Post-processing utilities in Elmer

Peter Råback

ElmerTeam
CSC – IT Center for Science

Elmer/Ice advanced course
CSC, 4-6.11.2013

Alternative postprocessors for Elmer

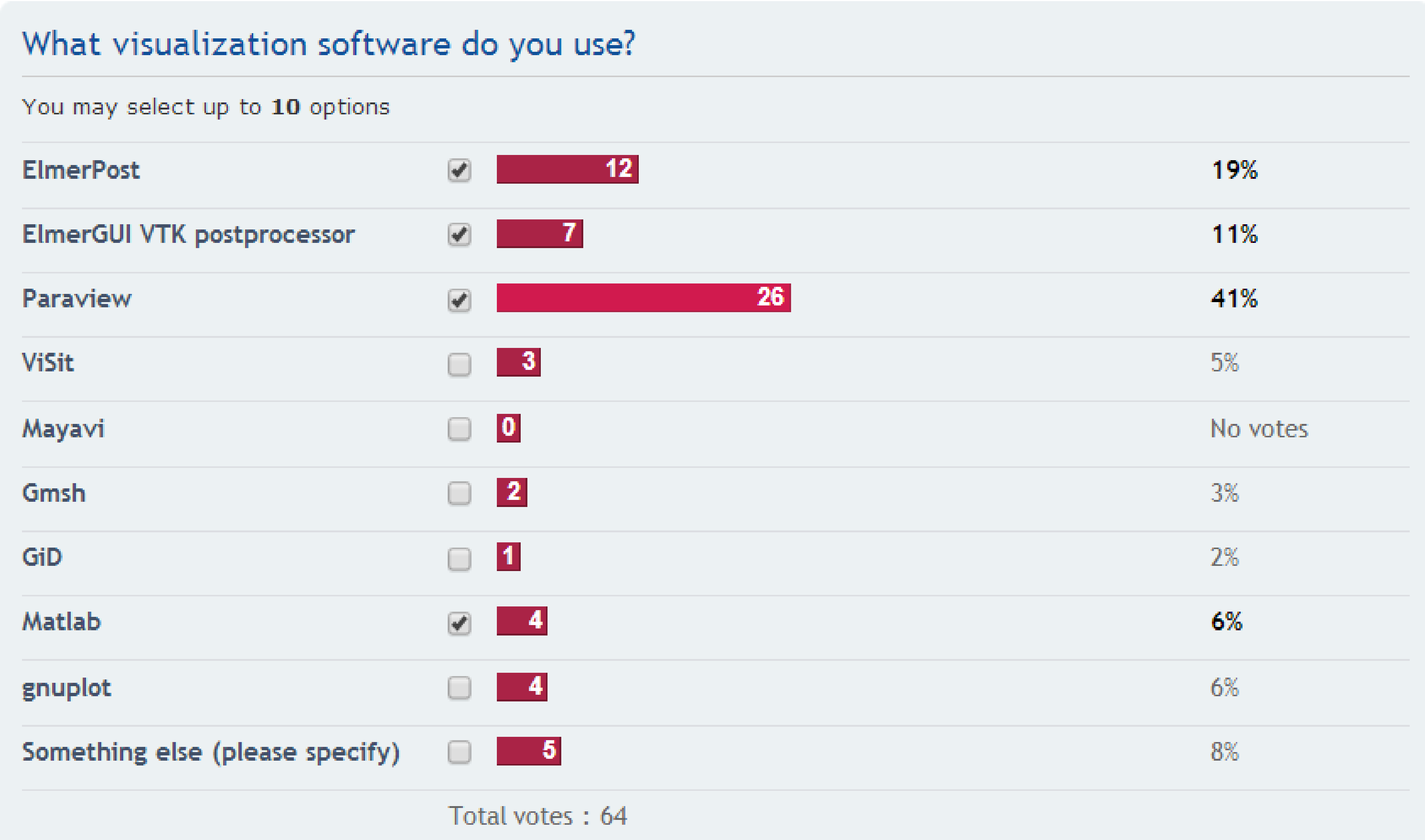
Open source

- ElmerPost
  - Postprocessor of Elmer suite
- ParaView, Visit
  - Use ResultOutputSolve to write .vtu or .vtk
  - Visualization of parallel data
- OpenDX
  - Supports some basic elementtypes
- Gmsh
  - Use ResultOutputSolve to write data
- Gnuplot, R, Octave, …
  - Use SaveData to save results in ascii matrix format
  - Line plotting

Commercial

- Matlab, Excel, …
  - Use SaveData to save results in ascii matrix format
  - Line plotting

Visualization tools – Poll (10/2013)



What visualization software do you use?

You may select up to **10** options

ElmerPost	☑	12	**19%**
ElmerGUI VTK postprocessor	☑	7	**11%**
Paraview	☑	26	**41%**
ViSit	☐	3	5%
Mayavi	☐	0	No votes
Gmsh	☐	2	3%
GiD	☐	1	2%
Matlab	☑	4	**6%**
gnuplot	☐	4	6%
Something else (please specify)	☐	5	8%

Total votes : 64

Exporting 2D/3D data: ResultOutputSolve

- Apart from saving the results in .ep format it is possible to use other postprocessing tools
- ResultOutputSolve offers several formats
  - vtk: Visualization tookit legacy format
  - vtu: Visualization tookit XML format
  - Gid: GiD software from CIMNE: http://gid.cimne.upc.es
  - Gmsh: Gmsh software: http://www.geuz.org/gmsh
  - Dx: OpenDx software
- **Vtu** is the recommended format!
  - offers parallel data handling capabilities
  - Has binary and single precision formats for saving disk space
  - Suffix **.vtu** in Post File does this automatically

Exporting 2D/3D data: ResultOutputSolve

An example shows how to save data in unstructured XML VTK (.vtu) files to directory ”results” in single precision binary format.

```
Solver n
  Exec Solver = after timestep
  Equation = "result output"
  Procedure = "ResultOutputSolve" "ResultOutputSolver"
  Output File Name = "case"
  Output Format = String ”vtu”
  Binary Output = True
  Single Precision = True
End
```

Derived fields

- Many solvers have internal options for computing derived fields (fluxes, heating powers,…)
- Elmer offers several auxiliary solvers
  - SaveMaterials: makes a material parameter into field variable
  - Streamlines: computes the streamlines of 2D flow
  - FluxComputation: given potential, computes the flux $q = - c \nabla\phi$
  - VorticitySolver: computes the vorticity of flow, $w = \nabla\times\phi$
  - PotentialSolver: given flux, compute the potential $- c \nabla\phi = q$
  - Filtered Data: compute filtered data from time series (mean, fourier coefficients,…)
  - …
- Usually auxiliary data need to be computed only after the iterative solution is ready
  - Exec Solver = after timestep
  - Exec Solver = after all
  - Exec Solver = before saving

Derived nodal data

- By default Elmer operates on distributed fields but sometimes nodal values are of interest
  - Multiphysics coupling may also be performed alternatively using nodal values for computing and setting loads
- Elmer computes the nodal loads from *Ax-b* where *A*, and *b* are saved before boundary conditions are applied
  - **`Calculate Loads = True`**
- This is the most consistant way of obtaining boundary loads
- Note: the nodal data is really pointwise
  - expressed in units N, C, W etc. (rather than N/m^2, C/m^2, W/m^2 etc.)
  - For comparison with distributed data divided by the ~size of the surface elements

Derived lower dimensional data

- Derived boundary data
  - SaveLine: Computes fluxes on-the-fly
- Derived lumped (or 0D) data
  - SaveScalars: Computes a large number of different quantities on-the-fly
  - FluidicForce: compute the fluidic force acting on a surface
  - ElectricForce: compute the electrostatic froce using the Maxwell stress tensor
  - Many solvers compute lumped quantities internally for later use
    (Capacitance, Lumped spring,…)

Saving 1D data: SaveLine

- Lines of interest may be defined on-the-fly
- Flux computation using integration points on the boundary – not the most accurate
- By default saves all existing field variables

Saving 1D data: SaveLine…

```
Solver n
 Equation = "SaveLine"
 Procedure = File "SaveData" "SaveLine"
 Filename =  "g.dat"
 File Append = Logical True
 Polyline Coordinates(2,2) = Real 0.0 1.0 0.0 2.0
End

Boundary Condition m
  Save Line = Logical True
End
```

Saving 0D data: SaveScalars

Operators on bodies

- Statistical operators
  - Min, max, min abs, max abs, mean, variance, deviation
- Integral operators (quadratures on bodies)
  - volume, int mean, int variance
  - Diffusive energy, convective energy, potential energy

Operators on boundaries

- Statistical operators
  - Boundary min, boundary max, boundary min abs, max abs, mean, boundary variance, boundary deviation, boundary sum
  - Min, max, minabs, maxabs, mean
- Integral operators (quadratures on boundary)
  - area
  - Diffusive flux, convective flux

Other operators

- nonlinear change, steady state change, time, timestep size,…

Saving 0D data: SaveScalars…

```
Solver n
  Exec Solver = after timestep
  Equation = String SaveScalars
  Procedure = File "SaveData" "SaveScalars"
  Filename = File "f.dat"
  Variable 1 = String Temperature
  Operator 1 = String max
  Variable 2 = String Temperature
  Operator 2 = String min
  Variable 3 = String Temperature
  Operator 3 = String mean
End

Boundary Condition m
  Save Scalars = Logical True
End
```

Case: TwelveSolvers

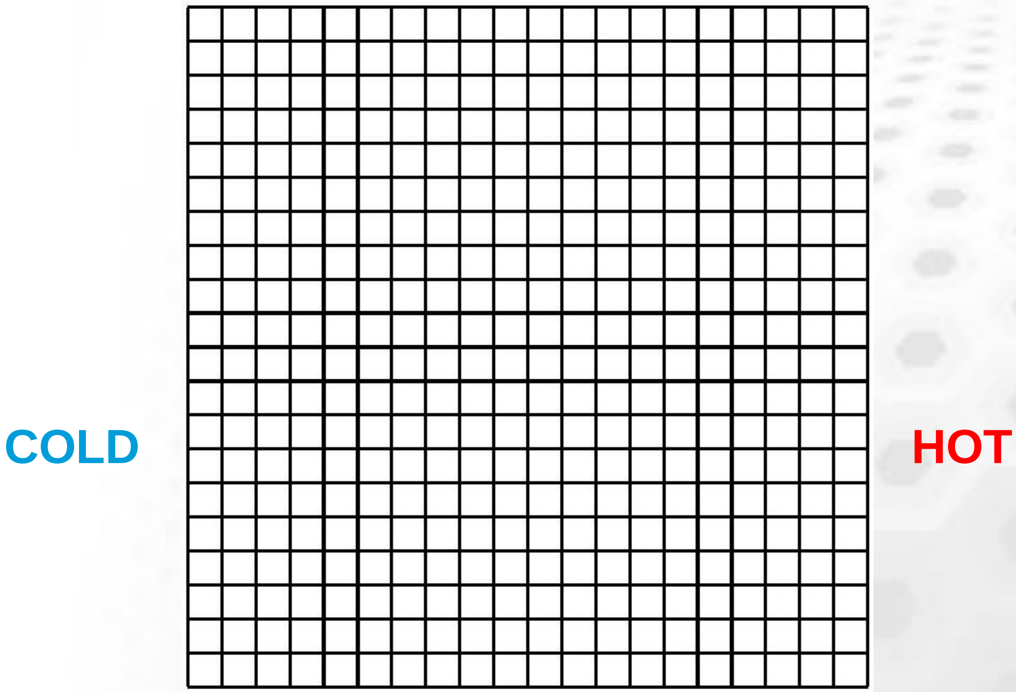
Natural convection with ten auxialiary solvers

Case: Motivation

- The purpose of the example is to show the flexibility of the modular structure
- The users should not be afraid to add new atomistic solvers to perform specific tasks
- A case of 12 solvers is rather rare, yet not totally unrealitistic

Case: preliminaries

- Square with hot wall on right and cold wall on left
- Filled with viscous fluid
- Bouyancy modeled with Boussinesq approximation
- Temperature difference initiates a convection roll



