CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Visualization of blood flow in an Aneurysm Simulation

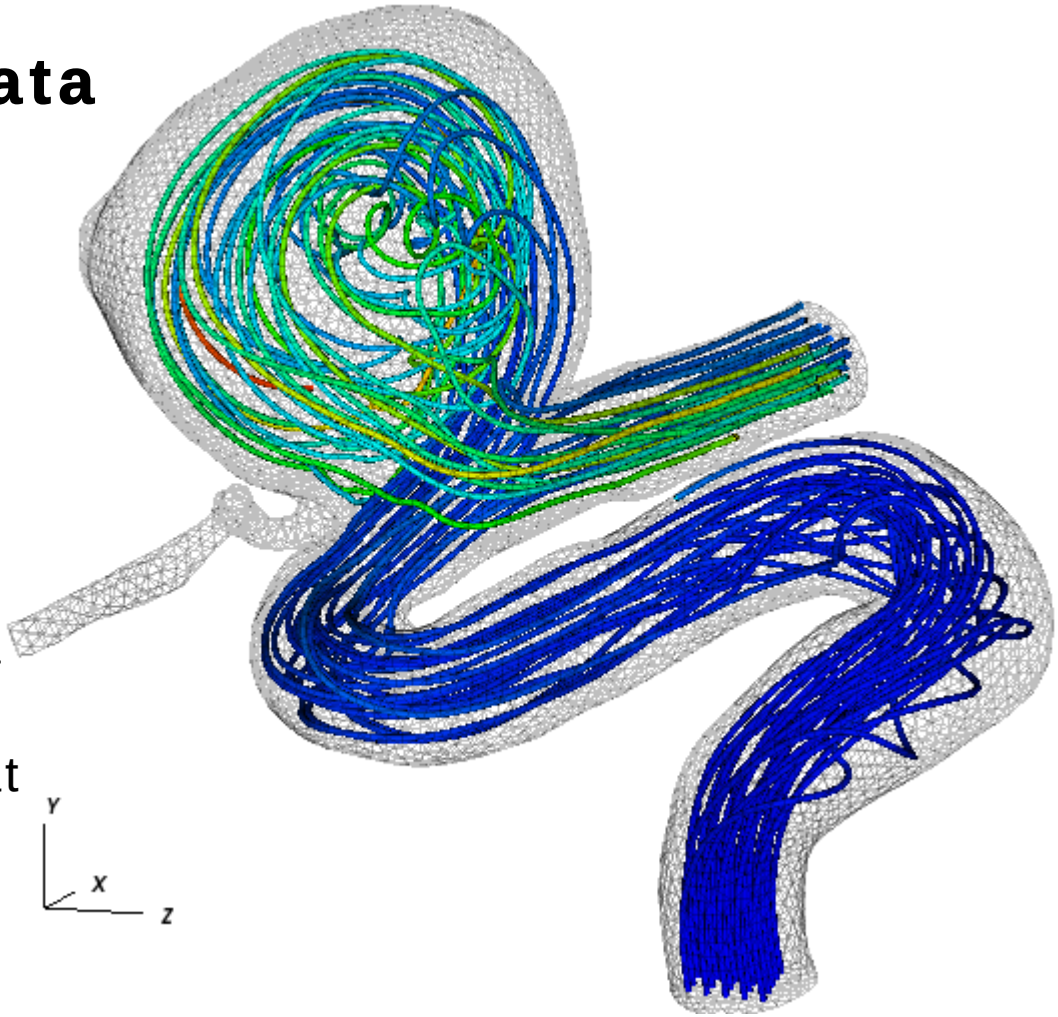
Jean M. Favre

Visualization Task Leader

November 2013

Outline

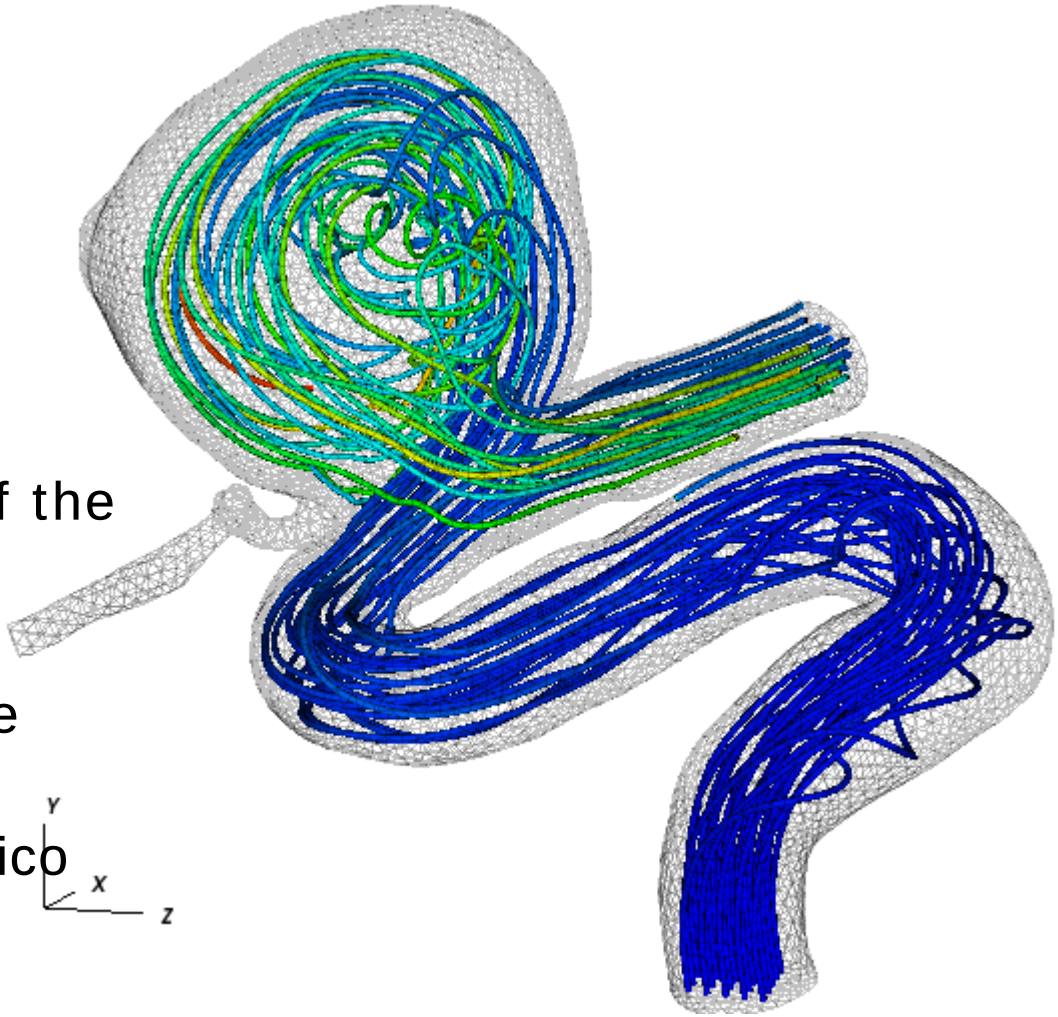
- **Description of the data**
- **Static Analysis**
  - Pressure display
  - Velocity glyphs
  - Picking
  - Streamlines
  - Flux thru a surface
- **Transient Analysis**
  - Multi-view of Transient data
  - Pressure Curve vs. Time
  - Average over one heart beat
  - Pathlines
  - Expressions
  - Python code
  - Making it pretty



CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

10 heart beats thru an aneurysm

- The simulation was run by Gilles Fourestey, on an IBM Blue Gene, at EPFL Switzerland.
- The solver is LifeV, a Finite Element library providing implementations of state of the art mathematical and numerical methods, a collaboration between École Polytechnique Fédérale de Lausanne (CMCS), Politecnico di Milano (MOX), INRIA (REO, ESTIME) and Emory University (Sc. Comp).



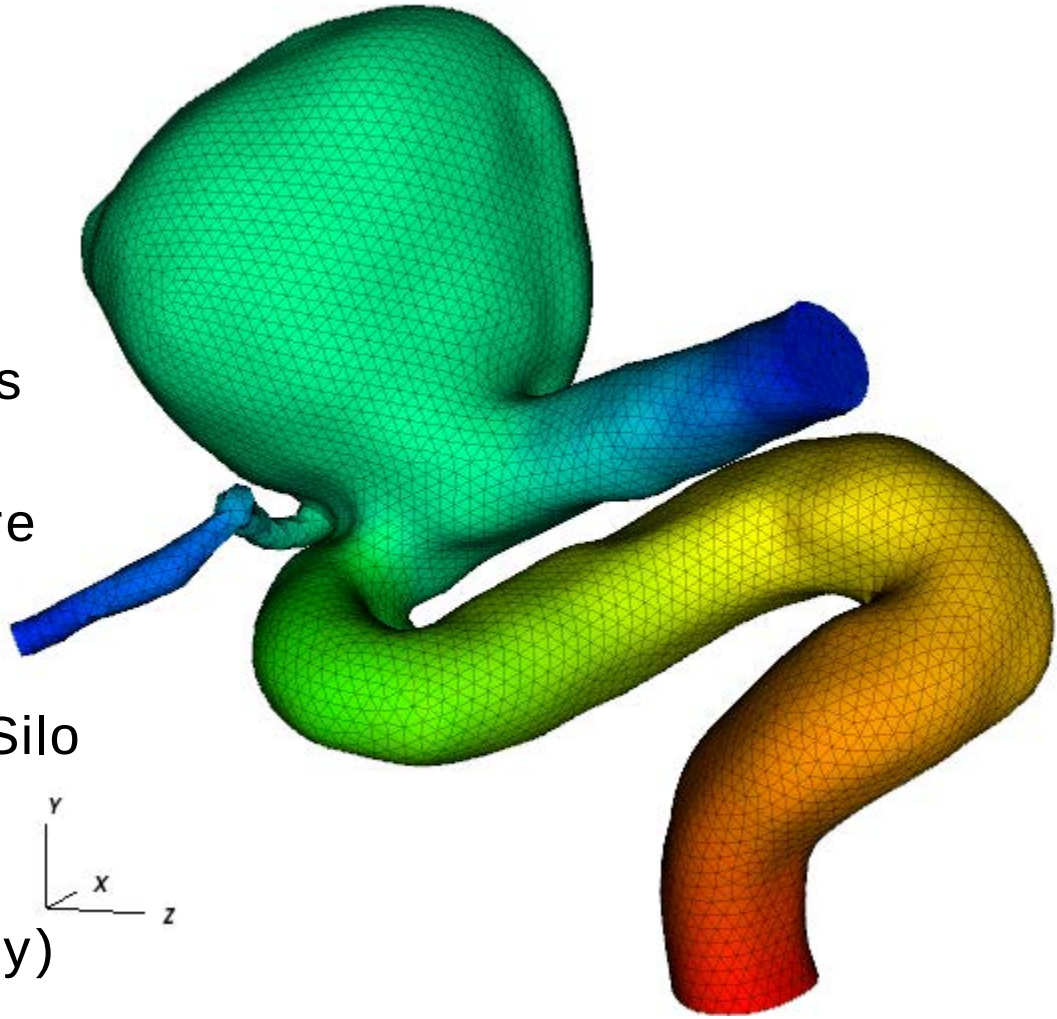
10 heart beats thru an aneurysm

For this tutorial, we use a simpler version of the mesh.

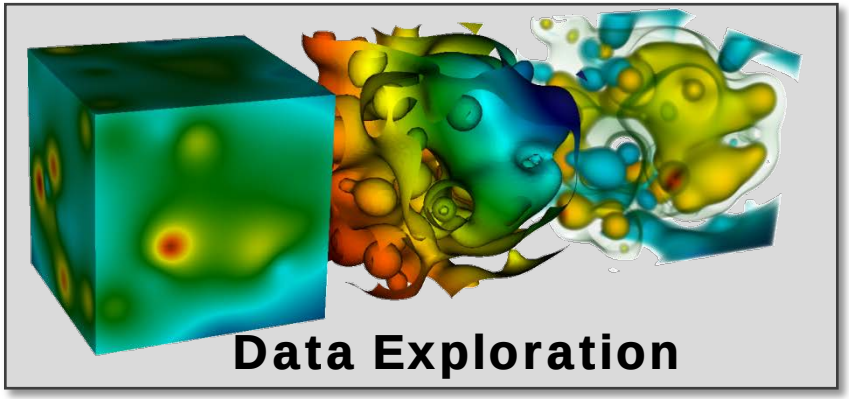
- 130754 tetrahedral cells
- 25656 nodes
- 200 out of 1467 time steps

"**pressure**" and "**velocity**" are available for each time step.

