ParaView Documentation

Mehrdad Yousefi (CCIT Visualization Group)

May 24, 2017

Abstract

In this documentation, we will provide some basic and intermediate information about visualization toolkits by using ParaView like slicing, threshold value, glyph and streamline representations of vectorial fields and create high resolution animation from visualization framework and then we will focus on how to run ParaView in a server/client mode on Clemson University Palmetto cluster. Also some basic scripting information to create programmable visualization filter by using C++ or Python programming languages will be provided.

Introduction

ParaView is a powerful visualization framework which is built based on VTK visualization toolkit from Kitware company. It has some unique features to create robust visualization from scientific data like slicing and streamline representation of vectorial fields. Also because it is built based on VTK, it is possible to create user defined filters by using C++ and Python programming languages. Also it can be used for big data visualization by using high preformance computing (HPC) facilities like Palmetto cluster which is useful for parallel visualization technique. In this documentation, all those mentioned features of ParaView will be covered in order to help students and faculties of Clemson University to create more powerful visualizations from their scientific data.

Downloading and installation of ParaView

You could download the open source ParaView software from official ParaView website and the recommended version for all Clemson University community is 5.0.1 and it should be built from source in order to have compliance with Palmetto cluster. But if you want to just use ParaView in client mode you could download and install any pre-compiled version of ParaView according to your expectations.

Pre-defined filters in ParaView to create visualization from scientific data

ParaView can support a lot of output file formats of simulation and visualization packages like VTK, VMTK, ITK, VisIt, NASTRAN, ANSYS, LAMMPS and etc. So we could say that ParaView can cover a broad range scientific simulation outputs. In order to open and load your raw data in ParaView, you could use main menu as File $\longrightarrow$ Open... and choose your file in order to load it in ParaView. After opening of your file you should push the apply button in order to load it in ParaView. If you do these steps successfully, you should see this loaded structure as is shown in Fig. 1.

Now we could use several pre-defined filters in ParaView to create a better visualization from our raw data. First filter that will be introduced here is Contour filter which you can choose this from Filters $\longrightarrow$ Alphabetical $\longrightarrow$ Contour. This filter can extract the isosurfaces or isocurves of a scalar field. You could define your iso value and then it will show you the related isosurface. Also it is possbile to choose several iso values and create several isosurfaces with multi color visualization. The choosing iso value does not have any pre-defined rule and it depends on user needs but for many data structure half maximum of scalar field can show some useful information about isosurfaces. In the loaded data structure we choosed half maximum of scalar field as iso value and then by apply the Contour filter you could see the visualized isosurface which is related to our iso value as is shown in Fig. 2.

Now according to Fig. 2, it seems the loaded data structure is related to medical imagin of human skull and scalar field is the pixel values which can

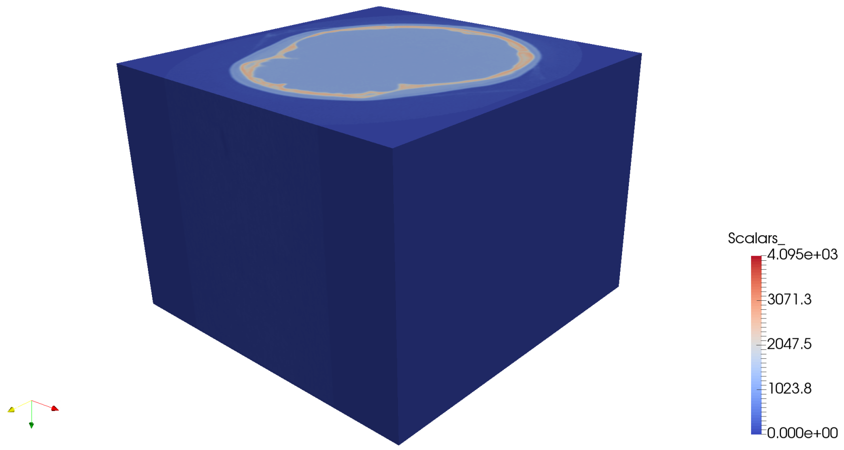

Figure 1: Loaded structured grid data to the ParaView main screen.

define different isosurfaces. The next pre-defined filter is Clip which can clip your data structure to see the values of your field in the depth of your data structure. You could choose the Clip by using Filters ⟶ Alphabetical ⟶ Clip. In the Clip filter you could define the point and normal vector of the plane which you are interested to use as clipper tool and then by apply this filter to the last Contour filter you should see the clipped data structure which is shown in Fig. 3.

