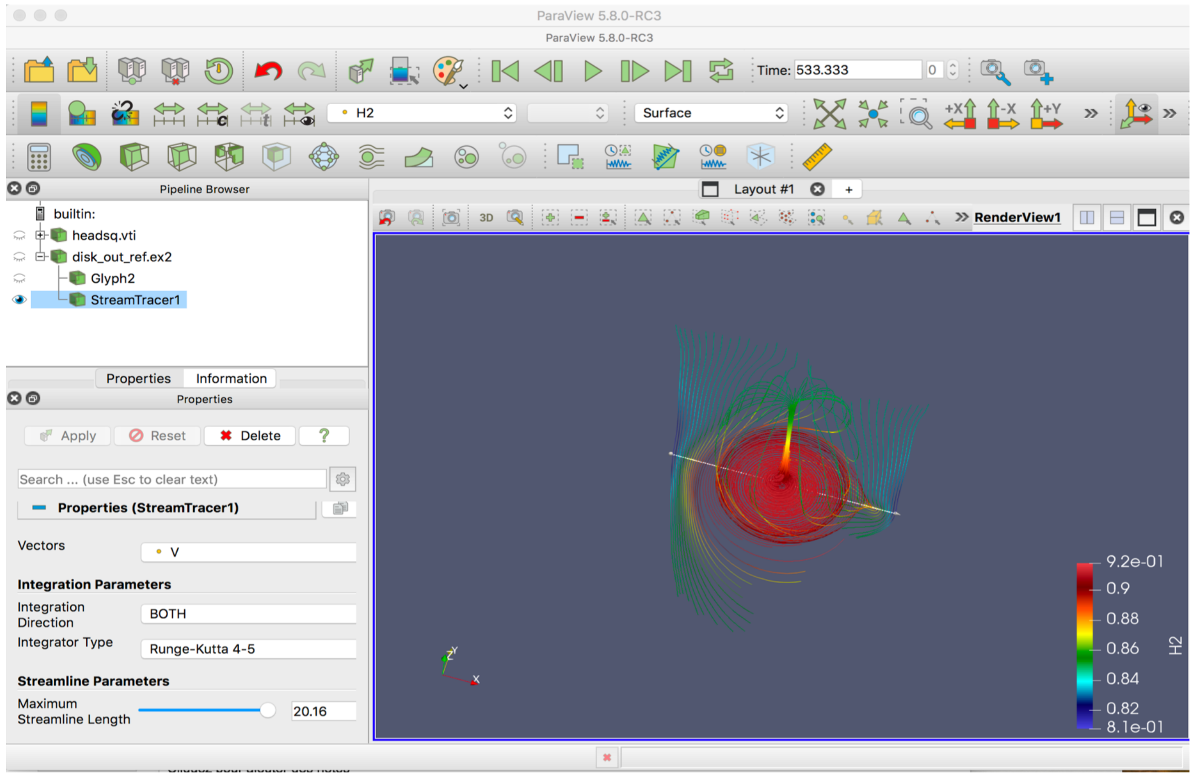
Time: 533.333
H2
Surface

- [✓] Blocks
- [✓] Unnamed block ID: 1

Check Selected Blocks · Uncheck

- [ ] Face Blocks
- [ ] Generate File Id Array

Use Separate Color Map



Time: 533.333
H2
Surface

Properties (StreamTracer1)