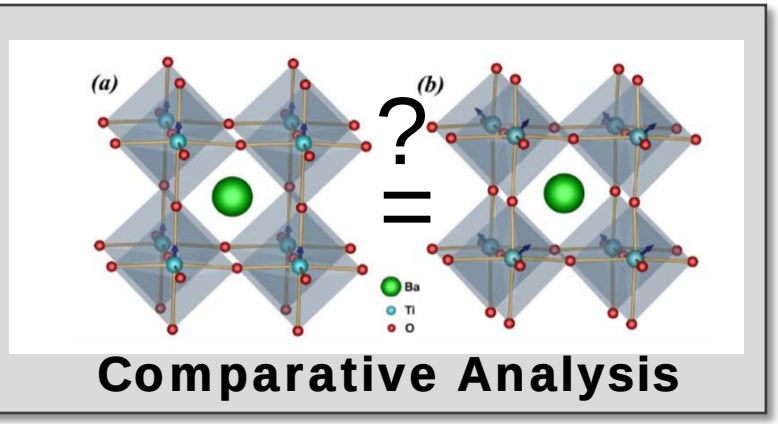
Each time step is stored in a Silo file format, and can be read stand-alone (i.e. each file contains the mesh connectivity)



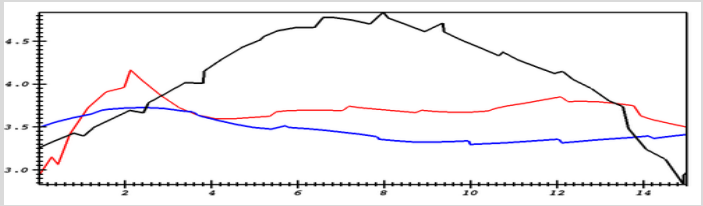
Visualization is many complementary things



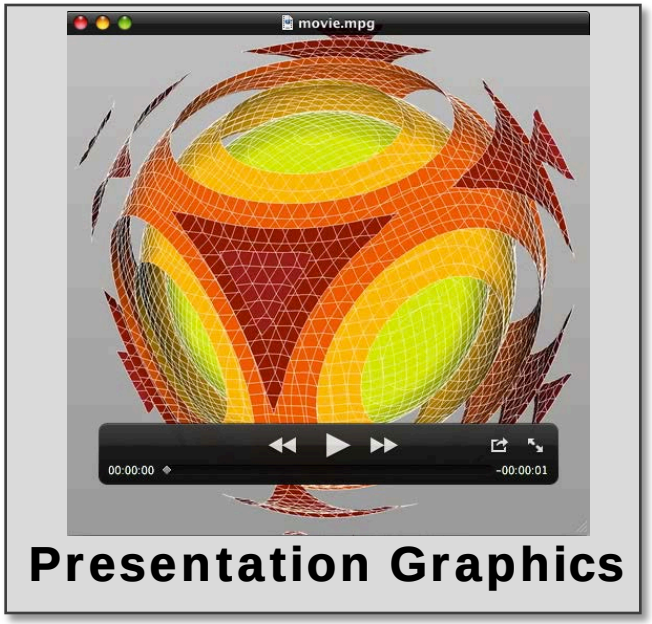
Data Exploration



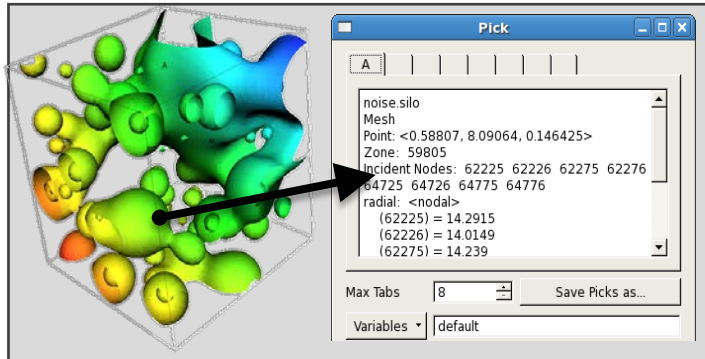
Comparative Analysis



Quantitative Analysis



Presentation Graphics



Visual Debugging

Static Analysis

Uses data from a single timestep

- Do multiple plots
- The plot will be updated when you change the timeslider

Plots

Example:

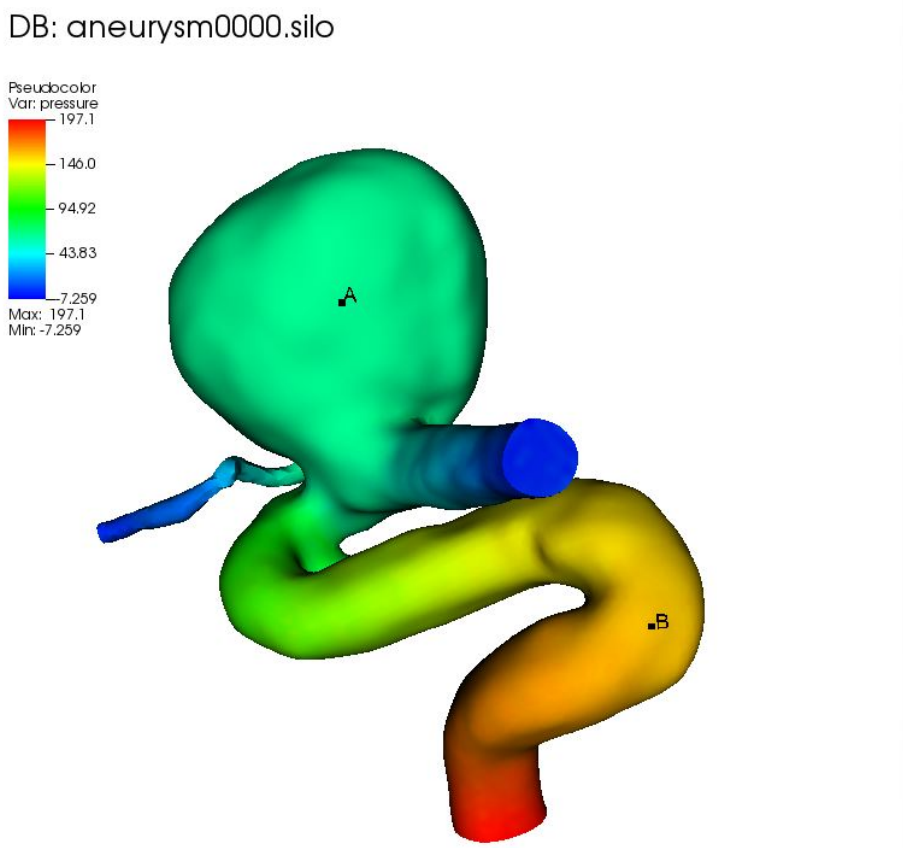
- **Pseudocolor Plots**
- **Vector Plots**