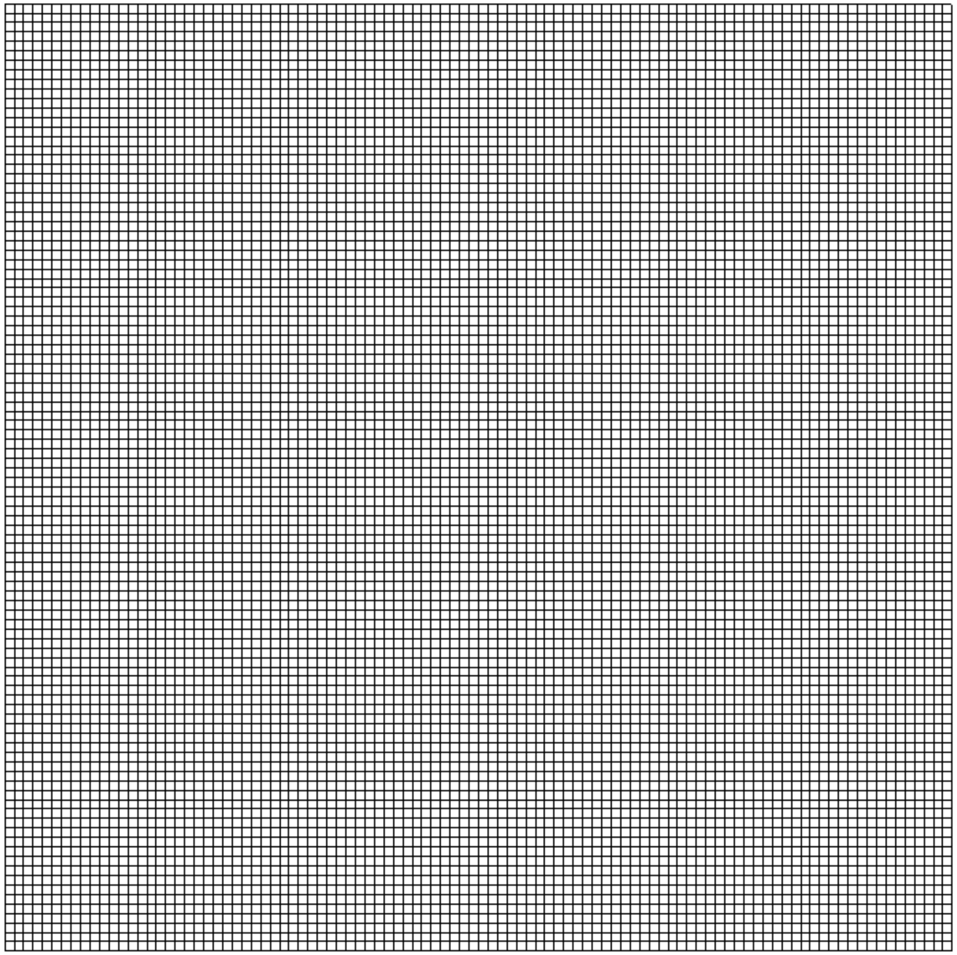
Case: 12 solvers

1. Heat Equation
2. Navier-Stokes

---

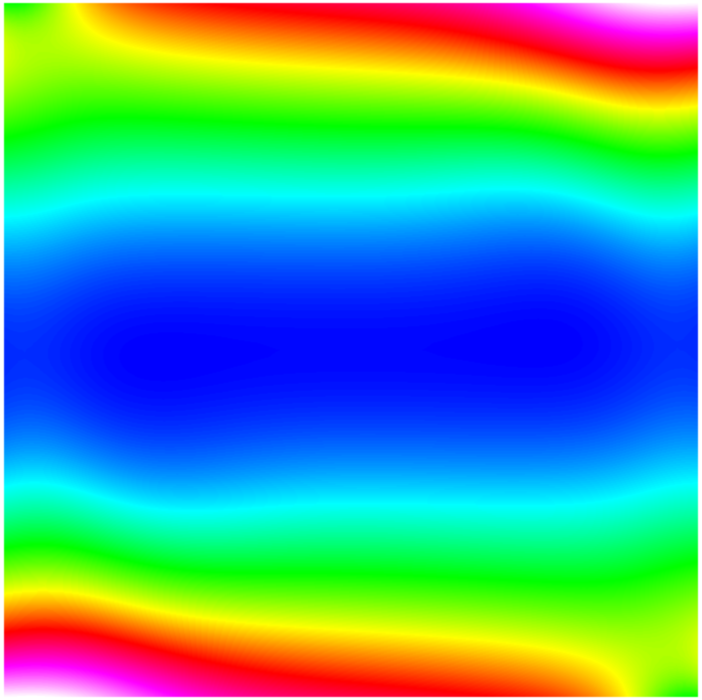
3. FluxSolver: solve the heat flux
4. StreamSolver
5. VorticitySolver
6. DivergenceSolver
7. ShearrateSolver
8. IsosurfaceSolver
9. ResultOutputSolver
10. SaveGridData
11. SaveLine
12. SaveScalars