Vectors: V

Integration Parameters

Integration Direction: BOTH
Integrator Type: Runge-Kutta 4-5

Streamline Parameters

Maximum Streamline Length: 20.16

Save file and re-use

Save image, movie and Paraview state file

Save image/ movie

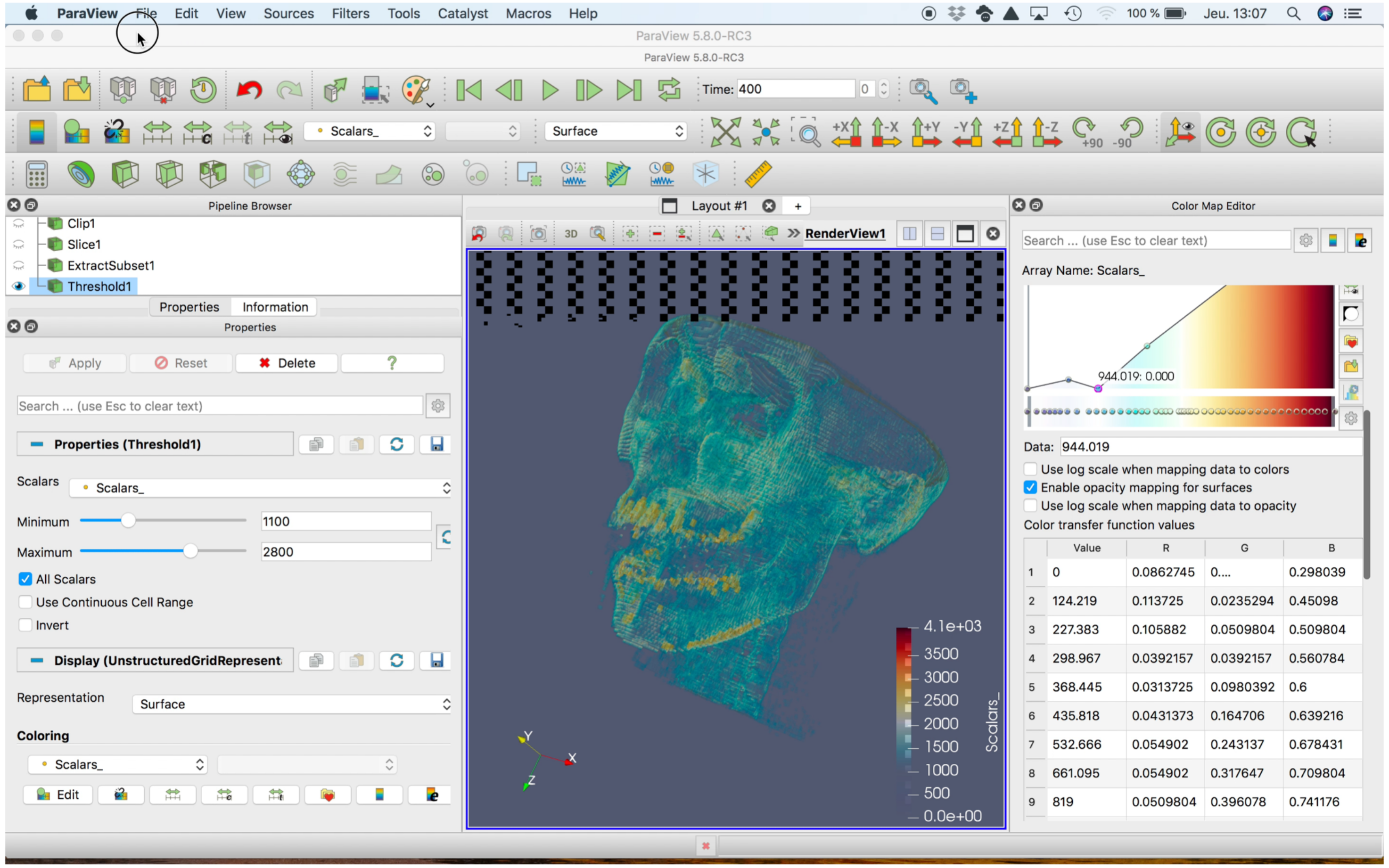
- jpeg, png, tiff, avi …

Save data

- vtk

Save state

- pvsm(paraview xml), python



Pipeline Browser: Clip1, Slice1, ExtractSubset1, Threshold1

Properties (Threshold1)

Scalars: Scalars_

Minimum: 1100

Maximum: 2800

[✓] All Scalars

[ ] Use Continuous Cell Range

[ ] Invert

Display (UnstructuredGridRepresent

Representation: Surface

Coloring: Scalars_

Array Name: Scalars_

Data: 944.019

[ ] Use log scale when mapping data to colors

[✓] Enable opacity mapping for surfaces

[ ] Use log scale when mapping data to opacity

Color transfer function values

	Value	R	G	B
1	0	0.0862745	0....	0.298039
2	124.219	0.113725	0.0235294	0.45098
3	227.383	0.105882	0.0509804	0.509804
4	298.967	0.0392157	0.0392157	0.560784
5	368.445	0.0313725	0.0980392	0.6
6	435.818	0.0431373	0.164706	0.639216
7	532.666	0.054902	0.243137	0.678431
8	661.095	0.054902	0.317647	0.709804
9	819	0.0509804	0.396078	0.741176

Save and load state

ParaView XML Record and Save State Files

The save state can be re-used

Change the input file (of the same familly)

```
<ParaView>
  <ServerManagerState version="5.8.0">
    <Proxy group="animation" type="AnimationScene" id="263"
servers="16">
      <Property name="AnimationTime"
id="263.AnimationTime" number_of_elements="1">
        <Element index="0" value="400"/>
      </Property>
…
 <Property name="RGBPoints" id="11273.RGBPoints"
number_of_elements="32">
        <Element index="0" value="0"/>
        <Element index="1" value="0.278431372549"/>
        <Element index="2" value="0.278431372549"/>
        <Element index="3" value="0.858823529412"/>
        <Element index="4" value="585.584999999999"/>
….
 <Property name="CameraFocalPoint"
id="10848.CameraFocalPoint" number_of_elements="3">
        <Element index="0" value="127.5000000000006"/>
        <Element index="1" value="127.49999999999993"/>
        <Element index="2" value="93"/>
      </Property>
…
```

Python script state

Load with paraview

Load with web client?