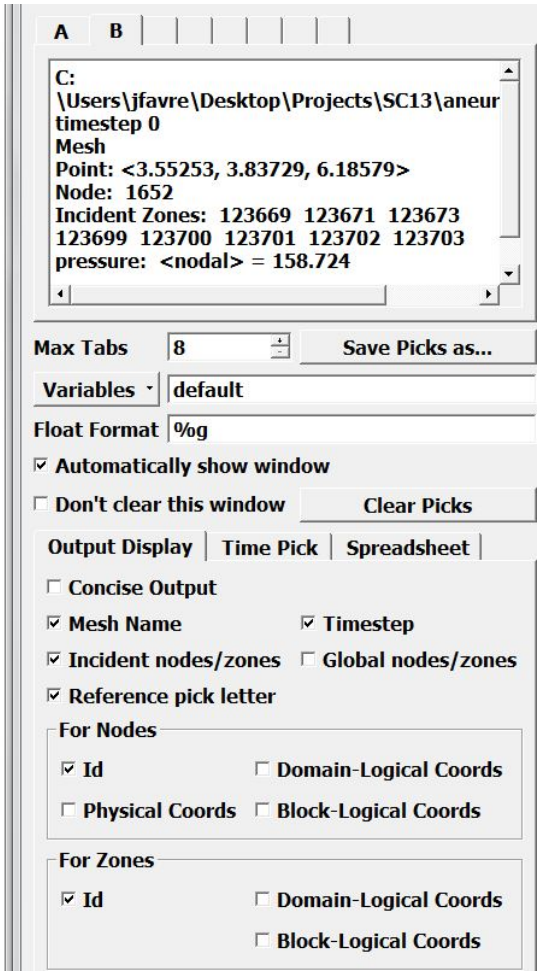
Plots will apply to the whole volumetric mesh, unless the domain is restricted , e.g., by applying an operator such as a slice cut, or an isosurface, etc.

The vector plot is a set of icons (glyphs) which is already limited to a fixed number to avoid visual clutter

Picking

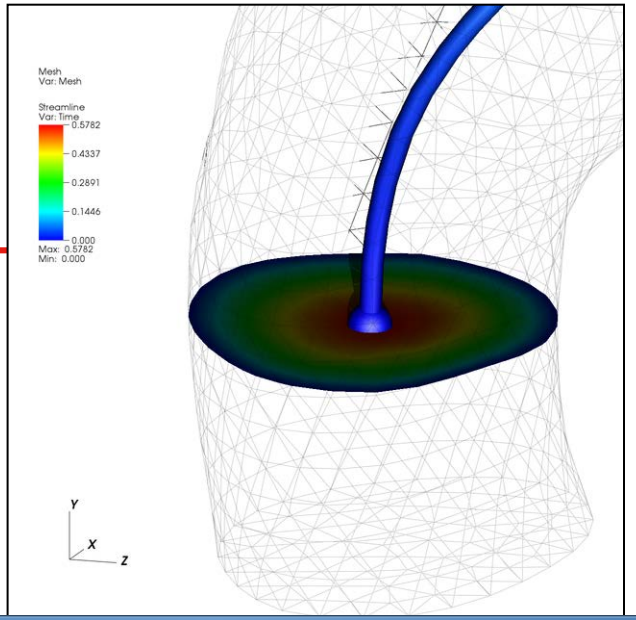
- The **Active** Object can be picked
- A rather verbatim output can be given; great for debugging





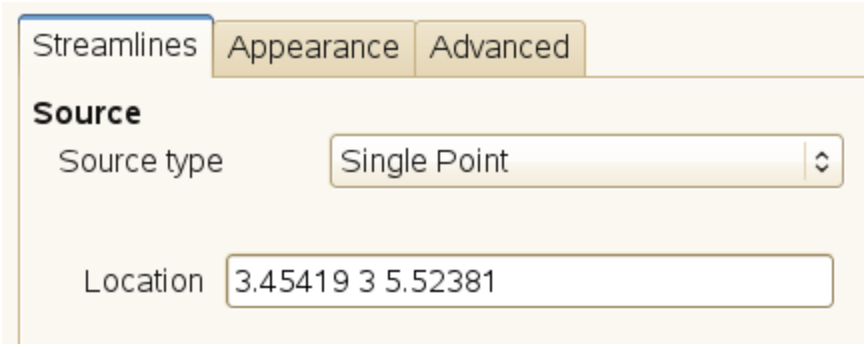
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

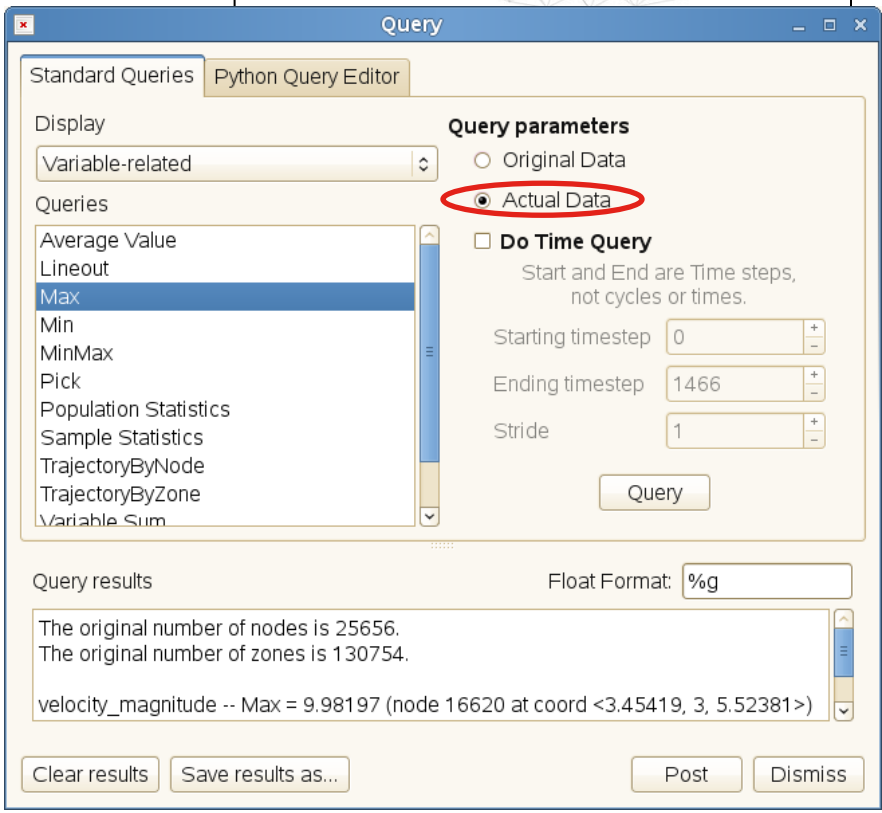
Streamlines (1)