Case: Computational mesh

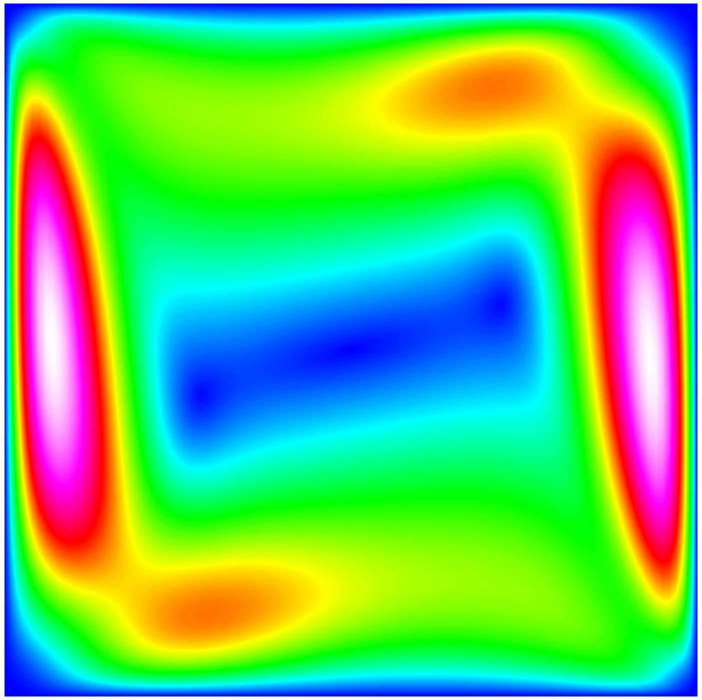


10000 bilinear elements

Case: Navier-Stokes, primary fields

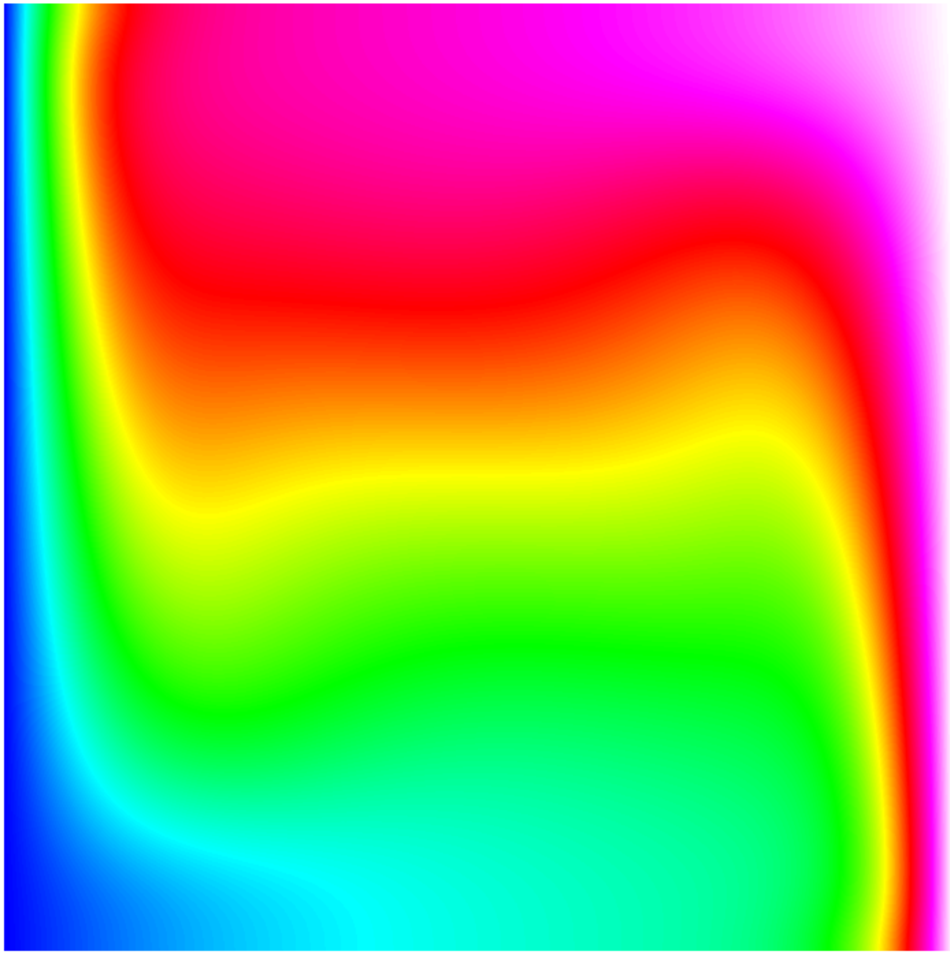


Pressure

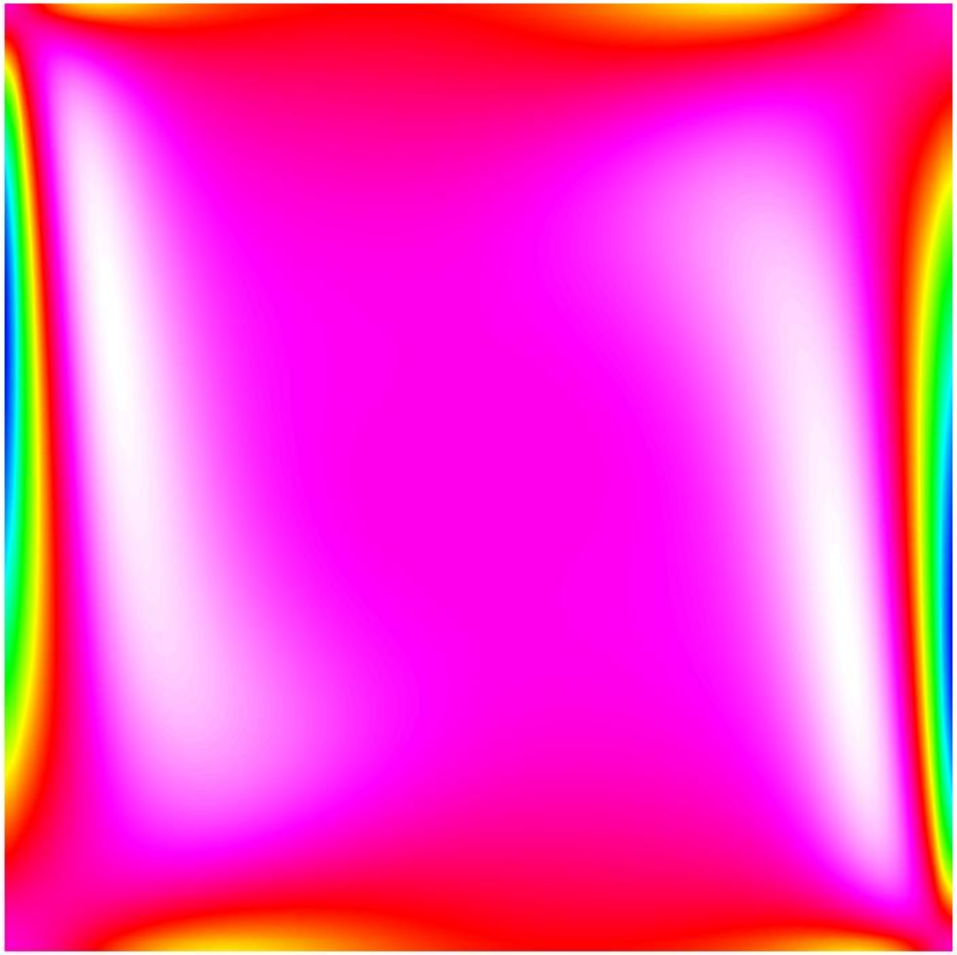


Velocity

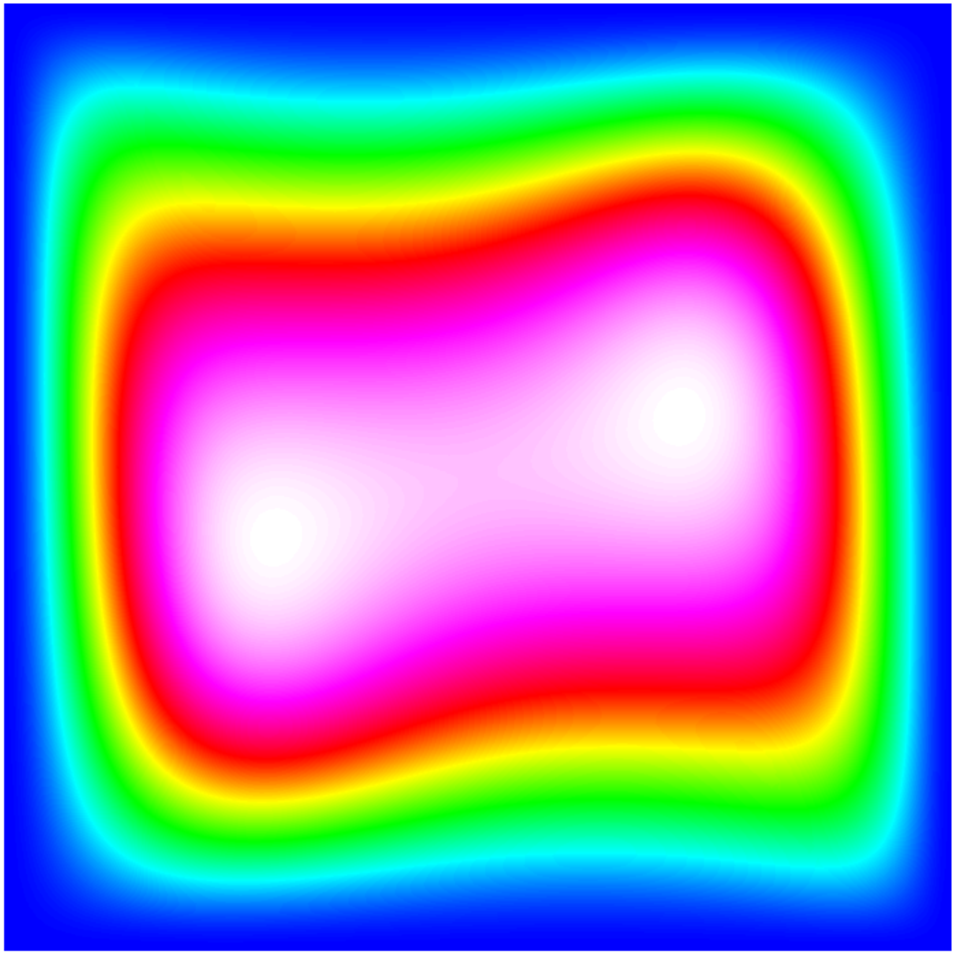
Case: Heat equation, primary field



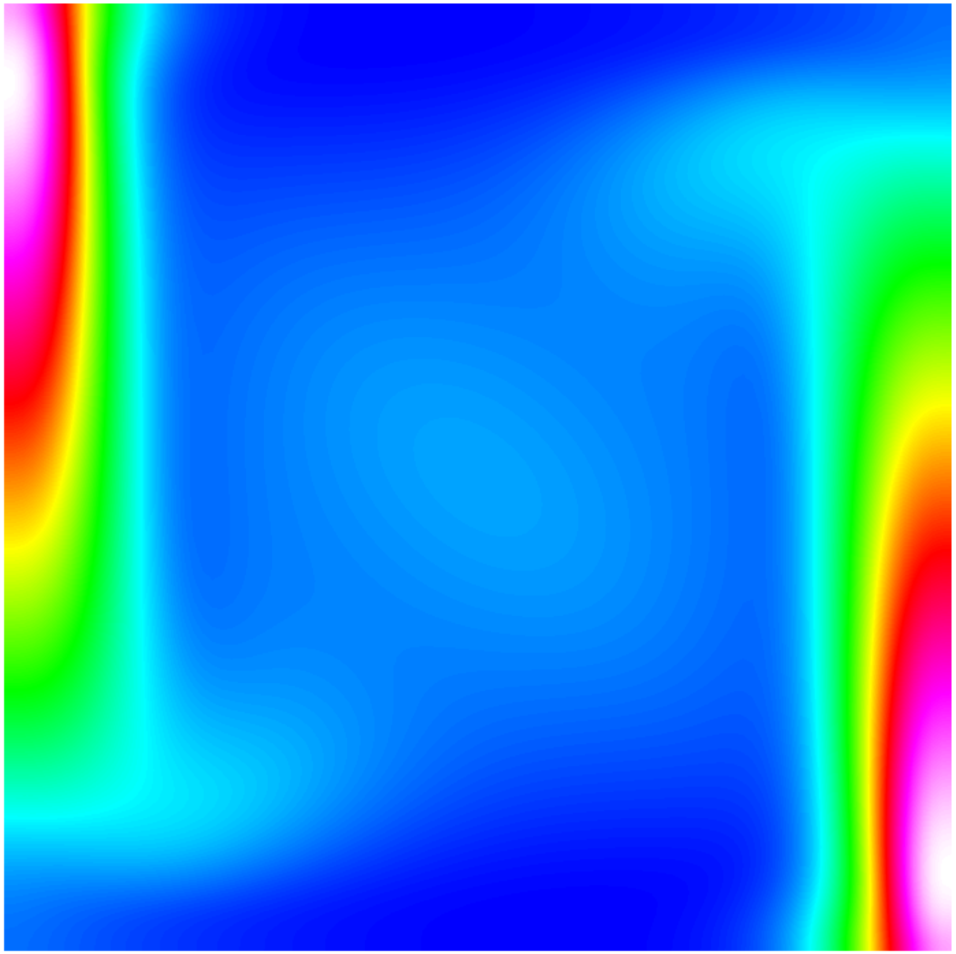
Case: Derived field, vorticity



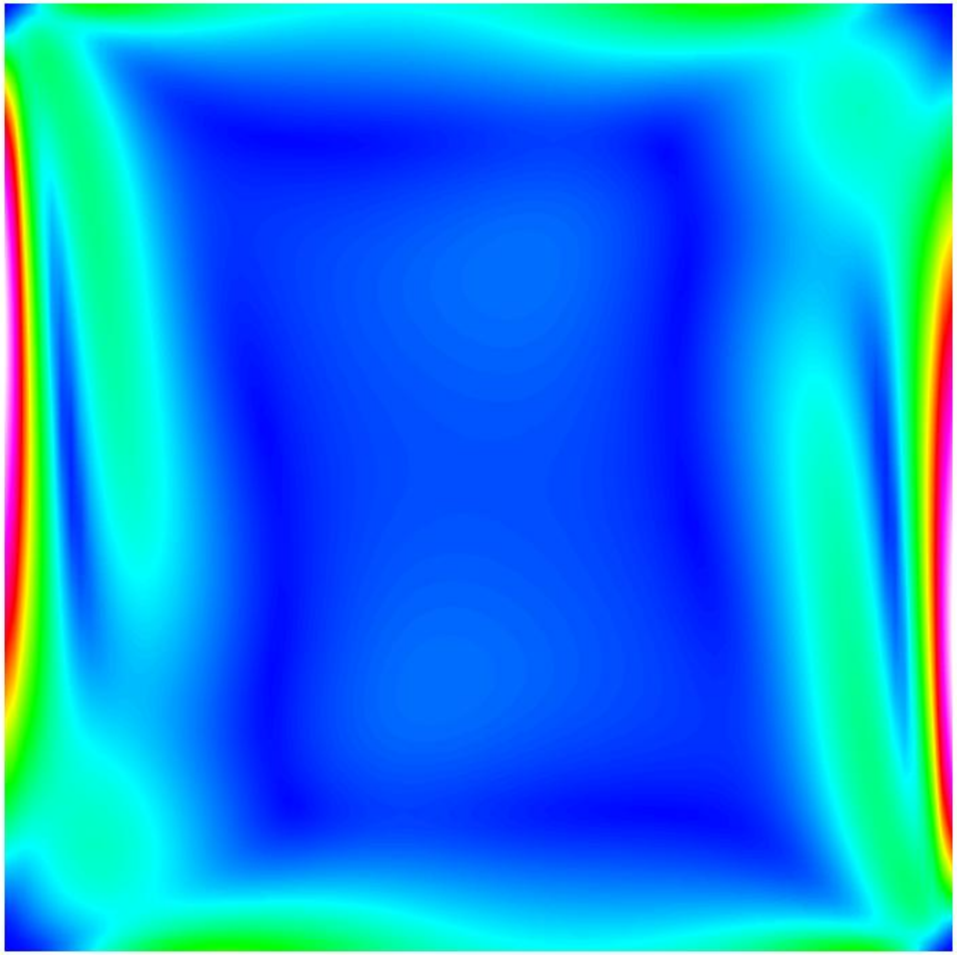
Case: Derived field, Streamlines



Case: Derived field, diffusive flux

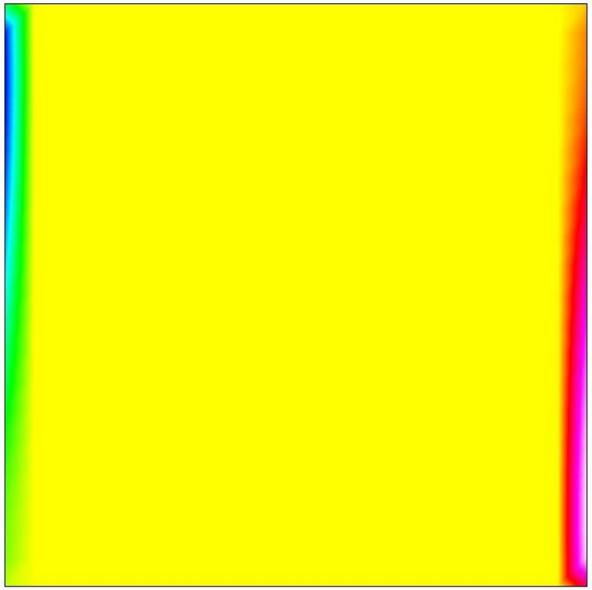


Case: Derived field, Shearrate



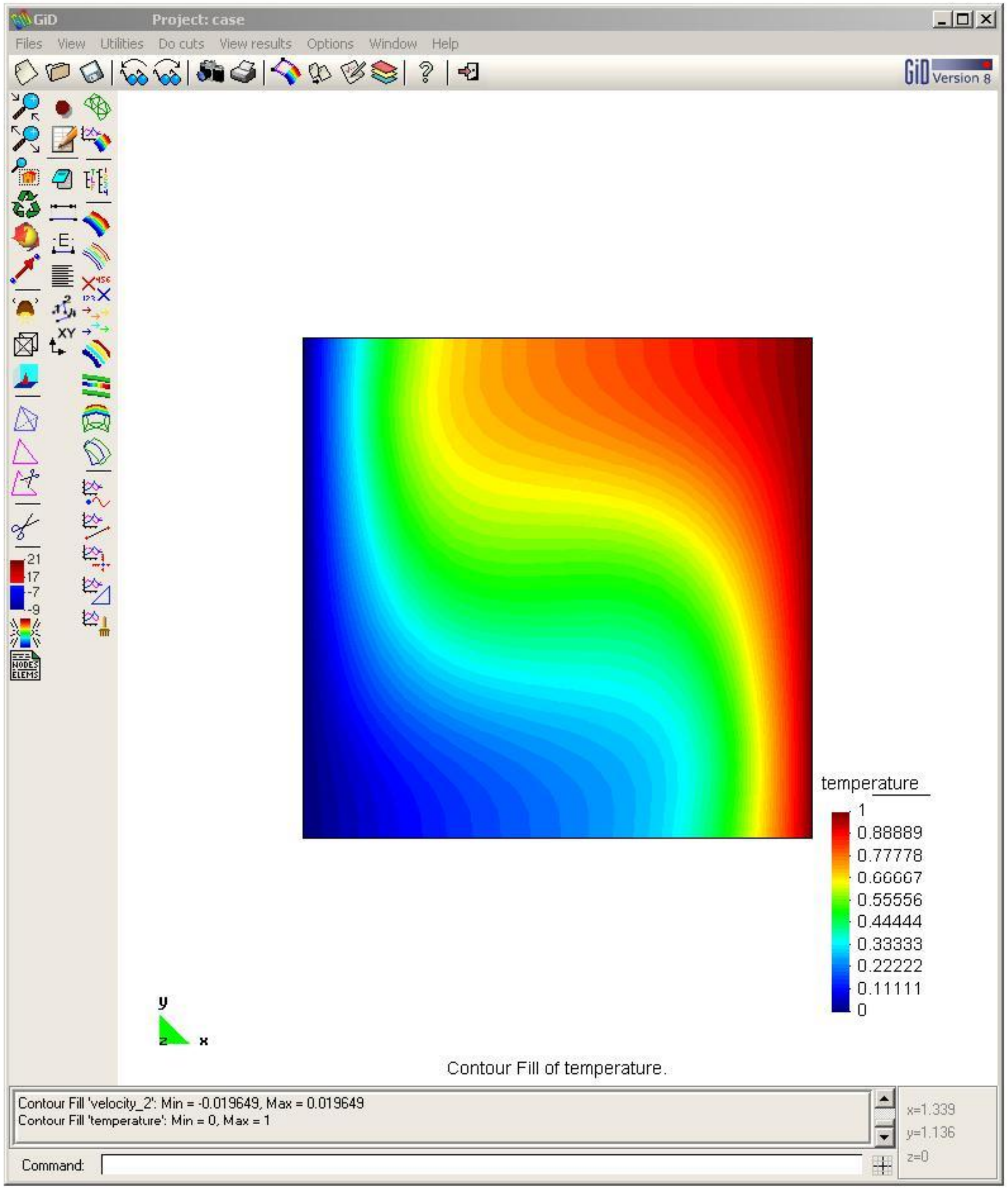
Example: nodal loads

- If equation is solved until convergence nodal loads should only occur at boundaries
- Element size h=1/20 ~weight for flux