Change to save movie and image and use with with batch

```
# state file generated using paraview version 5.8.0-RC3

# ----------------------------------------------------------------
# setup views used in the visualization
# ----------------------------------------------------------------

# trace generated using paraview version 5.8.0-RC3
#
# To ensure correct image size when batch processing, please search
# for and uncomment the line `# renderView*.ViewSize = [*,*]`

#### import the simple module from the paraview
from paraview.simple import *
#### disable automatic camera reset on 'Show'
paraview.simple._DisableFirstRenderCameraReset()

# get the material library
materialLibrary1 = GetMaterialLibrary()

# Create a new 'Render View'
renderView1 = CreateView('RenderView’)
…
disk_out_refex2 = ExodusIIReader(FileName=['/Applications/ParaView-5.8.0-
RC3.app/Contents/examples/disk_out_ref.ex2’])
…
# create a new 'Python Calculator'
pythonCalculator2 = PythonCalculator(Input=disk_out_refex2)
pythonCalculator2.Expression = "curl(inputs[0].PointData['V'])"
pythonCalculator2.ArrayName = 'curlV'
```

VTK collection

VTK Collection - Easy to create by hand if necessary

- A General purpose holder for vtk XML files of all types
- vtu=unstructured, vtp=polydata, vtr=rectilinear, vti=imagedata
- Each individual file can be binary/text, compressed or not

The VTK Collection is in fact a generic holder for MultiBlock composite datasets which can store time information too.

The vtkXMLReader family is responsible for loading this kind of data.

User can use vtkXML_xxx_Writer to write N time steps of any kind of data and then add a little XML meta data to describe it.

- <VTKFile type="Collection" version="0.1" byte_order="LittleEndian">
-   <Collection>
-     <DataSet timestep="0.01" group="" part="0" file="Foo_001.vtu"/>
-     <DataSet timestep="0.02" group="" part="0" file="Foo_002.vtu"/>
-     <DataSet timestep="0.03" group="" part="0" file="Foo_003.vtu"/>
-   </Collection>
- </VTKFile>

VTK file

- VTK Simple Legacy Format
  - ASCII or binary
  - Supports all VTK grid types
  - Easiest for data conversion

- *Note: use VTK XML format for parallel I/O*

Numpy to VTK

VTK library

- Convert numpy array to VTK object

Compute with numpy

import numpy as np

data = np.genfromtxt("data.csv",   dtype=None, names=True,   delimiter=',', autostrip=True)

for name **in** data.dtype.names:

- array = data[name]
- output.RowData.append(array, name)

Python with Paraview

Python with Paraview

- Standard python interpreter (***python***)
    - Set *PYTHON_PATH* to directory containing ParaView modules
    - Import relevant ParaView modules
- ParaView's python client (***pvpython***)
    - Python interpreter with ParaView initialization plus sets the path to ParaView modules
- ParaView's batch client (***pvbatch***)
    - Same as *pvpython* without remote server connection capabilities
    - Can be run in parallel (using *mpirun* etc.)
- ParaView GUI (***paraview***)
    - GUI provides python shell comparable to *pvpython*
- Python for data processing
    - Python Programmable filter