The next filter which is very similar to Clip filter is Slice. As its name, it can slice your data structure according to your pre-defined plane. The input for this filter is the normal vector and point of slicer plane. You could choose this filter from Filters ⟶ Alphabetical ⟶ Slice. If you apply this filter to the original data structure you could see this which is shown in Fig. 4.

ParaView has a lot of powerful data analysis tools which can help scientists to create better visualization from their data structures. For example you could plot your field in a user defined line or over simulation time at a specified location. For example if you want to plot the scalar field of this loaded data structure in a pre-defined line you could choose data analysis filter from Filters ⟶ Data Analysis ⟶ Plot Over Line. In Fig. 5 the user defined line and one dimensional plot over that line for the scalar field are shown.

Now we learned the basic of ParaView pre-defined filters and now we could move on to vectorial field visualizations. In comparison to scalar fields, vectorial fields are more complex but because of their directional dependent nature we could extract more information from them to create more powerful visualization. The first vectorial filter which can be used to visualize the magnitude and direction of vectors is Glyph. Glyph filter can visualize the vectorial field as arrows to show the direction, density and magnitude of vectorial fields. This vectorial field can be velocity, electrical or magnetic fields or etc. You could choose Glyph filter from Filters ⟶ Alphabetical ⟶ Glyph and then you just define your arrow size in scaling option and then just apply that filter. So you could see the visualized glyphs as are shown in Fig. 6.

The next powerful filter for vectorial field visualization is Stream Tracer

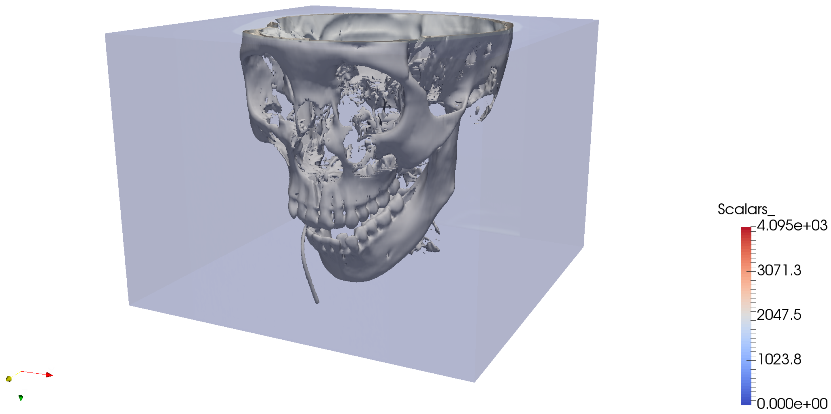

Figure 2: Isosurface which is created from loaded data structure by using Contour pre-defined filter.

which can be choosen from Filters ⟶ Alphabetical ⟶ Stream Tracer. Stream Tracer can visualize the vectorial field as streamlines in order to show the direction, density and magnitude of vectorial fields. It is important to specify the integration method over streamlines and also specify the number of seed points and the radius of seed points distribution. Finally you could find the visualized streamlines for this vectorial field as is shown in Fig. 7.

Sometimes in order to create a better visualization from vectorial fields like displacements or velocities, it is useful to visualize the displaced data structure according to that vectorial field. It's a powerful technique in structural mechanics simulation to visualize the defomred structures. ParaView has a unique filter which is called Warp By Vector to visualize the deformed structure according to a vectorial field. You could choose this filter from Filters ⟶ Alphabetical ⟶ Warp By Vector. Also you could define the scale displacement to tune your visualization. For example you could deformed structure which is created by using Warp By Vector filter as is shown in Fig. 8.