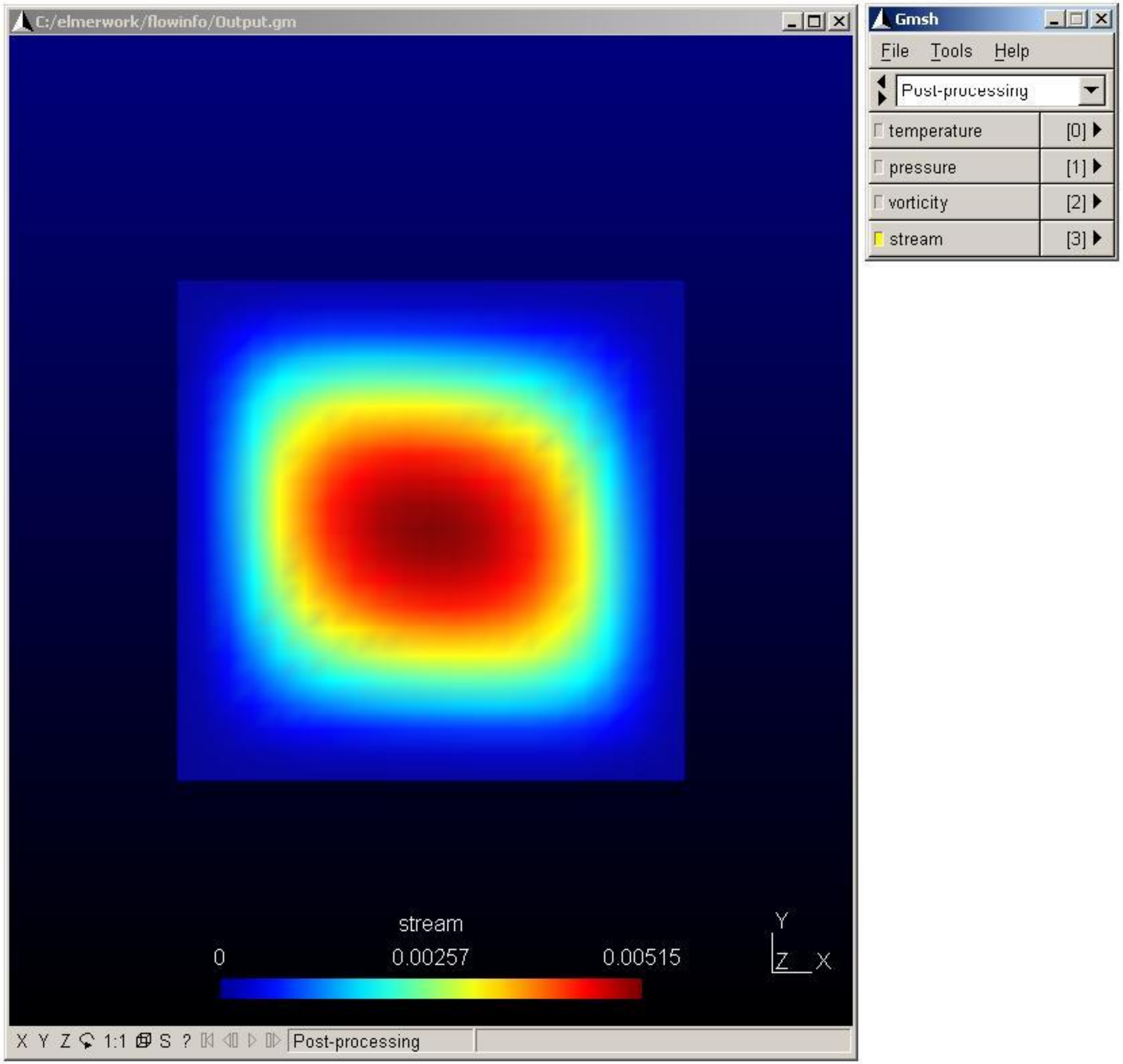
Nodal heat loads

Example:
view in GiD

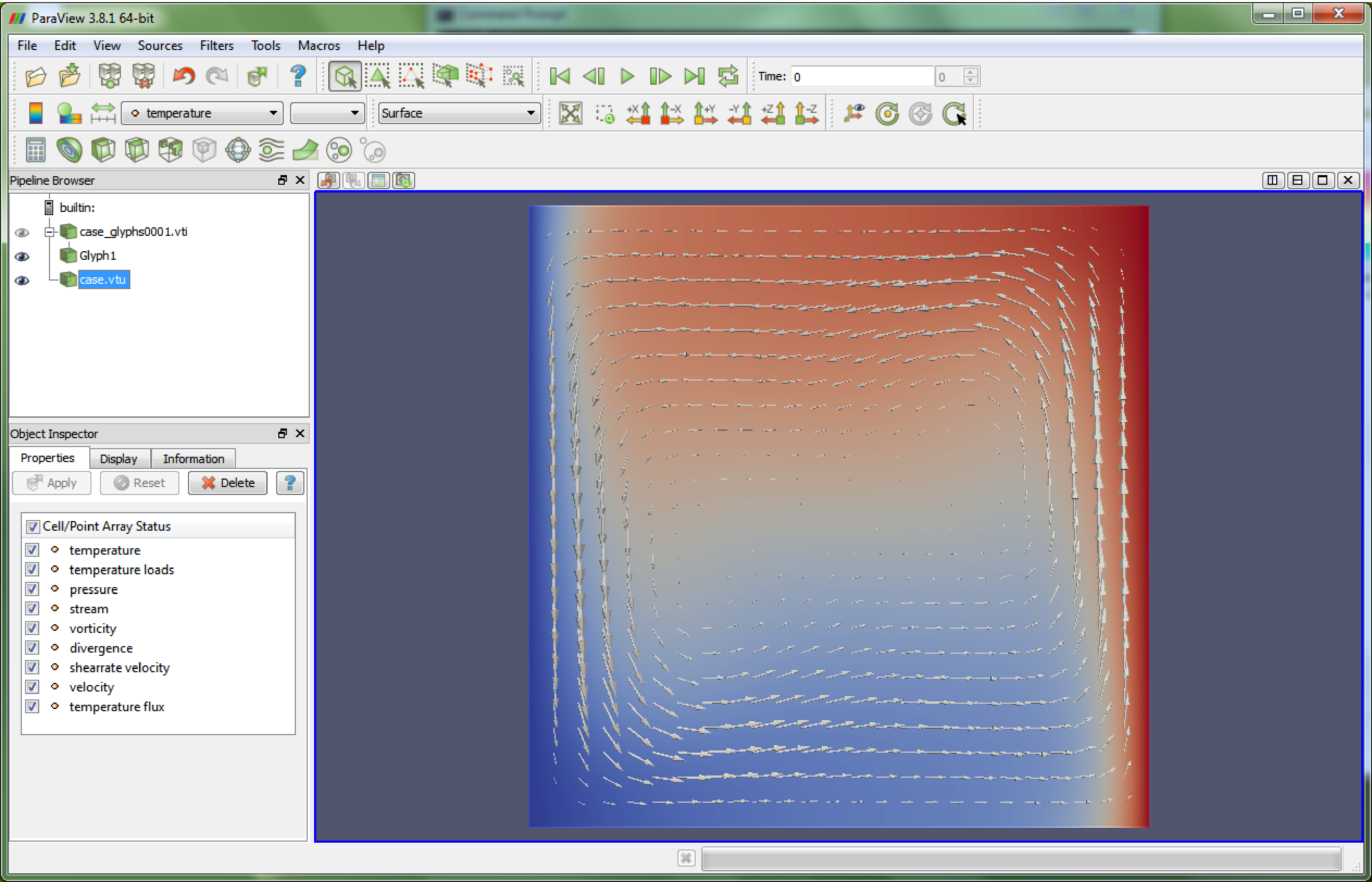


```
Contour Fill 'velocity_2': Min = -0.019649, Max = 0.019649
Contour Fill 'temperature': Min = 0, Max = 1
```

Example:
view in Gmsh



Case: View in Paraview



Time: 0

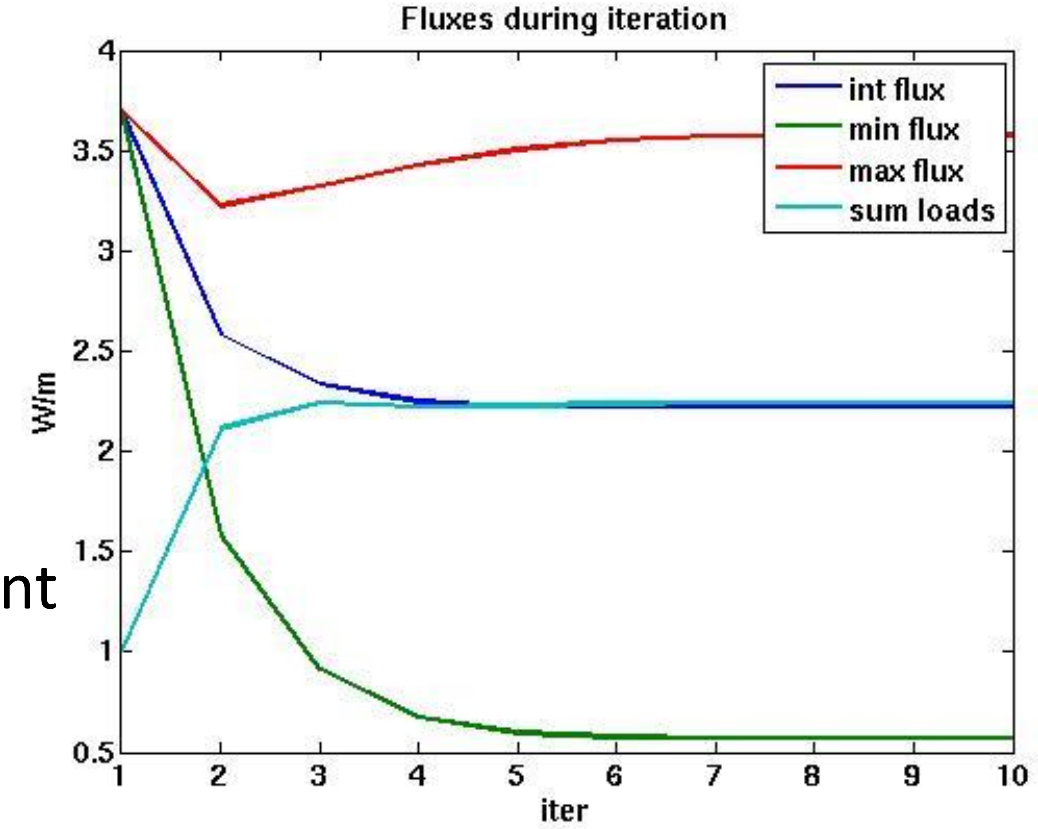
temperature

Surface

☑ Cell/Point Array Status
- ☑ temperature
- ☑ temperature loads
- ☑ pressure
- ☑ stream
- ☑ vorticity
- ☑ divergence
- ☑ shearrate velocity
- ☑ velocity
- ☑ temperature flux