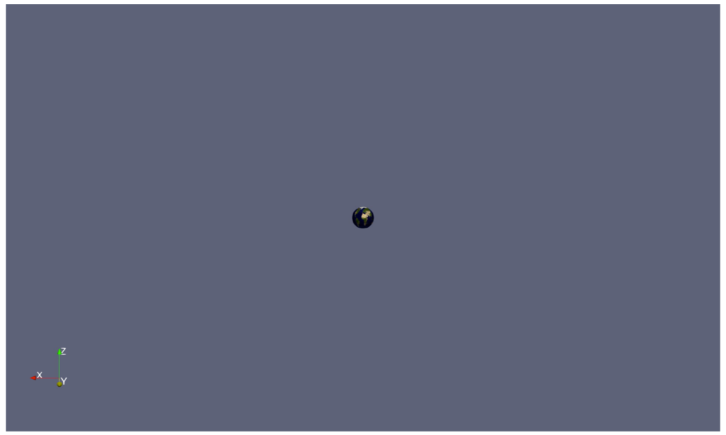
Programmable Source

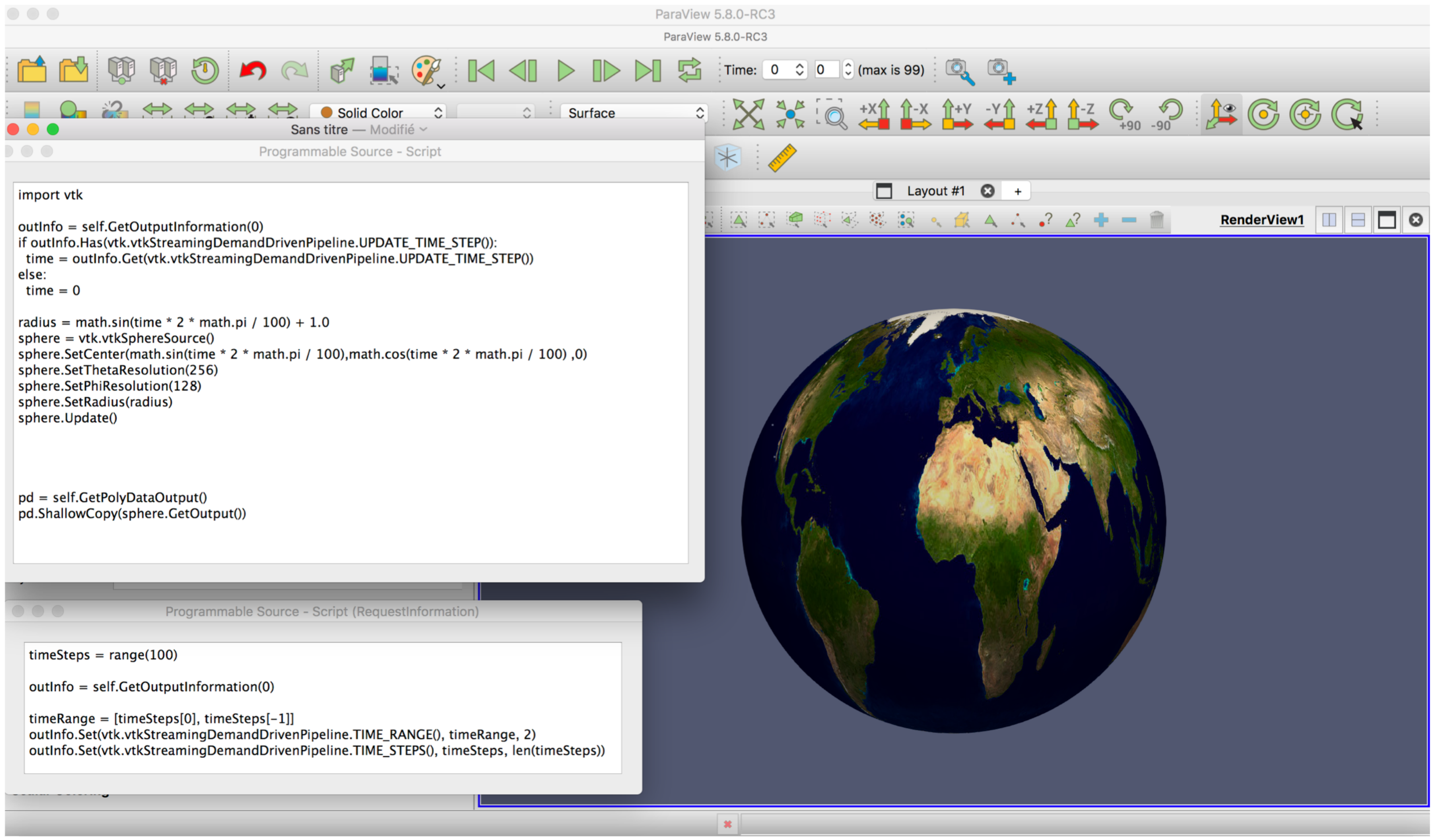
Script

- Use VTK library with phyton

Request

- Information on time grid





Programmable Source - Script

```
import vtk

outInfo = self.GetOutputInformation(0)
if outInfo.Has(vtk.vtkStreamingDemandDrivenPipeline.UPDATE_TIME_STEP()):
  time = outInfo.Get(vtk.vtkStreamingDemandDrivenPipeline.UPDATE_TIME_STEP())
else:
  time = 0

radius = math.sin(time * 2 * math.pi / 100) + 1.0
sphere = vtk.vtkSphereSource()
sphere.SetCenter(math.sin(time * 2 * math.pi / 100),math.cos(time * 2 * math.pi / 100) ,0)
sphere.SetThetaResolution(256)
sphere.SetPhiResolution(128)
sphere.SetRadius(radius)
sphere.Update()



pd = self.GetPolyDataOutput()
pd.ShallowCopy(sphere.GetOutput())
```