In order to show the powerful visualization features of ParaView to visualize time-dependent simulation fields, we toke some output files from ParaView workshop which is held at Purdue University on October, 26, 2016. These materials are distributed from works of Nicolas Guarin-Zapta and they are under the Creative Commons Attribution 4.0 license and Creative Commons Attribution 2.5 Generic by Kitware. First example is related to plastic deformation of a cylinder under non-symmetrical pressure. You can see the two screen shots of the deformation simulation on different simulation times as are shown in Fig. 9.

Another example which is provided in order to summarize the pre-defined filters is the chemical reactor simulation. In this simulation temperature, velocity of fluid, pressure and gas species concentrations are claculated by computational methods and you can see a comprehensive visualization as are shown in Fig. 10.

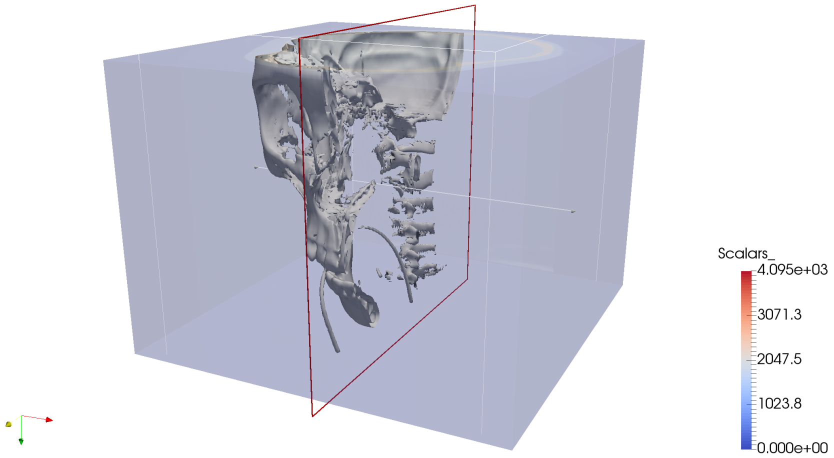

Figure 3: Clipped data structure which is created from loaded data structure by using Clip pre-defined filter.

Python scripting for ParaView by using VTK visualization toolkit

As we said before, ParaView is built on VTK visualization toolkit and it gives us some powerful and unique features to create programmable filters by using Python or C++ programming language. In order to show some intermediate visualization techniques based on Python/VTK, I wrote this code to create a generalized cylinder based on a numerical curve by VtkTubeFilter. You may find the script as Listing 1.

Listing 1: Python code to visualize generalized cylinder based on numerical curves and VtkTubeFilter

```
import vtk
import math
from numpy import genfromtxt
centerline = genfromtxt('centerlines.csv', delimiter=',')
npts       = 5901
vtkPoints = vtk.vtkPoints()
vtkPoints.SetNumberOfPoints(5901)
for i in range (npts):
x = centerline[i,0]
y = centerline[i,1]
z = centerline[i,2]
vtkPoints.SetPoint(i, (x,y,z))
vtkCellArray = vtk.vtkCellArray()
vtkCellArray.InsertNextCell(npts)
for i in range(npts):
vtkCellArray.InsertCellPoint(i)
value = lambda i: math.fabs(math.sin(math.pi*i/30.))
vtkFloatArray = vtk.vtkFloatArray()
```