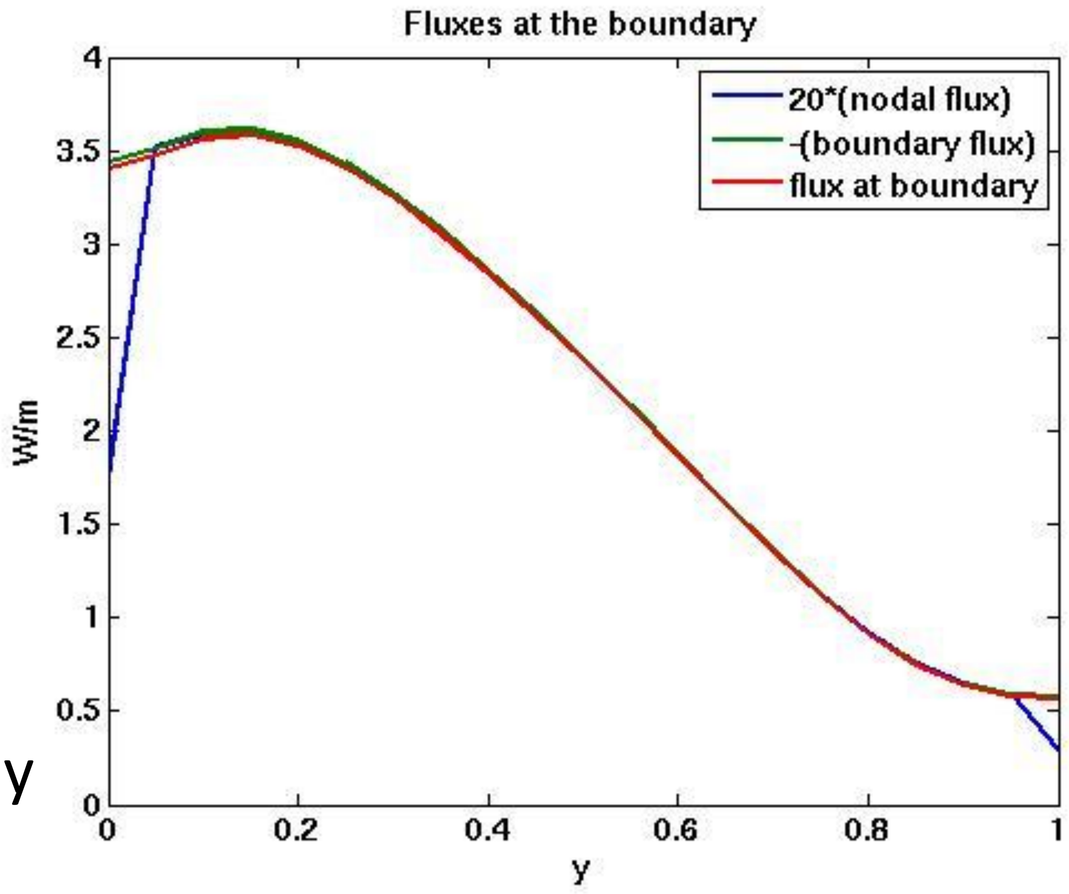
Example: total flux

- Saved by SaveScalars
- Two ways of computing the total flux give different approximations
- When convergence is reached the agreement is good



Example: boundary flux

- Saved by SaveLine
- Three ways of computing the boundary flux give different approximations
- At the corner the nodal flux should be normalized using only *h/2*



Exercise

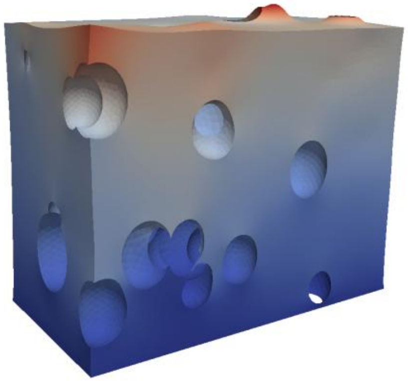
- Study the command file with 12 solvers
- Copy-paste an appropriate solver from there to some existing case of your own
  - ResultOutputSolver for VTU output
  - StreamSolver, VorticitySolver, FluxSolver,…
- Note: Make sure that the numbering of Solvers is consistant
  - Solvers that involve finite element solution you need to activate by **`Active Solvers`**
- Run the modified case
- Visualize results in ElmerPost or Paraview

Overcoming bottle-necks in postprocessing

- Despite good visualization tools sometimes the amount of data may become a bottle-neck
- Reducing data
  - Saving only boundaries
  - Uniform point clouds
  - A priori defined isosurfaces
  - Using coarser meshes for output when hierarchy of meshes exist
- Extracting data
  - Dimensional reduction (3D -> 2D)
  - Averaging over time
  - Integrals over BCs & bodies
- More robust I/O
  - Not all cores should write to disk in massively parallel simulations
  - Preliminary HDF5+XDML output available for Elmer, mixed experiences

Example, File size in Swiss Cheese

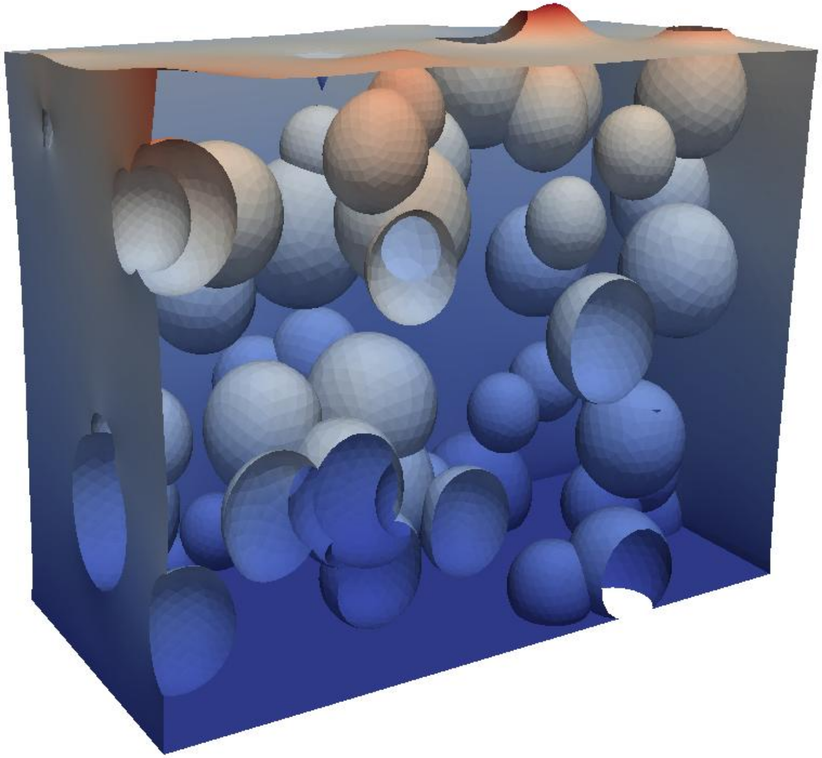
- Memory consumption of vtu-files (for Paraview) was studied in the ”swiss cheese” case
- The ResultOutputSolver with different flags was used to write output in parallel
- Saving just boundaries in single precision binary format may save over 90% in files size compared to full data in ascii
- With larger problem sizes the benefits are amplified



Binary output	Single Prec.	Only bound.	Bytes/node
-	X	-	376.0
X	-	-	236.5
X	X	-	184.5
X	-	X	67.2
X	X	X	38.5

Example, saving boundaries in .sif file

```
Solver 2
  Exec Solver = Always
  Equation = "result output"
  Procedure = "ResultOutputSolve” "ResultOutputSolver"
  Output File Name = case
  Vtu Format = Logical True
  Save Boundaries Only = Logical True
End
```



Conclusions

- It is good to think in advance what kind of data you need
  - 3D volume and 2D surface data
  - Derived fields
  - 1D line data
  - 0D lumped data
- Often the same operations may be done also at later stages but with significantly greater effort

Visualization with ElmerPost

How to write files for ElmerPost

- Default suffix is `.ep`
- May be requested in Simulation section

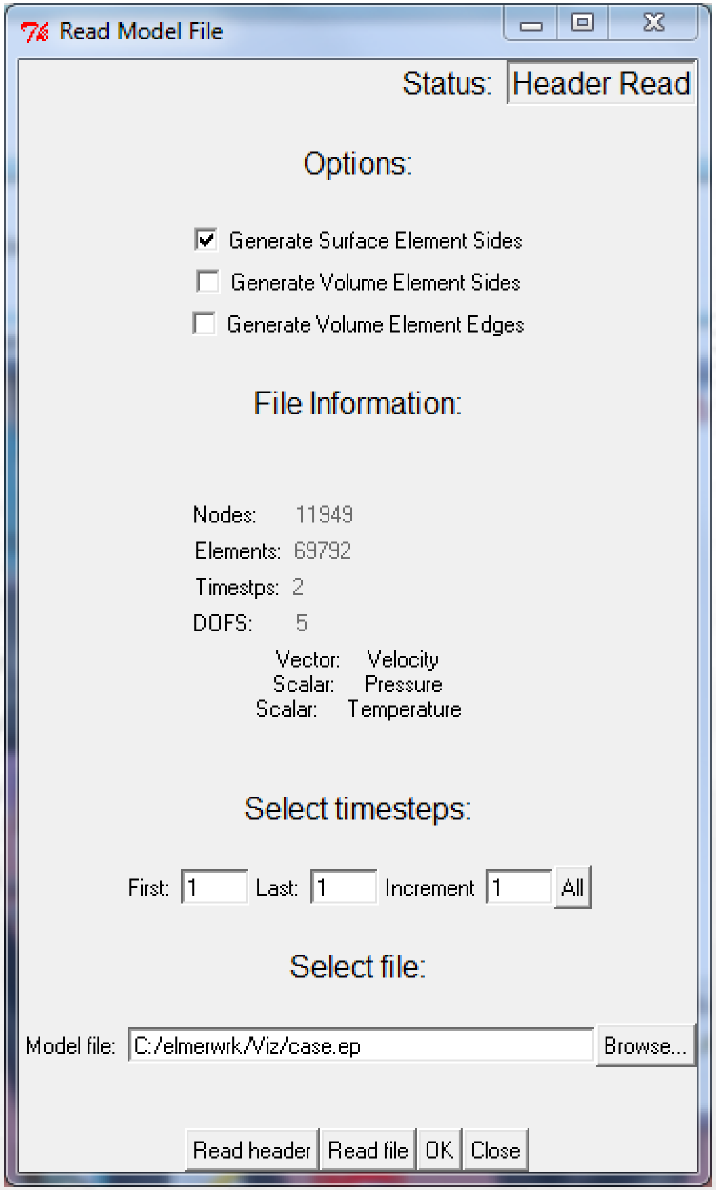
```
Post File = case.ep
```

- Or using ResultOutputSolver with

```
Output format = ElmerPost
```

Loading data

- Assume data in case.ep
- `File -> Open -> case.ep`
- Here the timesteps are chosen
- If element edges or sides are not defined for BCs they may have to be created here



Status: Header Read

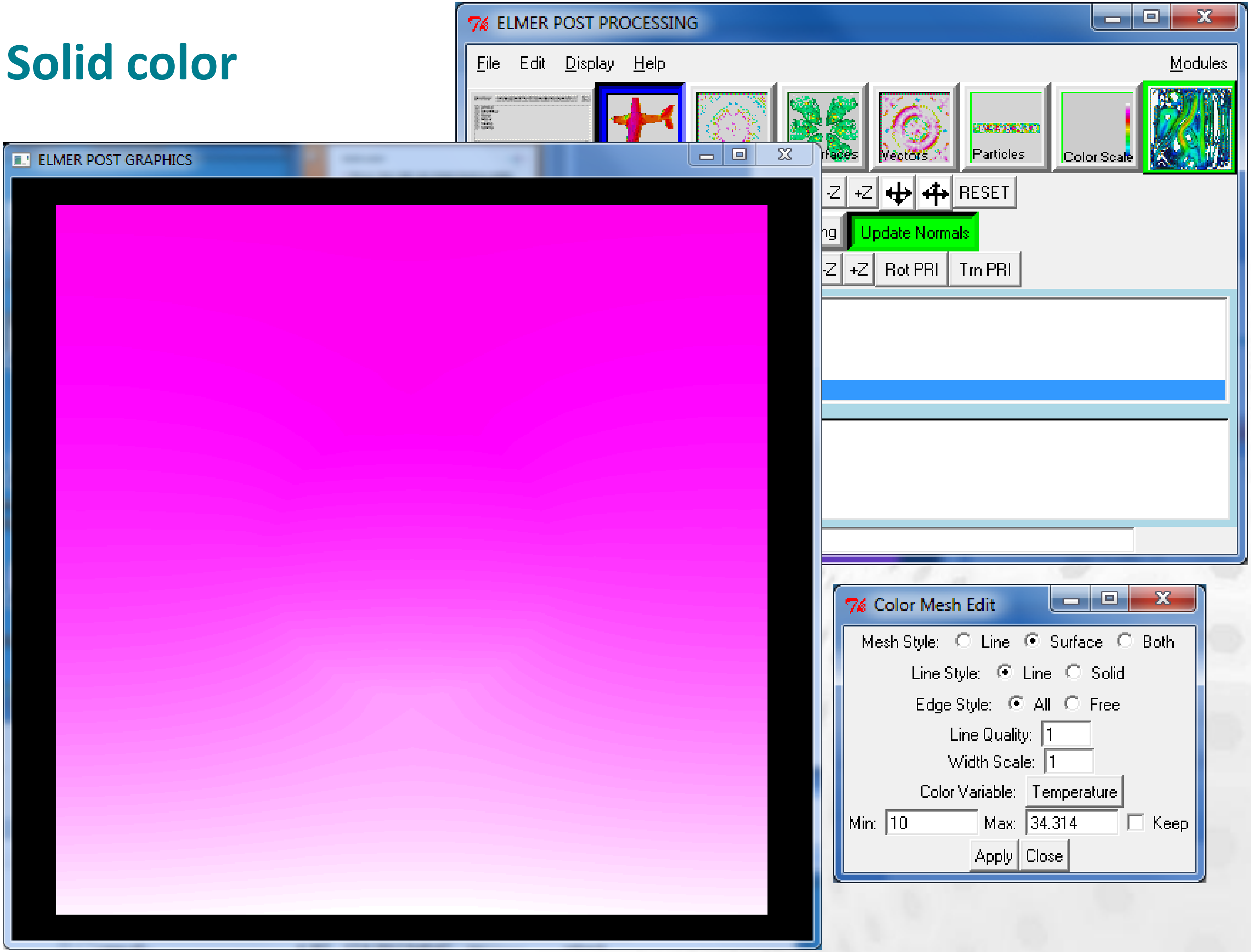
☑ Generate Surface Element Sides
☐ Generate Volume Element Sides
☐ Generate Volume Element Edges