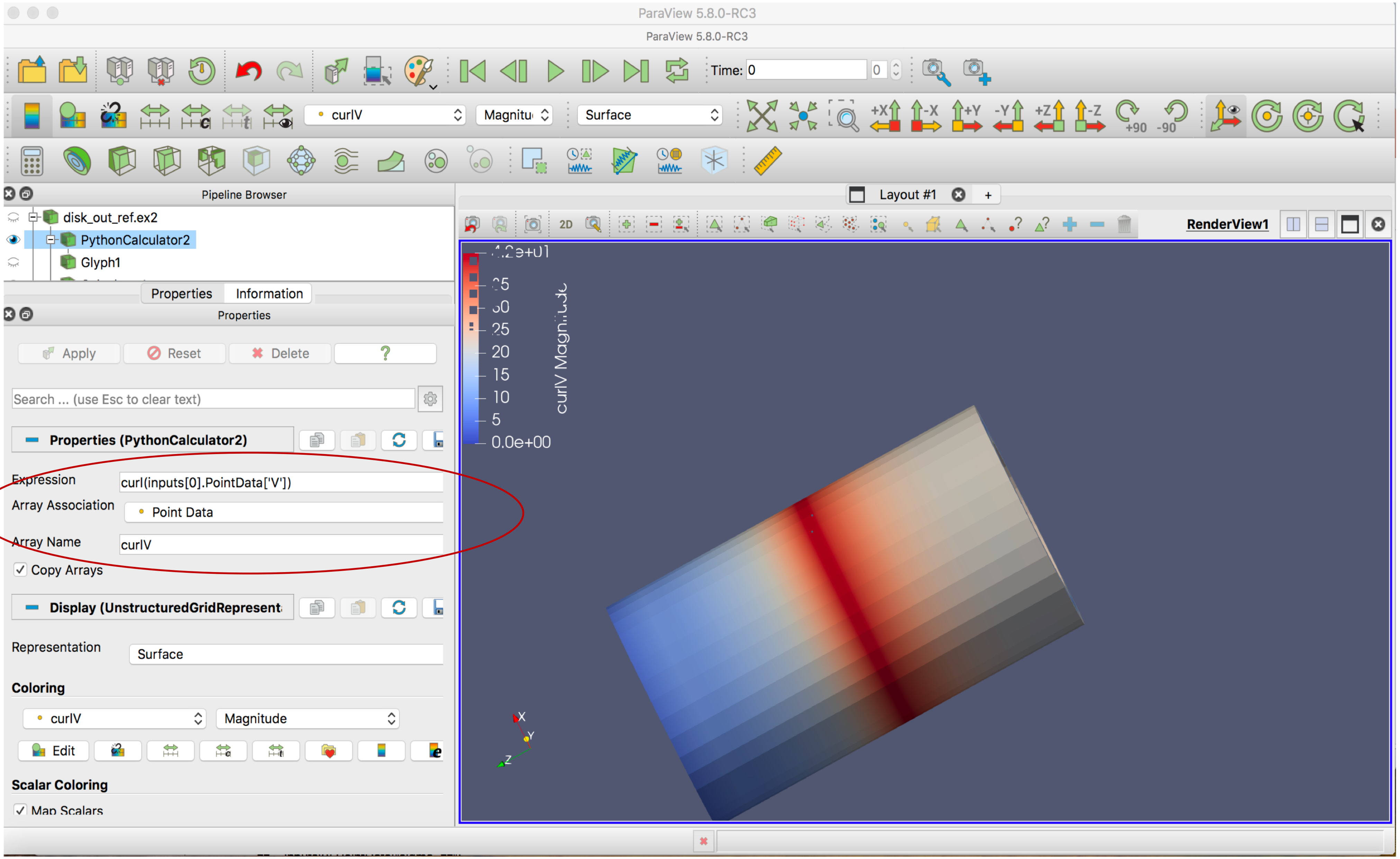
Programmable Source - Script (RequestInformation)

```
timeSteps = range(100)

outInfo = self.GetOutputInformation(0)

timeRange = [timeSteps[0], timeSteps[-1]]
outInfo.Set(vtk.vtkStreamingDemandDrivenPipeline.TIME_RANGE(), timeRange, 2)
outInfo.Set(vtk.vtkStreamingDemandDrivenPipeline.TIME_STEPS(), timeSteps, len(timeSteps))
```