Use a slice (intercept Y at (3,3,3) thru the mesh to:

1. **Position a single seed point near the inlet (upstream in the artery)**
2. **Select the seed point at the maximum of velocity_magnitude**
3. **Position a circle of seeds around that point**



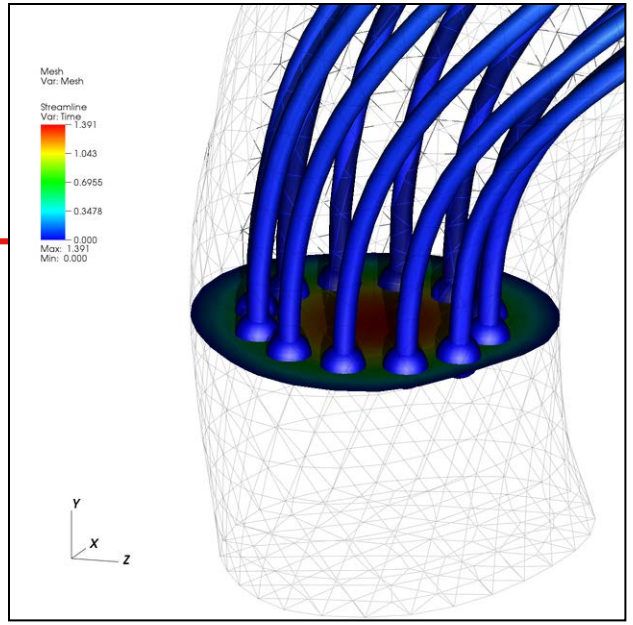


CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

Streamlines (2)

Use a slice thru the mesh to:

Position a circle of seeds around that point



Streamlines | Appearance | Advanced

Source

Source type: Circle

Origin: 3.45419 3 5.52381

Normal: 0 1 0

Up axis: 0 0 1

Radius: 0.12

Sampling

Sampling type: Uniform Random Samplir

Samples in Theta: 12

Streamlines | Appearance | Advanced

Coordinate transform

None Cylindrical to Cartesian Cartesian to Cylindrical

Phi scaling 1 (When displaying in cylindrical coordinates.)

Data

Data Value: Time

Limits Minimum 0 Maximum 1

Color

Color table: Default

Opacity: Fully Opaque

Display

Draw as: Tubes Display density: 10

Radius: 0.005 Fraction of Bounding Box

Vary radius

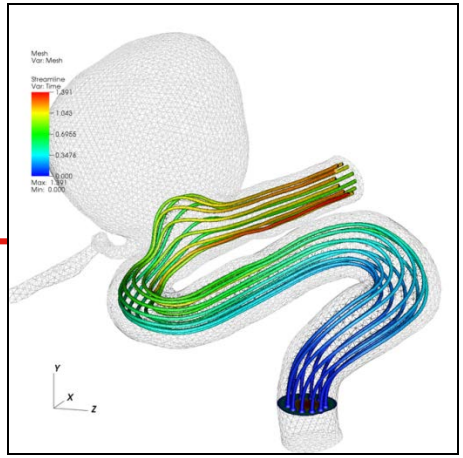
Show seeds Radius: 0.01 Fraction of Bounding Box

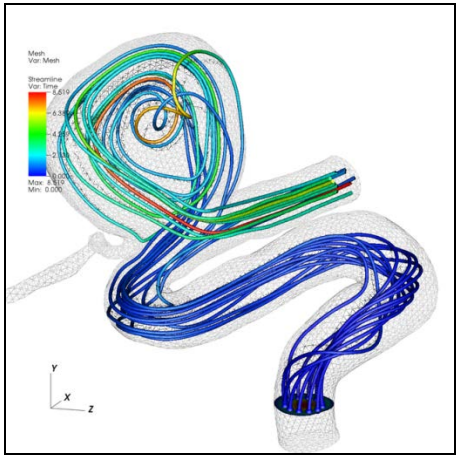
Show heads Display as: Sphere

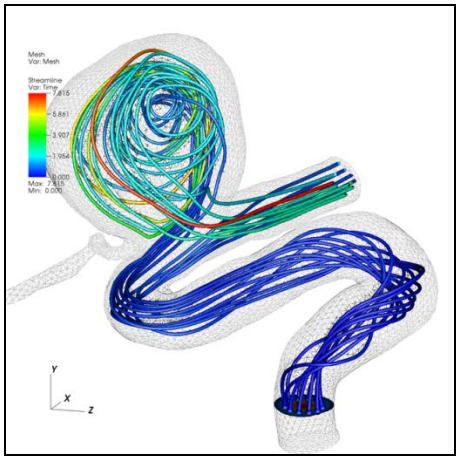
Streamlines (3)

Streamlines will be different at different instants of time.