Nodes: 11949
Elements: 69792
Timestps: 2
DOFS: 5
Vector: Velocity
Scalar: Pressure
Scalar: Temperature

First: 1 Last: 1 Increment 1

Model file: C:/elmerwrk/Viz/case.ep

Solid color



Mesh Style: ○ Line ● Surface ○ Both

Line Style: ● Line ○ Solid

Edge Style: ● All ○ Free

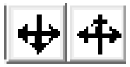
Line Quality: 1

Width Scale: 1

Color Variable: Temperature

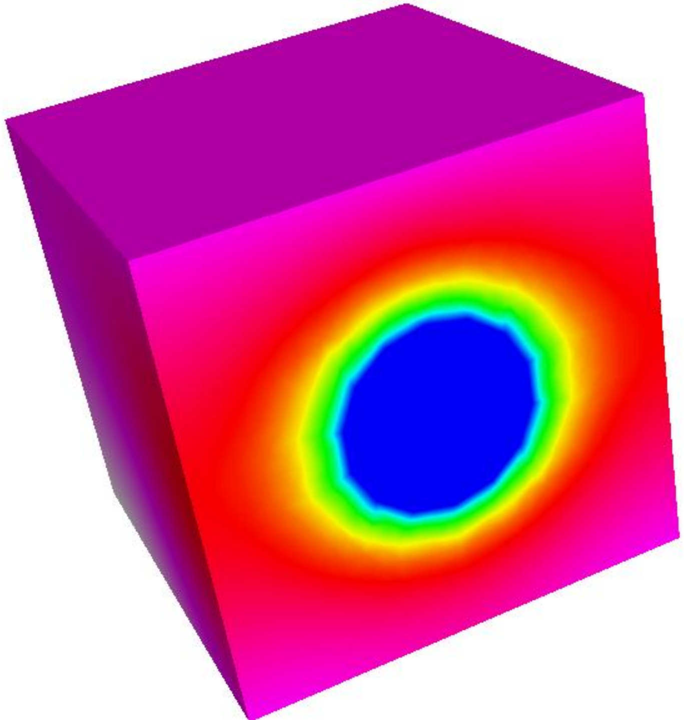
Min: 10 Max: 34.314 ☐ Keep

Moving object in ElmerPost

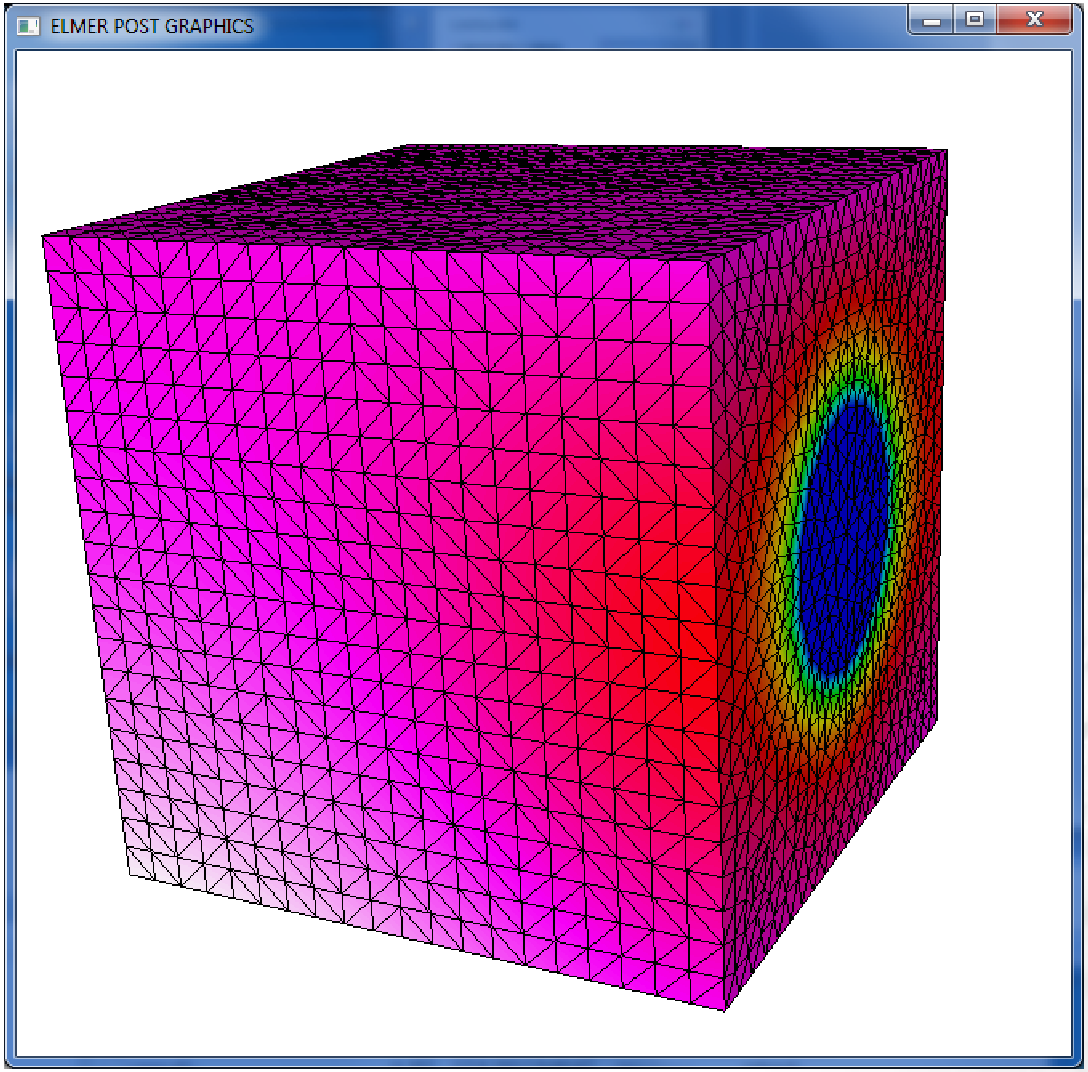
- Rotate
  - Mouse: Right bottom
  - Click: 
  - Command line, e.g.: `rotate 30 45 60`
- Scale
  - Mouse: Both bottoms
  - Click: 
  - Command line: `scale 1 10 1`
- Translate
  - Mouse: Left bottom
  - Click: 
  - Command line: `translate 1 2 3`

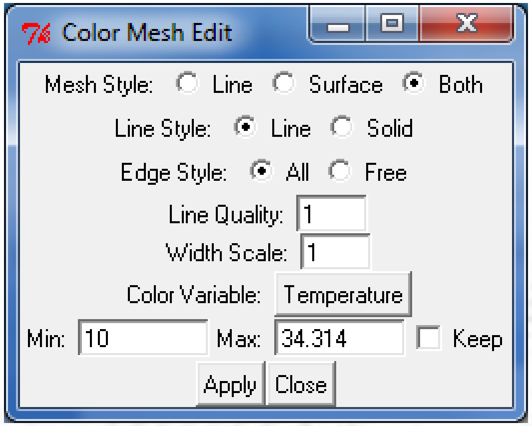
Setting background color

- Click:
  - Edit -> Background
  - Set 100.0 100.0 100.0 for white
- Command line
  - `background 100 100 100`