Python calculator (simple)



Expression: curl(inputs[0].PointData['V'])

Array Association: Point Data

Array Name: curlV

✓ Copy Arrays

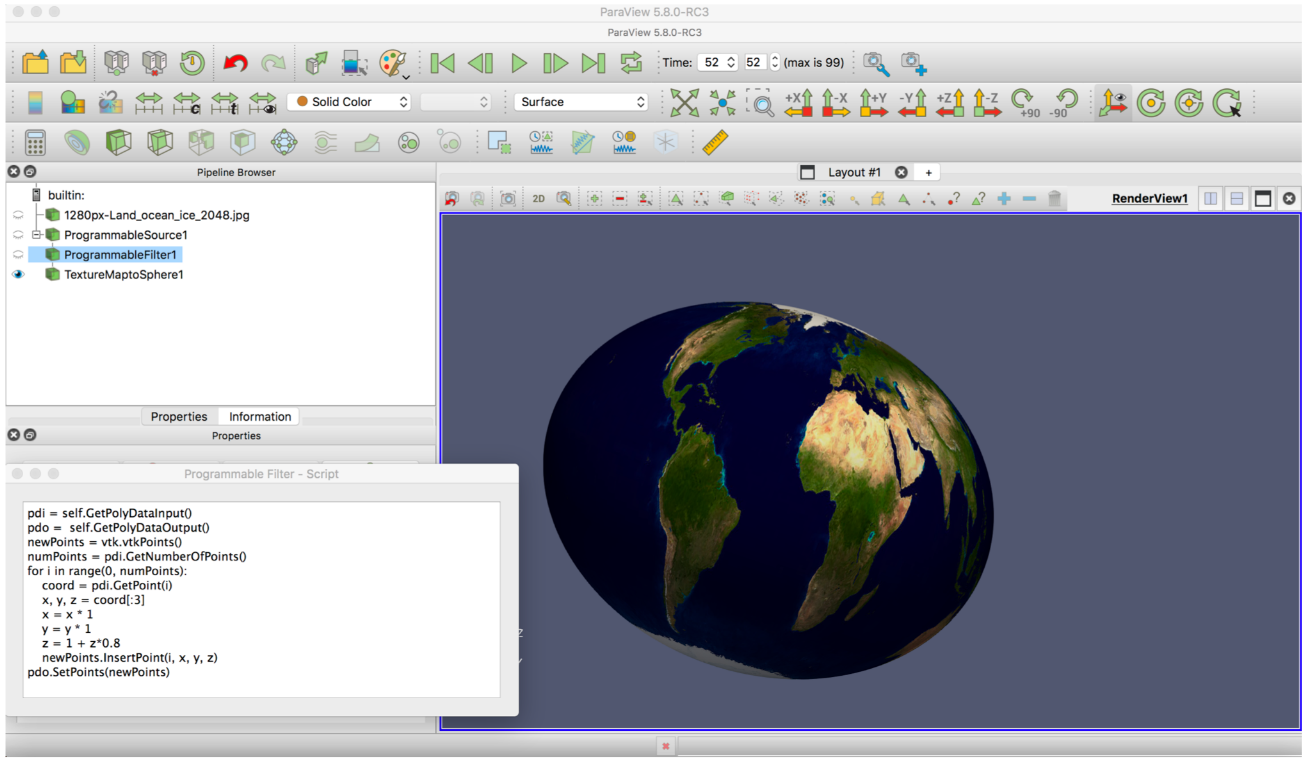
Programmable filter

Script

- Build a new filter applied to the input data

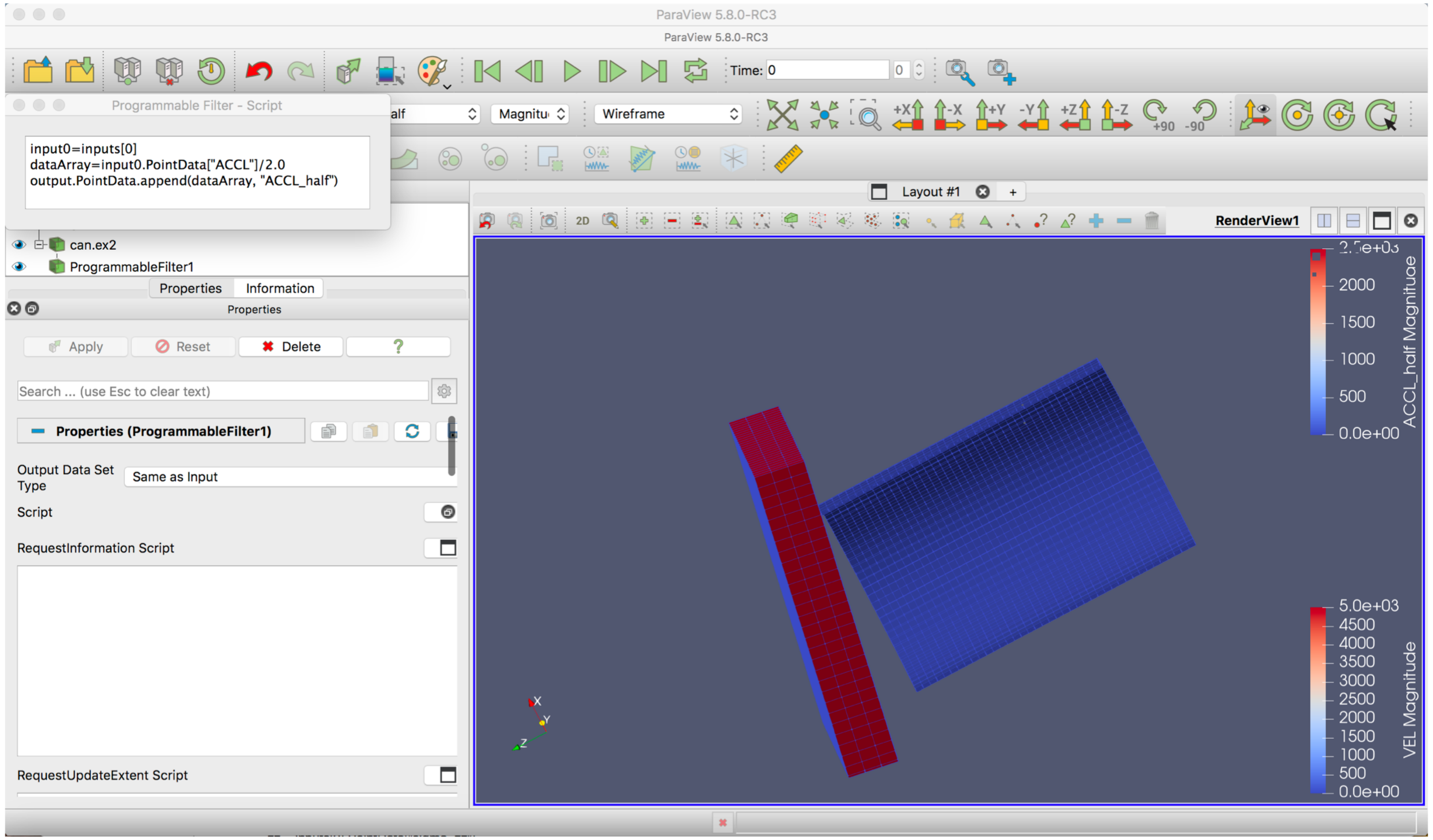
Request Information

Request Update Extent



```
pdi = self.GetPolyDataInput()
pdo =  self.GetPolyDataOutput()
newPoints = vtk.vtkPoints()
numPoints = pdi.GetNumberOfPoints()
for i in range(0, numPoints):
    coord = pdi.GetPoint(i)
    x, y, z = coord[:3]
    x = x * 1
    y = y * 1
    z = 1 + z*0.8
    newPoints.InsertPoint(i, x, y, z)
pdo.SetPoints(newPoints)
```

Programmable filter (2)



```
input0=inputs[0]
dataArray=input0.PointData["ACCL"]/2.0
output.PointData.append(dataArray, "ACCL_half")
```