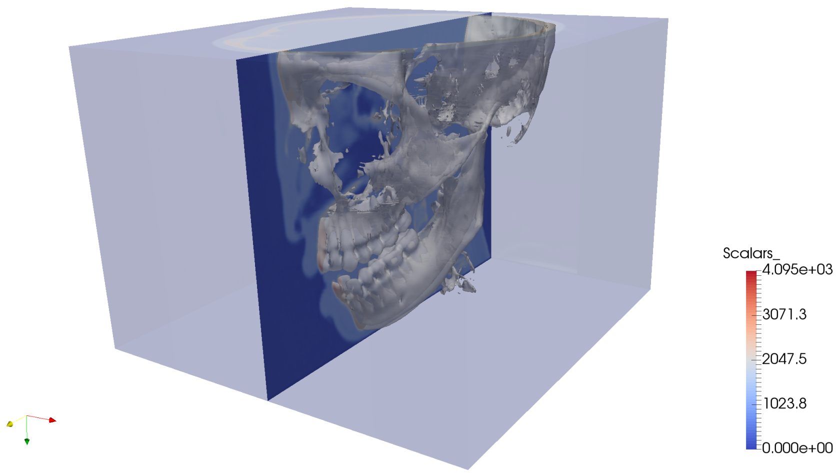

Figure 4: Sliced data structure which is created from loaded data structure by using Slice pre-defined filter.

```
vtkFloatArray.SetNumberOfValues(npts)
for i in range(npts):
vtkFloatArray.SetValue(i, value(i))
vtkPolyData = vtk.vtkPolyData()
vtkPolyData.SetPoints(vtkPoints)
vtkPolyData.SetLines(vtkCellArray)
vtkPolyData.GetPointData().SetScalars(vtkFloatArray)
vtkSplineFilter = vtk.vtkSplineFilter()
vtkSplineFilter.SetInputData(vtkPolyData)
vtkSplineFilter.SetNumberOfSubdivisions(5*npts)
vtkSplineFilter.Update()
vtkTubeFilter = vtk.vtkTubeFilter()
vtkTubeFilter.SetInputConnection(vtkSplineFilter.GetOutputPort())
vtkTubeFilter.SetRadius(0.5)
vtkTubeFilter.SetNumberOfSides(20)
vtkTubeFilter.CappingOn()
vtkPolyDataMapper = vtk.vtkPolyDataMapper()
vtkPolyDataMapper.SetInputConnection(vtkTubeFilter.GetOutputPort())
vtkActor = vtk.vtkActor()
vtkActor.SetMapper(vtkPolyDataMapper)
vtkRenderer = vtk.vtkRenderer()
vtkRenderer.AddActor(vtkActor)
vtkRenderWindow = vtk.vtkRenderWindow()
vtkRenderWindow.AddRenderer(vtkRenderer)
vtkRenderWindow.Render()
vtkRenderWindowInteractor = vtk.vtkRenderWindowInteractor()
vtkRenderWindowInteractor.SetRenderWindow(vtkRenderWindow)
vtkRenderWindowInteractor.Initialize()
vtkRenderWindowInteractor.Start()
```

The Listing 1 code can take a numerical curve which is defined as centerlines

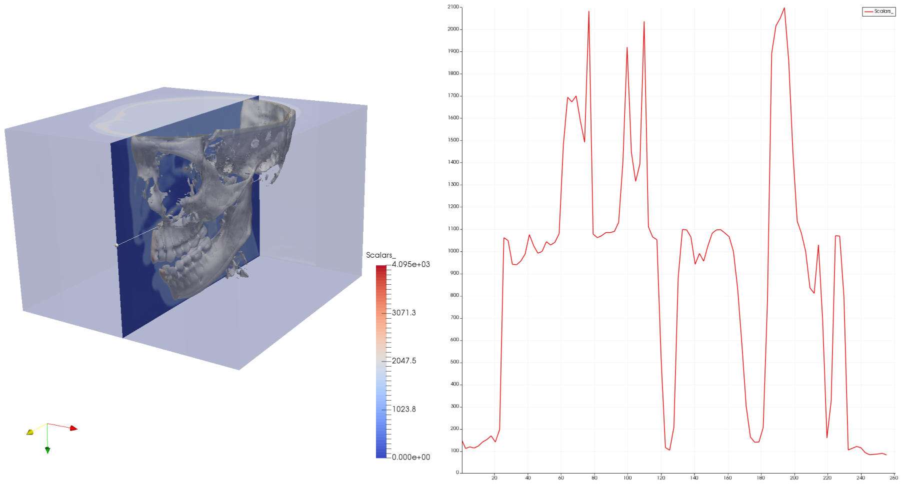

Figure 5: Plot over line of scalar field which is created from loaded data structure by using Plot Over Line pre-defined filter.

and then convert this centerline to generalized cylinder by using VtkTubeFilter and then assign color bars by user defined numerical distribution and visualize it as is shown in Fig. 11.