Color mesh with surface + edges





Mesh Style: ○ Line ○ Surface ● Both

Line Style: ● Line ○ Solid

Edge Style: ● All ○ Free

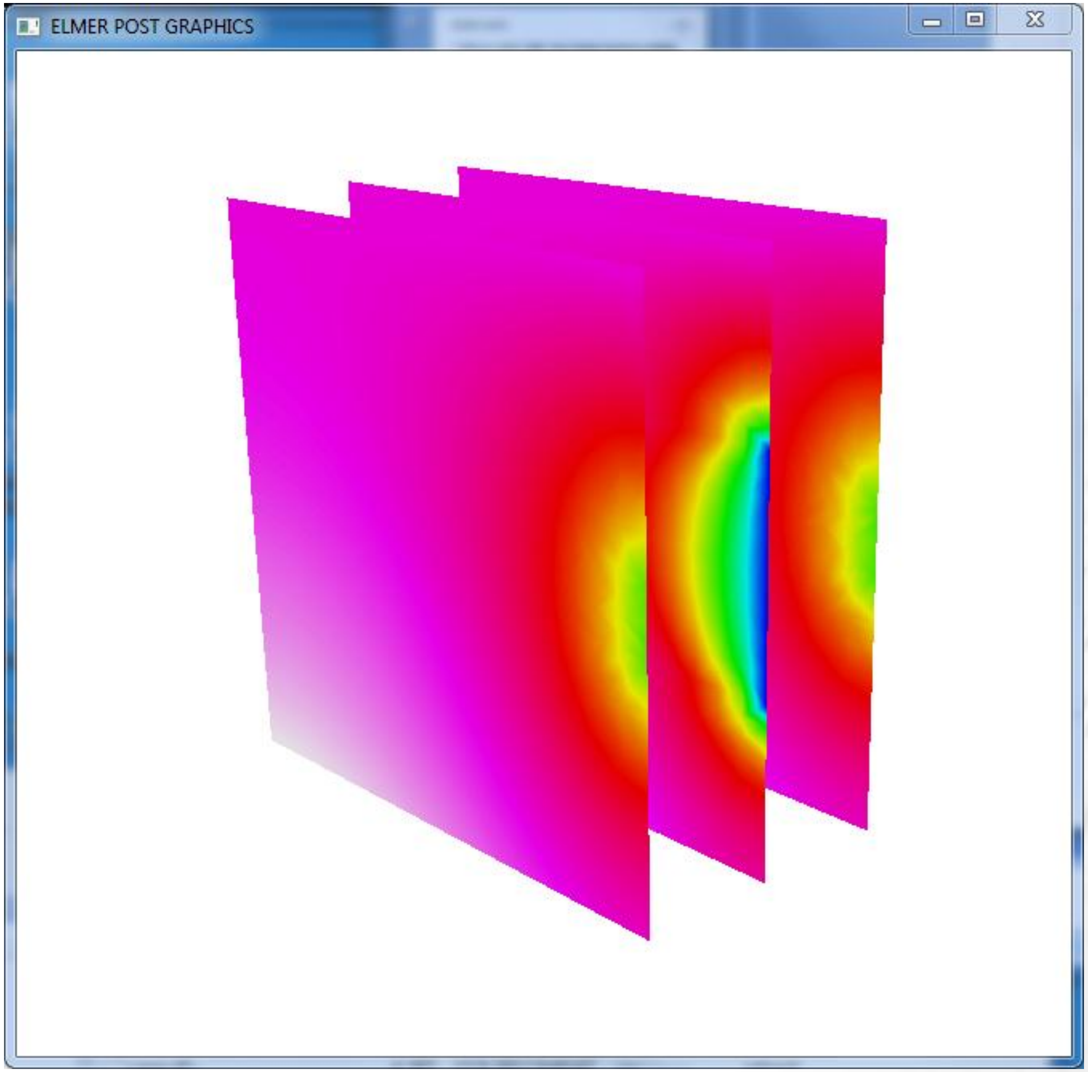
Line Quality: 1

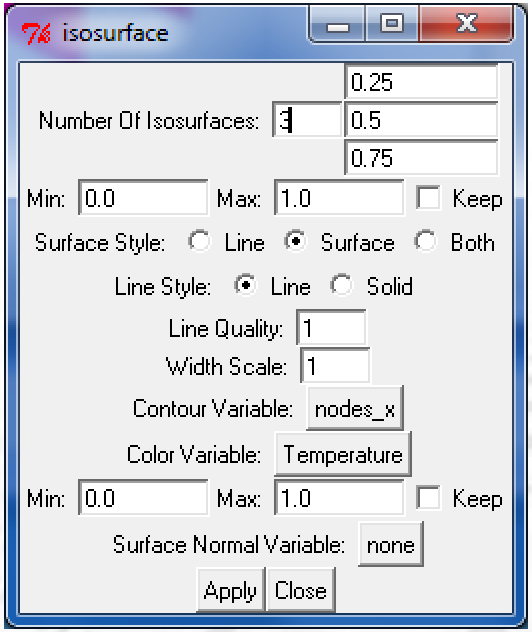
Width Scale: 1

Color Variable: Temperature

Min: 10 Max: 34.314 ☐ Keep

Plotting isosurfaces





Number Of Isosurfaces: 3

0.25

0.5

0.75

Min: 0.0 Max: 1.0 ☐ Keep

Surface Style: ○ Line ● Surface ○ Both

Line Style: ● Line ○ Solid

Line Quality: 1

Width Scale: 1

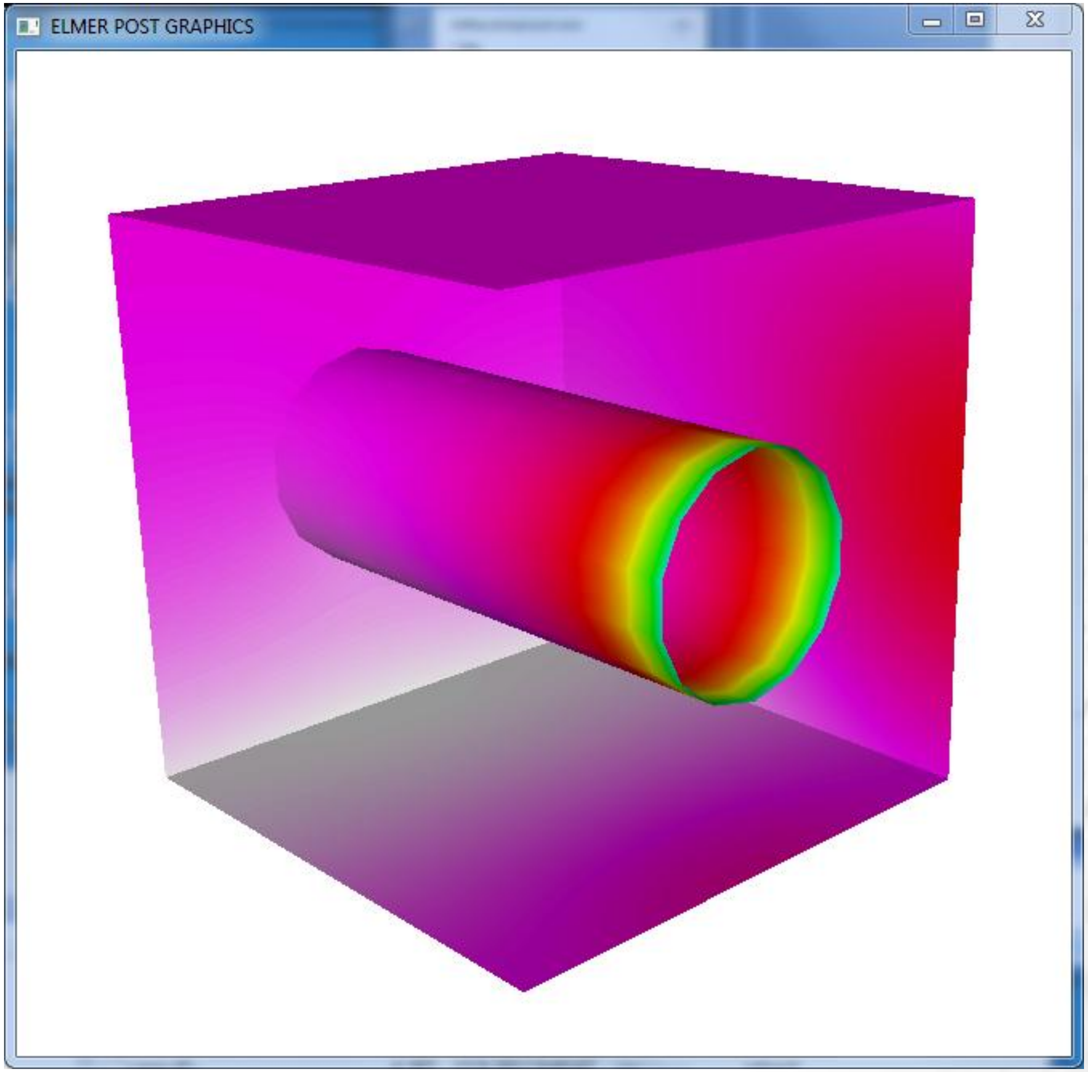
Contour Variable: nodes_x

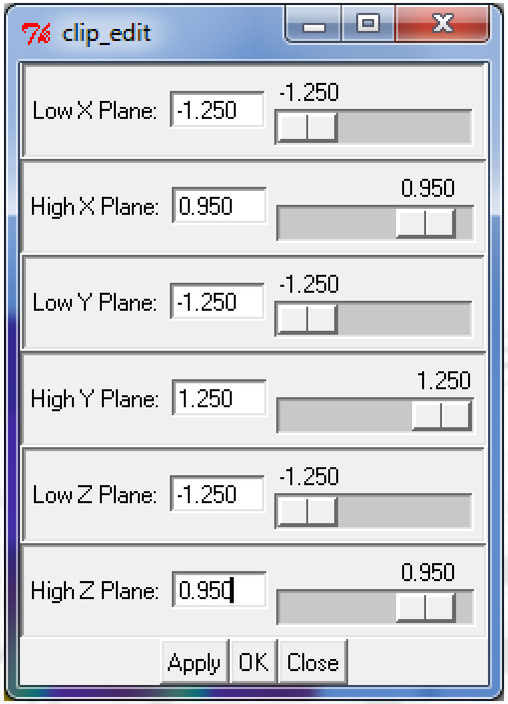
Color Variable: Temperature

Min: 0.0 Max: 1.0 ☐ Keep

Surface Normal Variable: none

Using clip planes





Low X Plane: -1.250 -1.250

High X Plane: 0.950 0.950

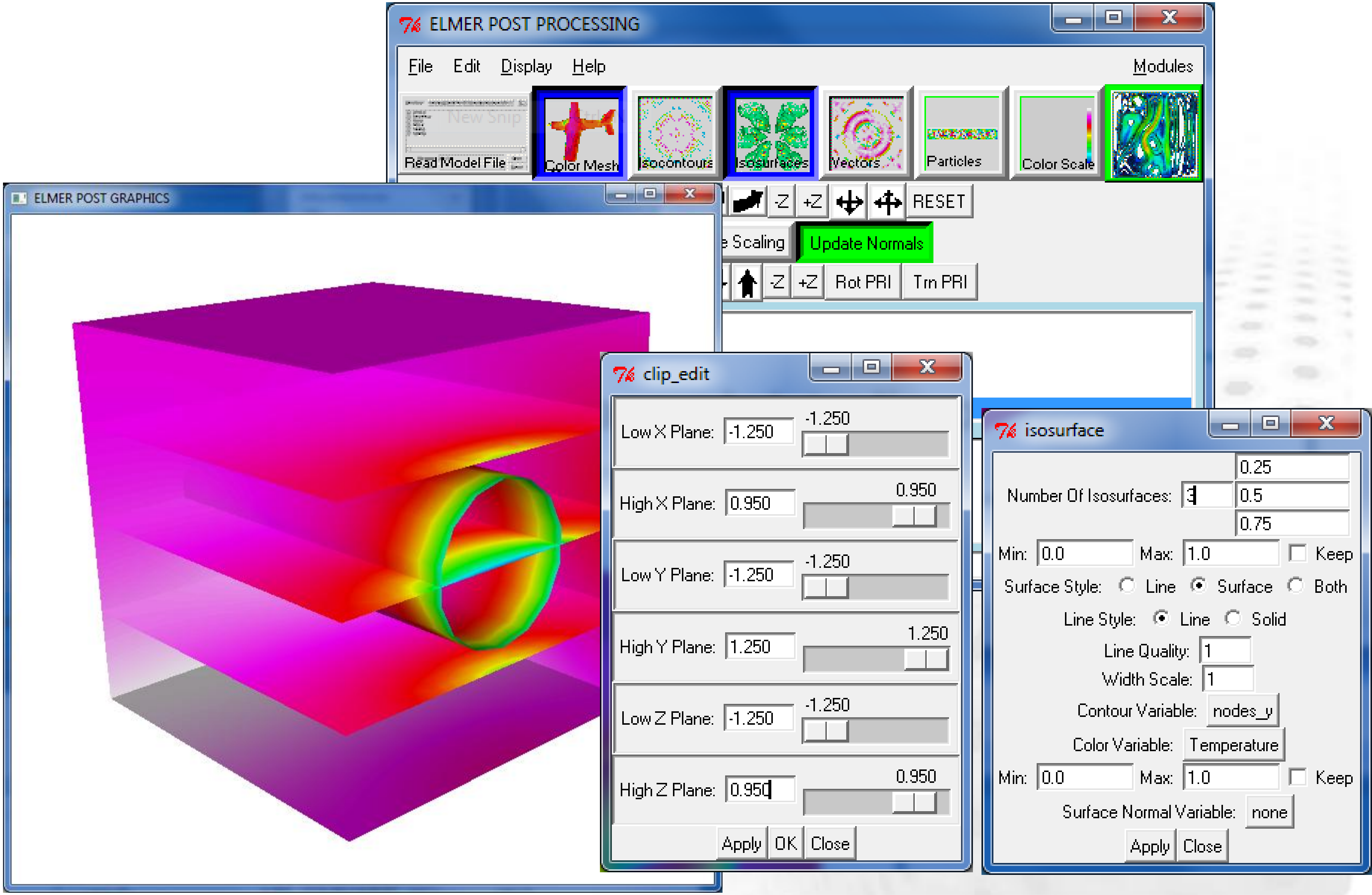
Low Y Plane: -1.250 -1.250

High Y Plane: 1.250 1.250

Low Z Plane: -1.250 -1.250

High Z Plane: 0.950 0.950

Isosurface + surface plot + clip planes



clip_edit

Low X Plane: -1.250 (slider -1.250)

High X Plane: 0.950 (slider 0.950)

Low Y Plane: -1.250 (slider -1.250)

High Y Plane: 1.250 (slider 1.250)

Low Z Plane: -1.250 (slider -1.250)