Output Data Set Type: Same as Input

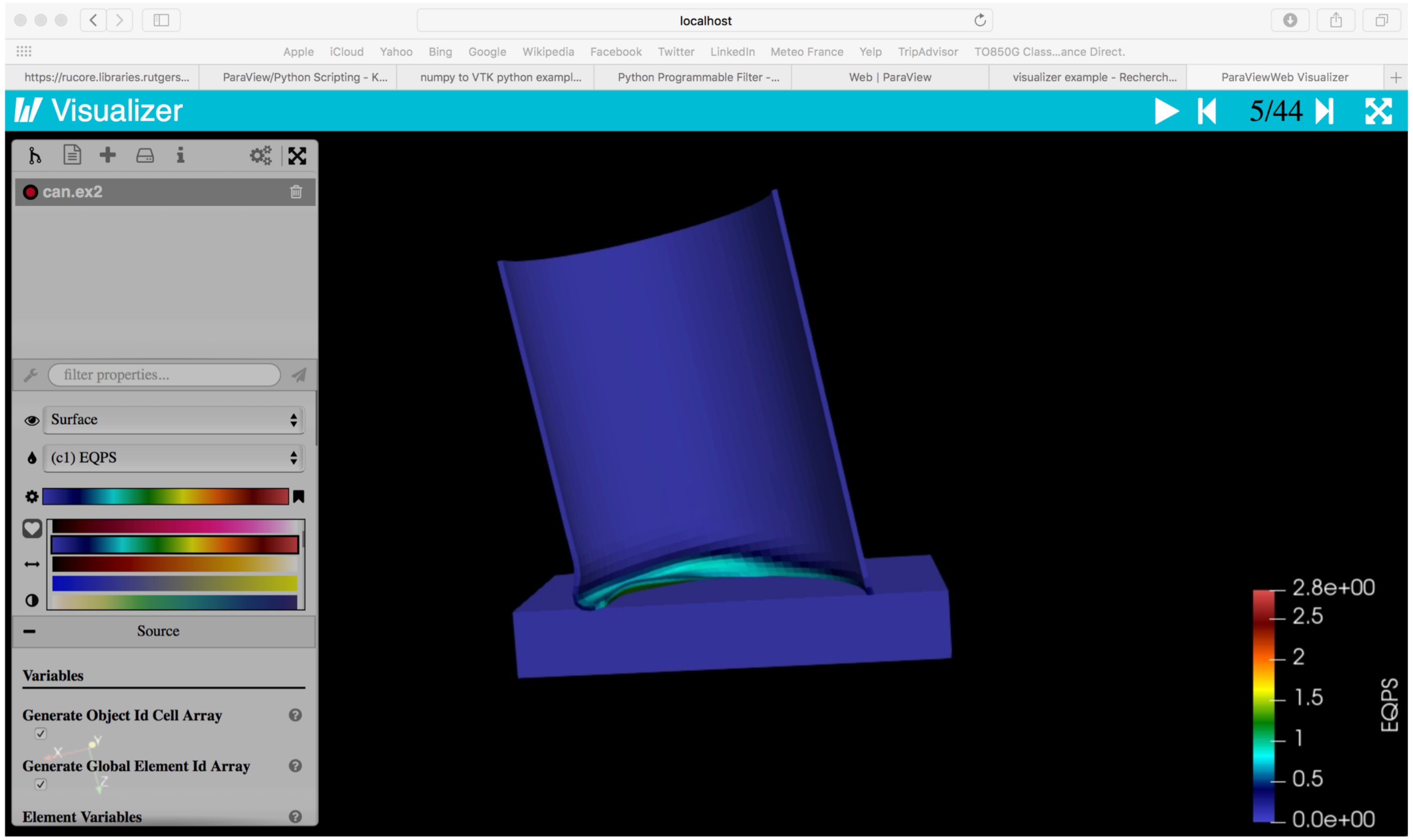
Paraview web with Visualizer

```
cd Paraview_folder/Contents/
```

```
./bin/pvpython                \
./Resources/web/visualizer/server/pvw-visualizer.py   \
--content ./Resources/web/visualizer/www/    \
--data /Applications/ParaView-5.8.0-
RC3.app/Contents/examples/                    \
--port 8080
```

Open browser:

- http://localhost:8080


localhost
Apple iCloud Yahoo Bing Google Wikipedia Facebook Twitter LinkedIn Meteo France Yelp TripAdvisor TO850G Class...ance Direct.
https://rucore.libraries.rutgers...
ParaView/Python Scripting - K...
numpy to VTK python exampl...
Python Programmable Filter -...
Web | ParaView
visualizer example - Recherch...
ParaViewWeb Visualizer
Visualizer
5/44
can.ex2
filter properties...
Source
2.8e+00
2.5
2
1.5
1
0.5
0.0e+00
EQPS

Surface

(c1) EQPS

Variables

Generate Object Id Cell Array ✓

Generate Global Element Id Array ✓

Element Variables