Running ParaView on Clemson University Palmetto cluster

As we said before, currently we have ParaView 5.0.1 on Palmetto cluster as a pre-compiled module and you could add it to your workstation by using module add paraview/5.0. Sometimes we need to visualize big data structures by using high performance computing facilities (HPC) in the parallel mode by using message passing interface (MPI) or NVIDIA GPU computing. Currently, Clemson University students and faculties can use ParaView in a server/client mode by following these steps:

1. In terminal 1, you should login as a X server user to Palmetto cluster and request a job with multi-threads and one GPU and specific memory and then add ParaView module and then run the ParaView server as:
   - ssh -X $user@user.palmetto.clemson.edu
   - qsub -I -X -l select=1:ncpus= $threads:mpiprocs= $threads:ngpus=1:mem=120gb:interconnect=fdr,walltime= $time
   - module add paraview/5.0
   - export DISPLAY=:0
   - mpirun -np $threads pvserver -display :0
2. In terminal 2, you should tunnel to the chunked node by using: ssh -L 11111:nodeWXYZ:11111 $user@user.palmetto.clemson.edu

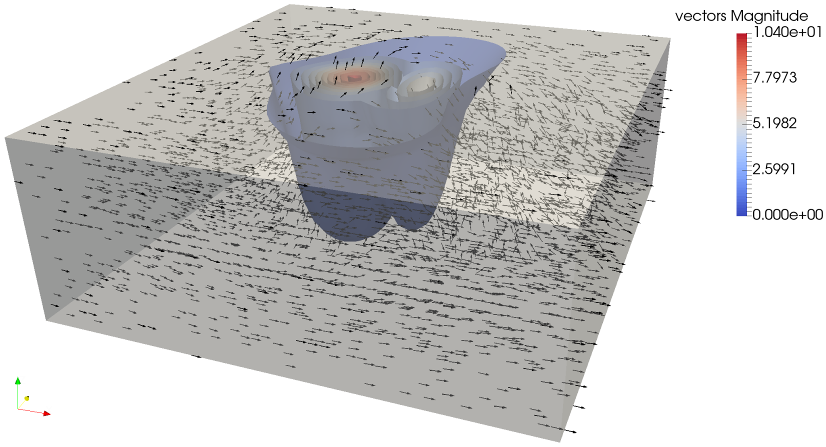

Figure 6: Glyph filter for vectorial field which is created from loaded data structure by using Glyph pre-defined filter.

3. In terminal 3, you should run ParaView by using: ./paraview

Now you could use parallel and CUDA GPU rendering to visualize your big data structure by using Palmetto cluster and one example which is taken from work of Mehrdad Yousefi at Dr. Schiller's research group is shown in Fig. 12.

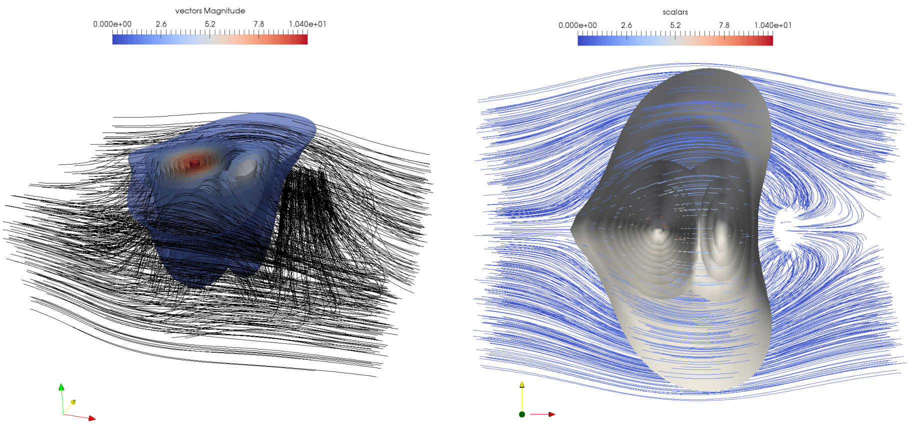

Figure 7: Stream Tracer filter for vectorial field which is created from loaded data structure by using Stream Tracer pre-defined filter.