Move the time slider and show t=0, t=99, t=199







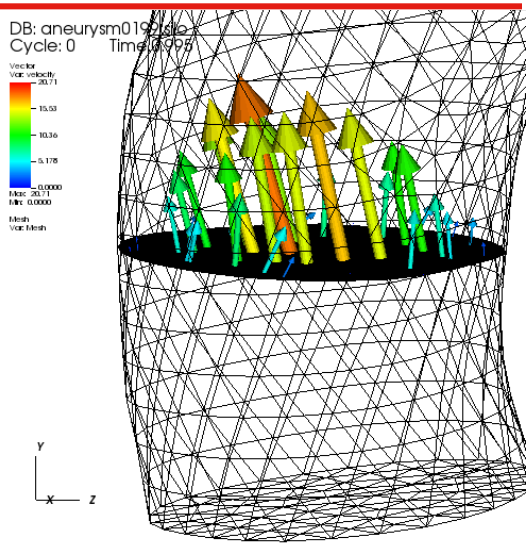
Flux through a surface (1)

Take the same slice cut at the inlet

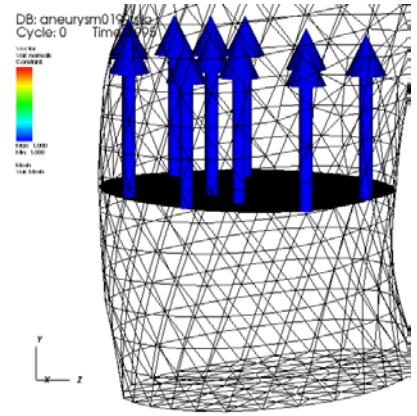
- Display the velocity vectors
- Display the surface normals
  make sure they point in the up direction

- Apply the flux operator

```
AddPlot("Pseudocolor",
"operators/Flux/Mesh", 0, 0)
AddOperator("Slice", 0)
opAtts = DeferExpressionAttributes()
opAtts.exprs = ("normals")
SetOperatorOptions(opAtts)
opAtts = FluxAttributes(1)
opAtts.flowField = "velocity"
SetOperatorOptions(opAtts
```



Flow vectors



Surface normals

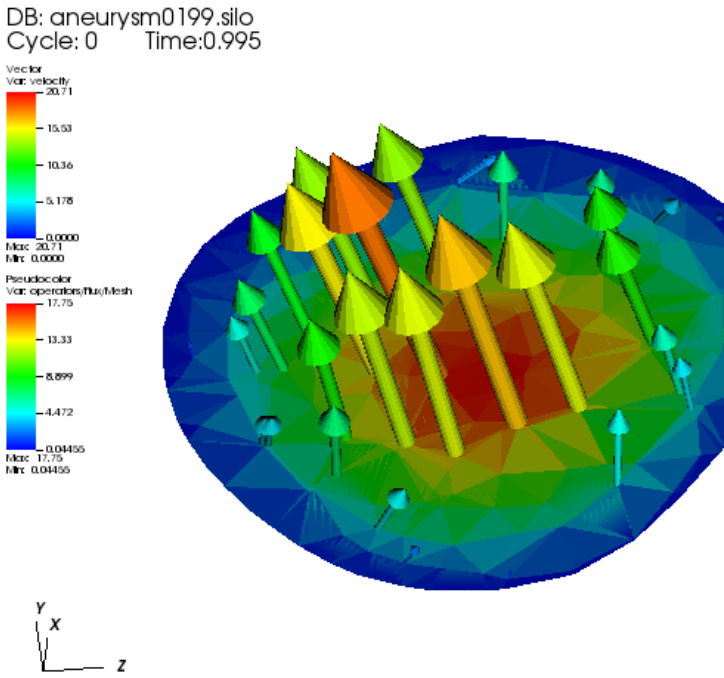
Flux through a surface (2)

- The slice is set to use the arbitrary vector <0, 1, 0>
- The flux is calculated on a cell-basis

- Do a summation by using the Query **“Weighted Variable Sum”**

Answer:

The total operators/Flux/Mesh is 0.790157



Transient Analysis

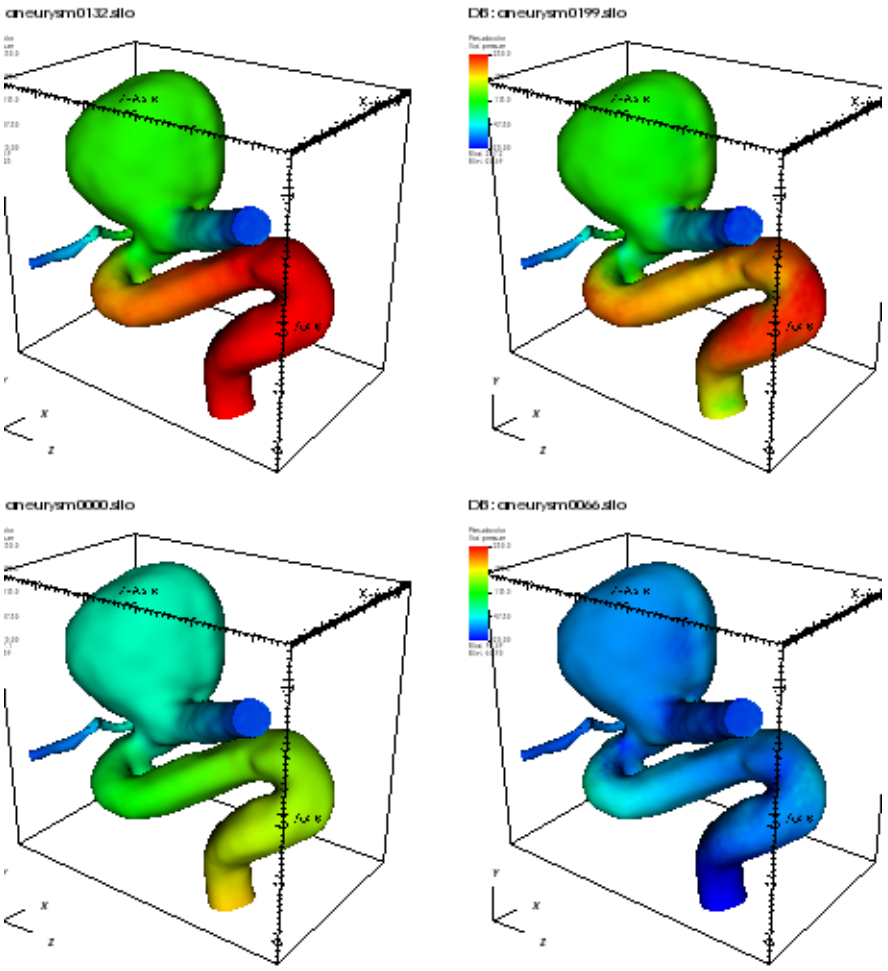
Uses data from multiple timesteps

- Aggregation, averages, …
- Comparisons between timesteps
- Pathlines

Multi-view

Uses data from multiple timesteps

- View orientations can be locked or independent
- Timesteps can be locked or independent