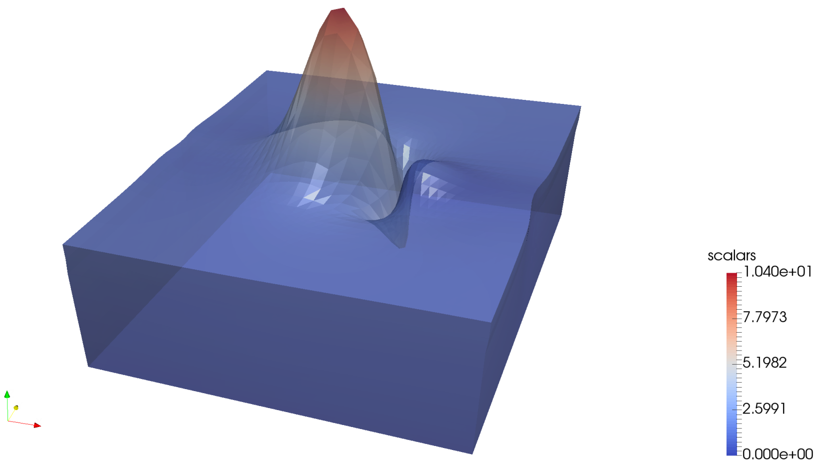

Figure 8: Warp By Vector filter for vectorial field which is created from loaded data structure by using Warp By Vector pre-defined filter.

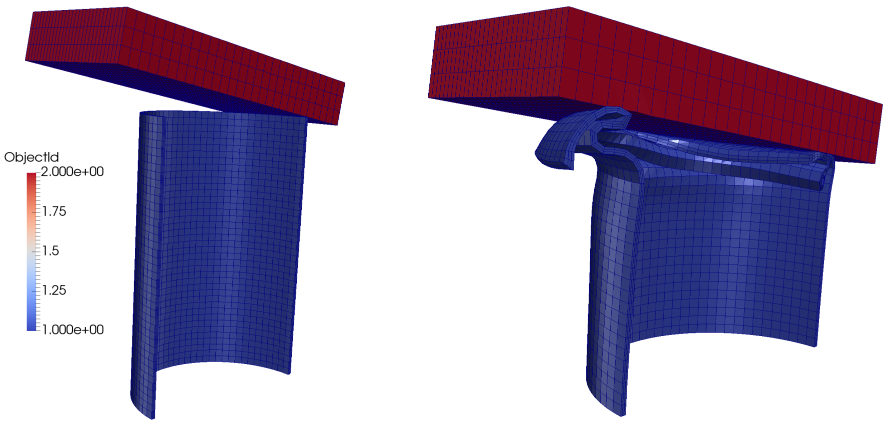

Figure 9: Plastic deformation of cylinder under non-symmetrical pressure at different simulation times.

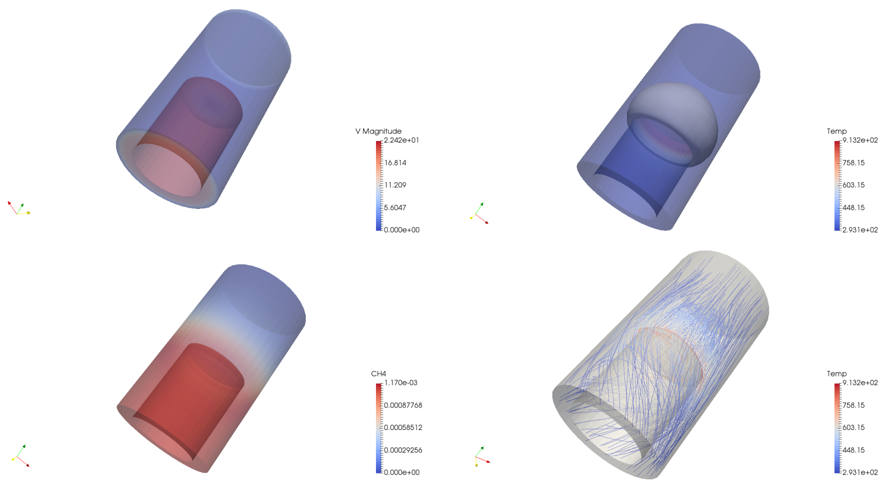

Figure 10: Comprehensive visualization of chemical reactor by using ParaView pre-defined filters.