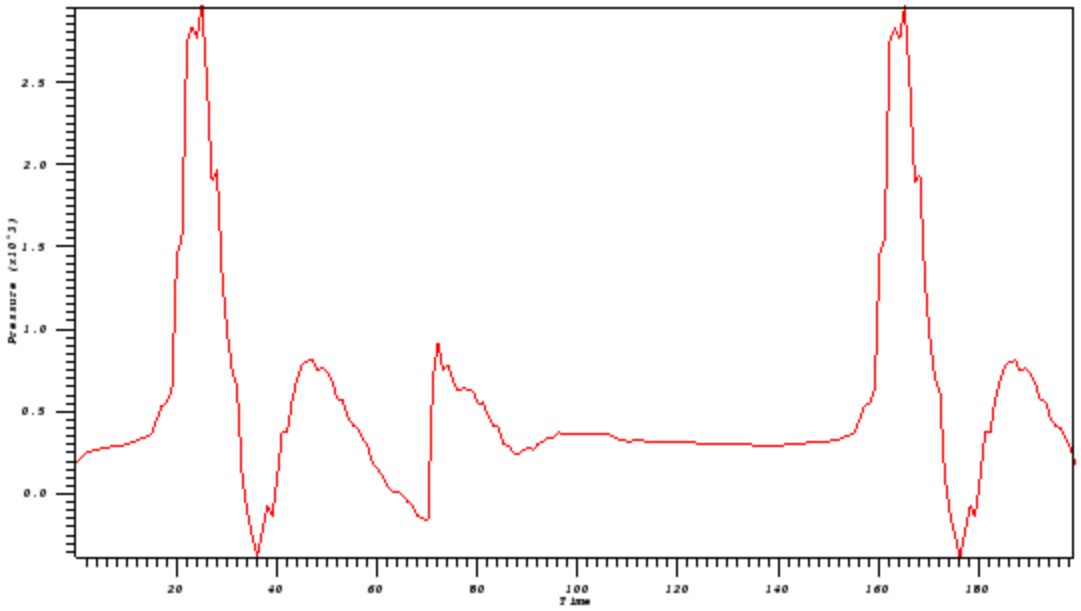


Pressure at a given node throughout time (1)

```
OpenDatabase("aneurysm.visit")
AddPlot("Pseudocolor", "pressure")
DrawPlots()

f = open("pressure.curve", "w")
N = TimeSliderGetNStates()
node_id = 7301

for i in range( N ):
  TimeSliderSetState(i)
  Query("Time")
  t = GetQueryOutputValue()
  p = PickByNode(node_id, "pressure")
  f.write("%g %g\n" % (t, p['pressure']))

f.close()
exit()
```

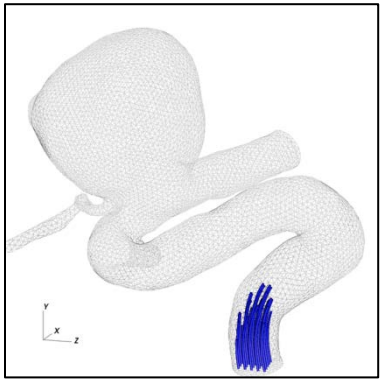
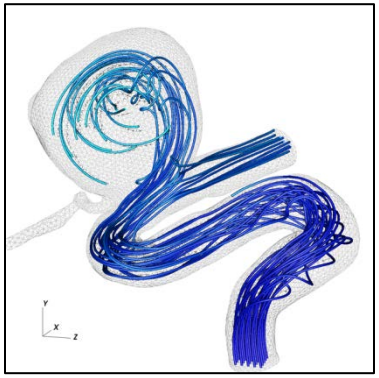
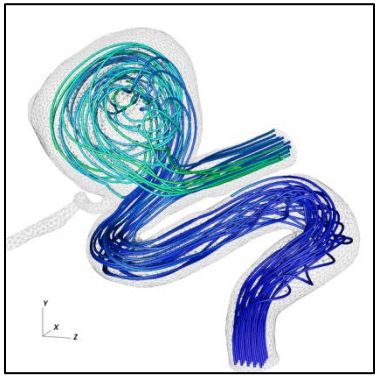
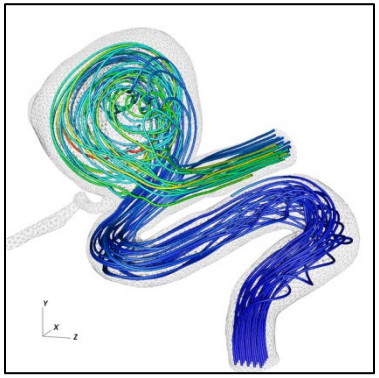
```
# run with:
# visit –nowin –cli –s picking.py
```

Pressure at a given node throughout time (2)




```
OpenDatabase("pressure.curve")
AddPlot("Curve", "curve")
DrawPlots()
```

Pathlines

Streamlines vs Pathlines

Streamline

○ Compute trajectories in an (instantaneous) snapshot of the vector field.
Uses and loads vector data from only the current time slice.

Pathline

◉ Compute trajectories in the time-varying vector field.
Uses and loads vector data from all relevant time slices

Pathlines Options

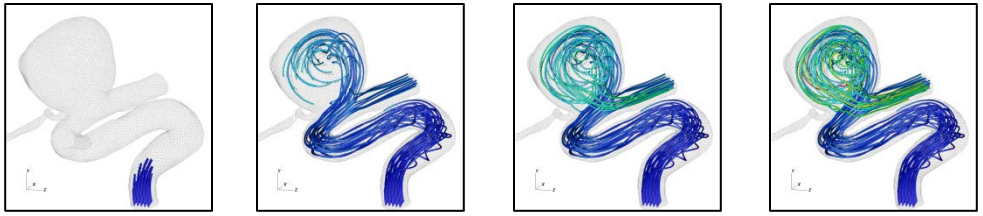
☐ Override Starting Time Time 0

How to perform interpolation over time

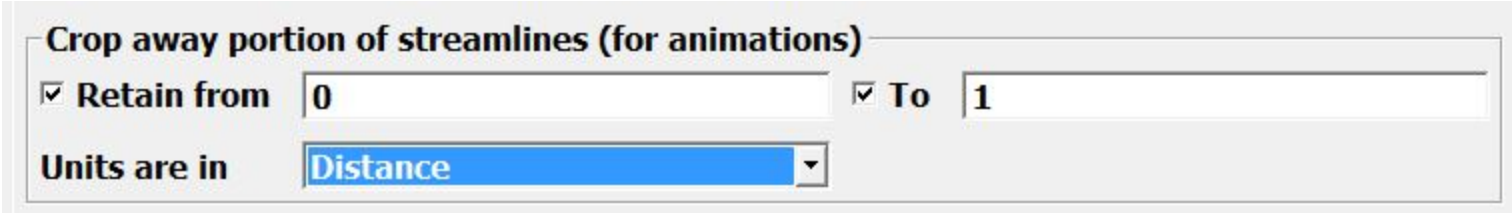
◉ Mesh is static over time (fast, but special purpose)

○ Mesh changes over time (slow, but robust)

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre



Pathlines



1. Compute the path (can be very long)
2. Display the path (very fast)
3. A simple python loop will enable you to animate the particles pathes

```
atts = StreamlineAttributes()
atts.displayEndFlag = 1
for i in range(100):
        atts.displayEnd = final_time + i / (101)
        SetPlotOptions(atts)
        SaveWindow()
```

See also http://portal.nersc.gov/project/visit/cyrush/sc12/Slides-Adv-3-Streamlines.pdf

Expressions...

1. **User-defined (use Controls->Expression menu)**

   average_over_time(pressure, "conn_cmfe", 0, 199, 10)

2. **User-defined (use Controls->Data-Level Comparisons menu)**

   p_diff = pressure - conn_cmfe(<[0]i:pressure>, Mesh)

   DefineScalarExpression("p_diff", "pressure - conn_cmfe(<[0]i:pressure>, Mesh)")

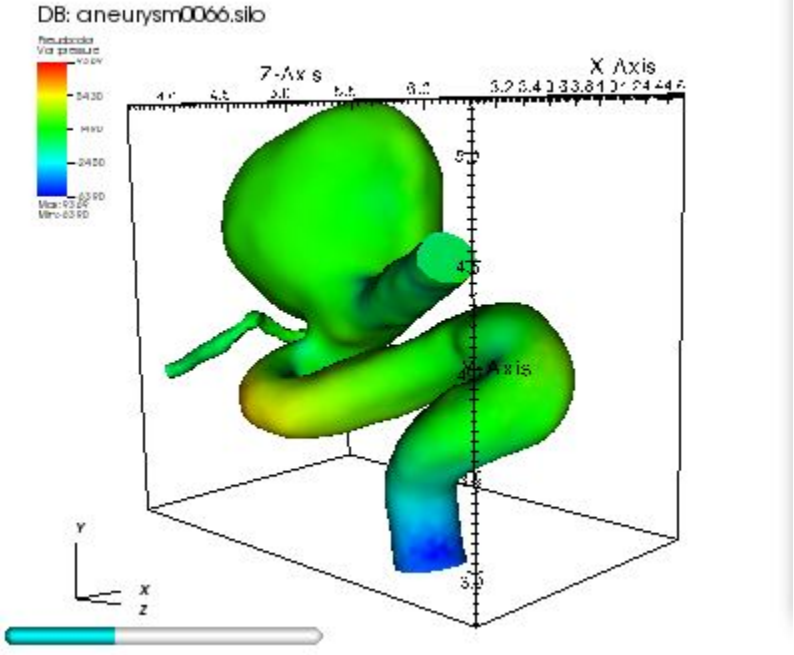
3. **Predefined expressions (see Preferences and File Information)**

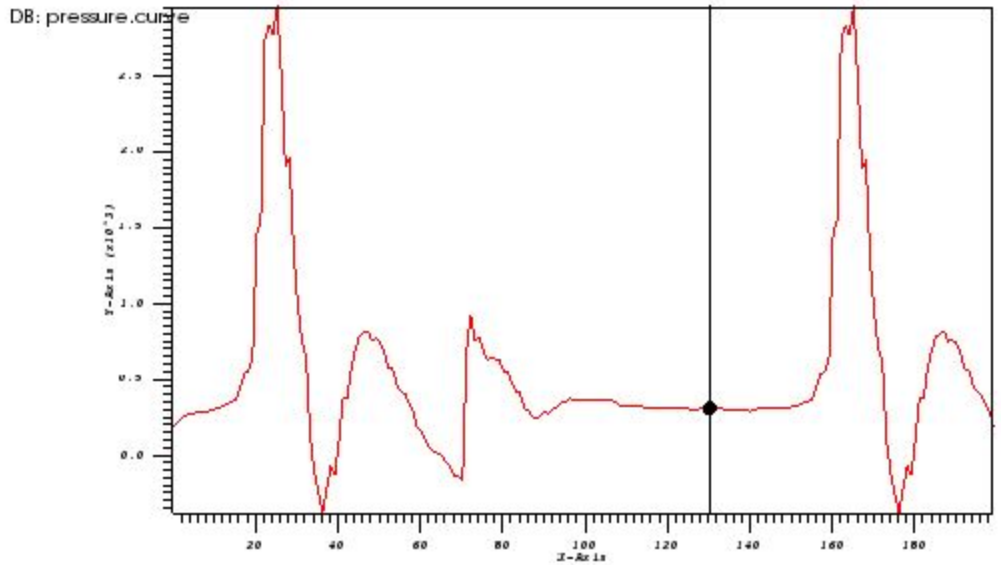
time_derivative/conn_based/pressure (scalar): (<pressure> - conn_cmfe(<[-1]id:pressure>, Mesh)) / (<time_derivative/conn_based/Mesh_time> - <time_derivative/conn_based/Mesh_lasttime>)

See also the Wiki article on CMFE

Making it pretty

1. Streamlines are shown as tubes
2. Shadows are switched on to better show the superposition of the streamlines (-tubes)
3. See Annotations
   - Time slider
   - Time Cue on the curves





Final Recommandations

Working with transient data can take a long time

1. Script everything in python, to replay it at will, and to execute it in batch (in parallel).

2. Save (Export Database) will let you archive the pathlines to replay them later.

©