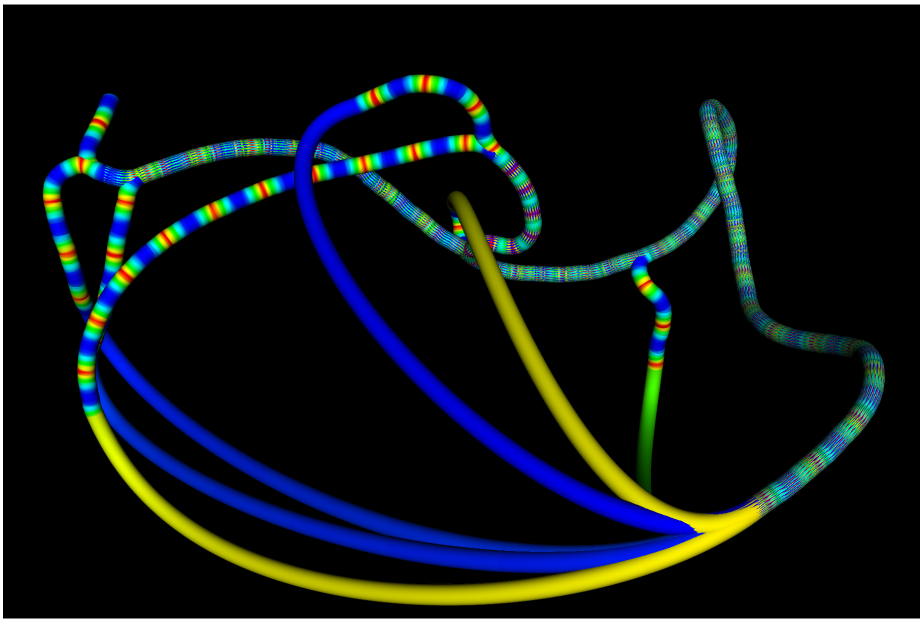

Figure 11: Python programmed generalized cylinder filter which is built based on VTK visualization toolkit.

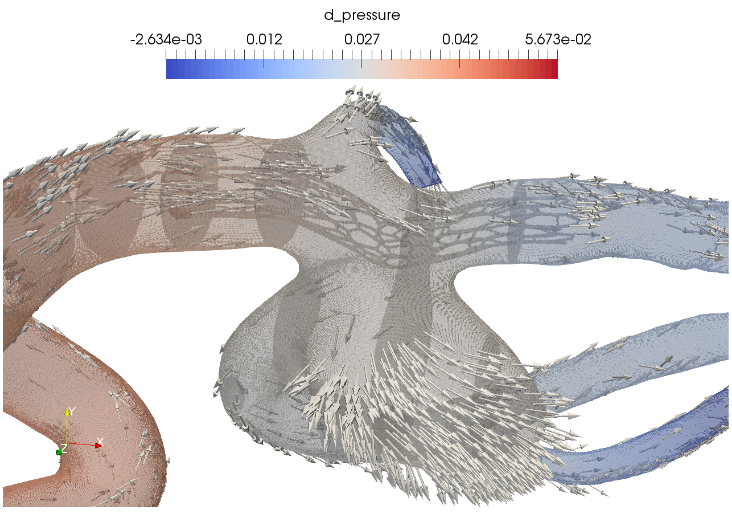

Figure 12: Three-dimensional fluid dynamics simulation in cerebrovascular human vessels which is taken from work of Mehrdad Yousefi at Dr. Schiller's research group.