High Z Plane: 0.950 (slider 0.950)

isosurface

Number Of Isosurfaces: 3 — 0.25, 0.5, 0.75

Min: 0.0 Max: 1.0 ☐ Keep

Surface Style: ○ Line ● Surface ○ Both

Line Style: ● Line ○ Solid

Line Quality: 1

Width Scale: 1

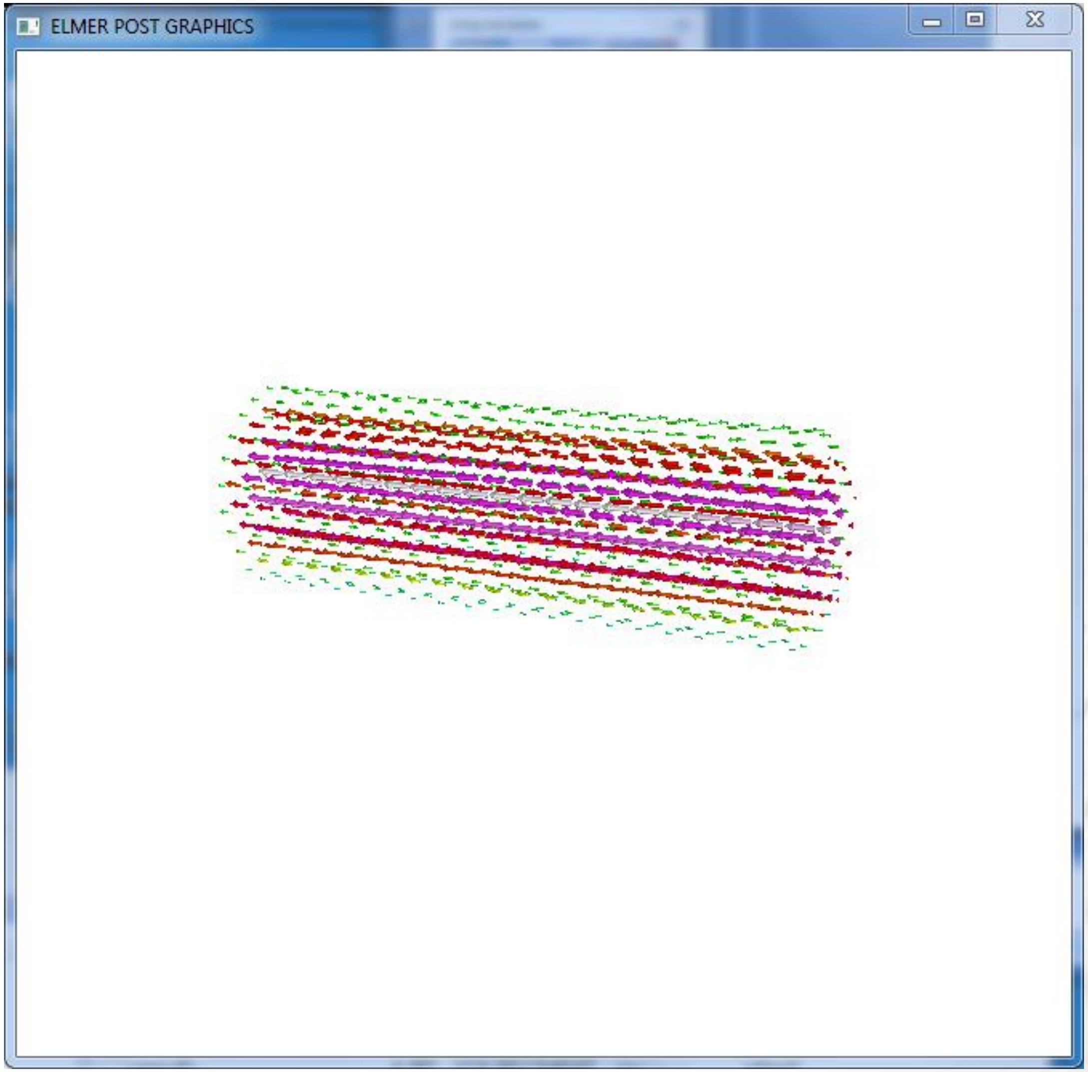
Contour Variable: nodes_y

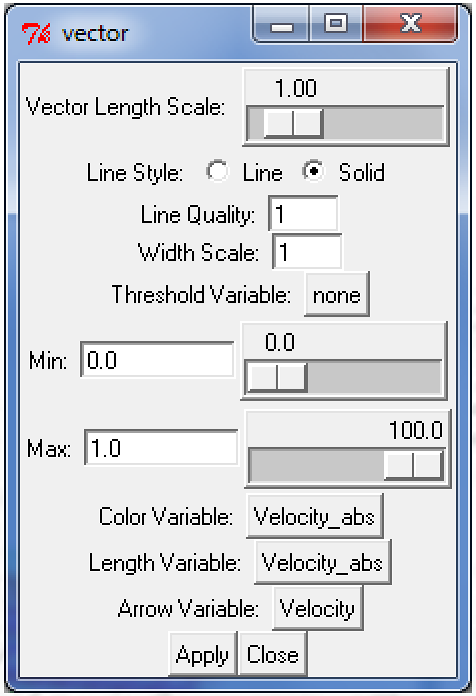
Color Variable: Temperature

Min: 0.0 Max: 1.0 ☐ Keep

Surface Normal Variable: none

Vector plots





Vector Length Scale: 1.00

Line Style: ○ Line ● Solid

Line Quality: 1

Width Scale: 1

Threshold Variable: none

Min: 0.0 0.0

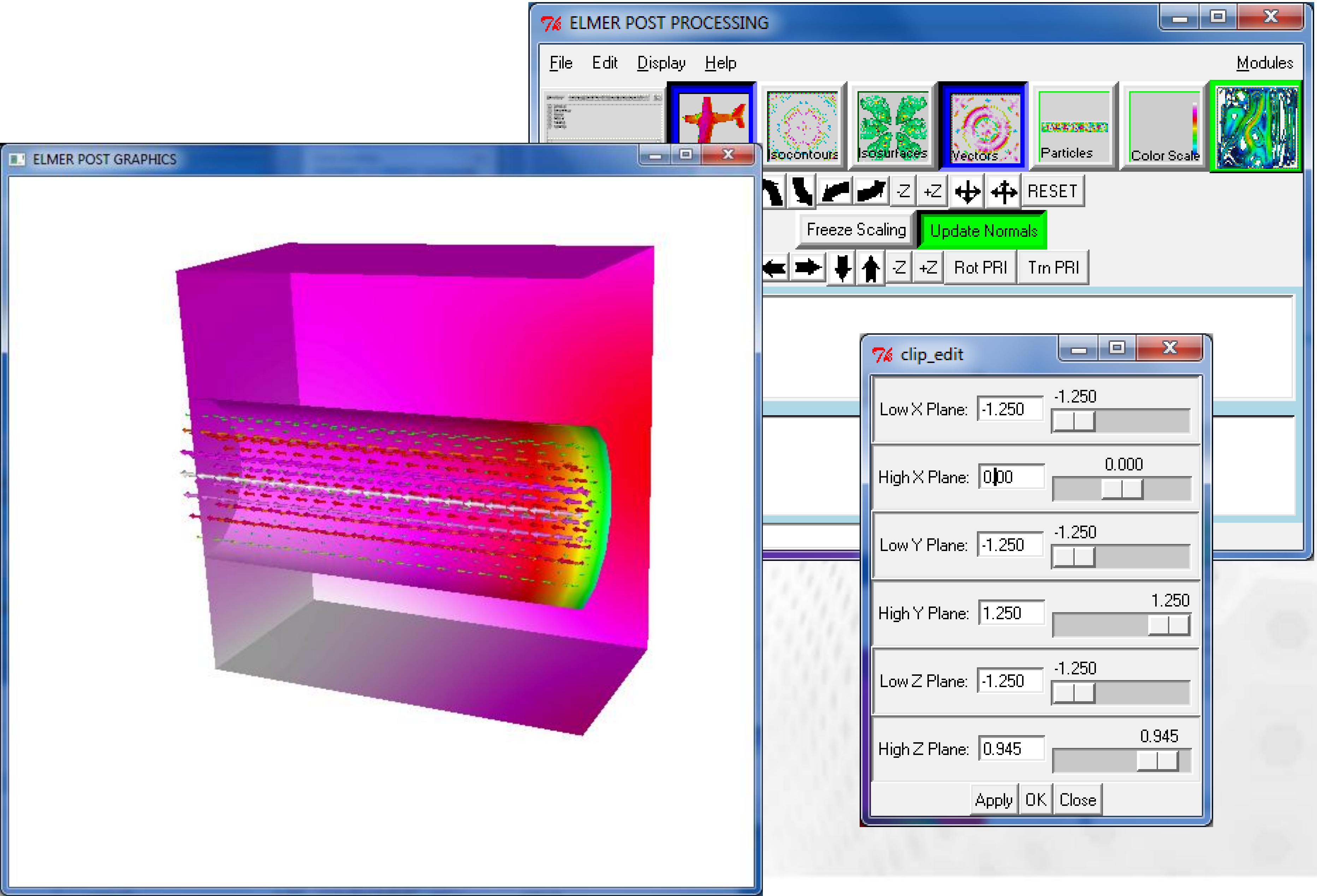
Max: 1.0 100.0

Color Variable: Velocity_abs

Length Variable: Velocity_abs

Arrow Variable: Velocity

Vector plot + solid surface



Low X Plane: -1.250 -1.250

High X Plane: 0.000 0.000

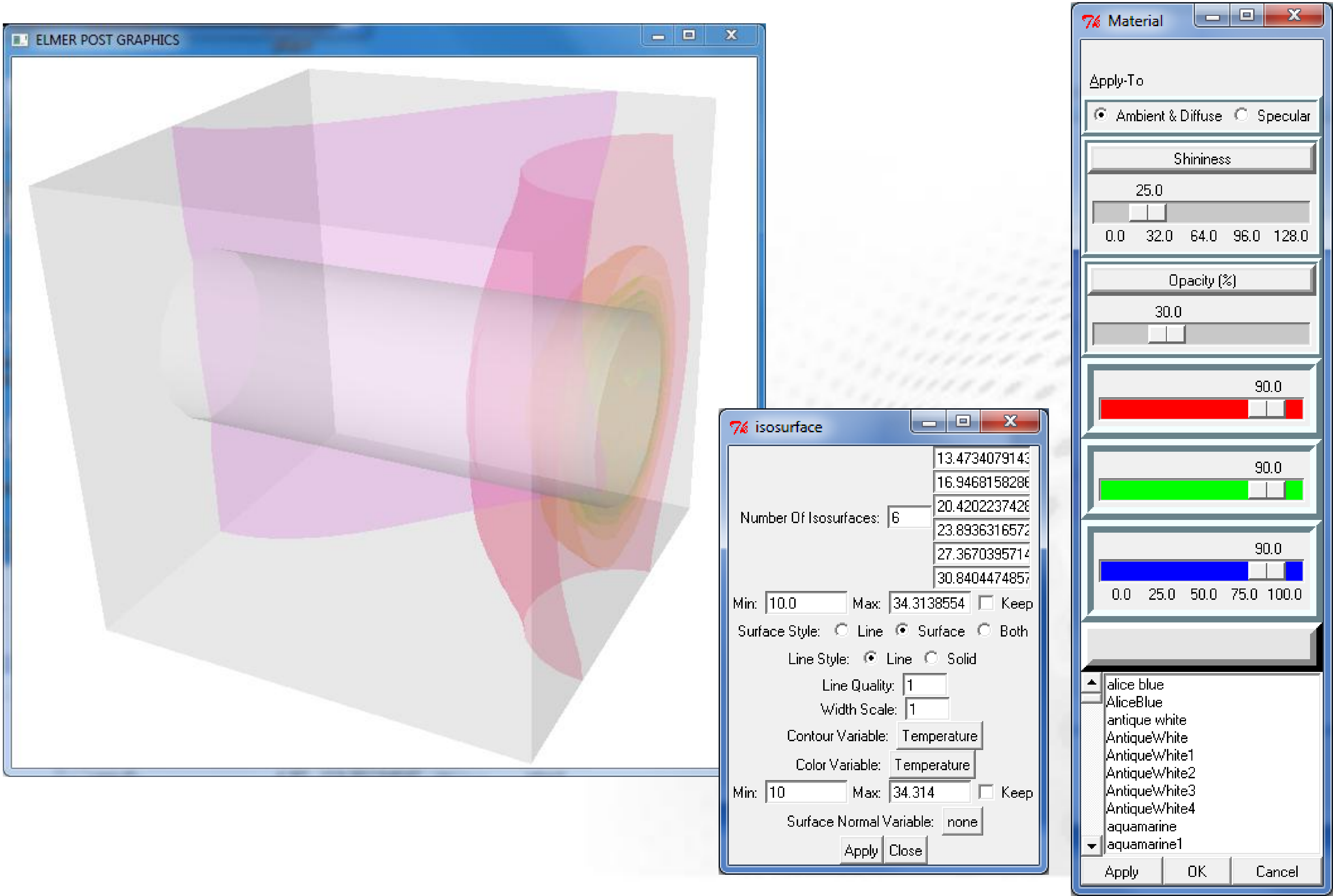
Low Y Plane: -1.250 -1.250

High Y Plane: 1.250 1.250

Low Z Plane: -1.250 -1.250

High Z Plane: 0.945 0.945

Surface plot + Isosurfaces + Opaque



isosurface

Number Of Isosurfaces: 6

13.4734079143
16.9468158286
20.4202237428
23.8936316572
27.3670395714
30.8404474857

Min: 10.0 Max: 34.3138554 ☐ Keep

Surface Style: ○ Line ● Surface ○ Both

Line Style: ● Line ○ Solid

Line Quality: 1

Width Scale: 1

Contour Variable: Temperature

Color Variable: Temperature

Min: 10 Max: 34.314 ☐ Keep

Surface Normal Variable: none

Apply Close

Material

Apply-To

● Ambient & Diffuse ○ Specular

Shininess

25.0

0.0 32.0 64.0 96.0 128.0

Opacity (%)

30.0

90.0

90.0

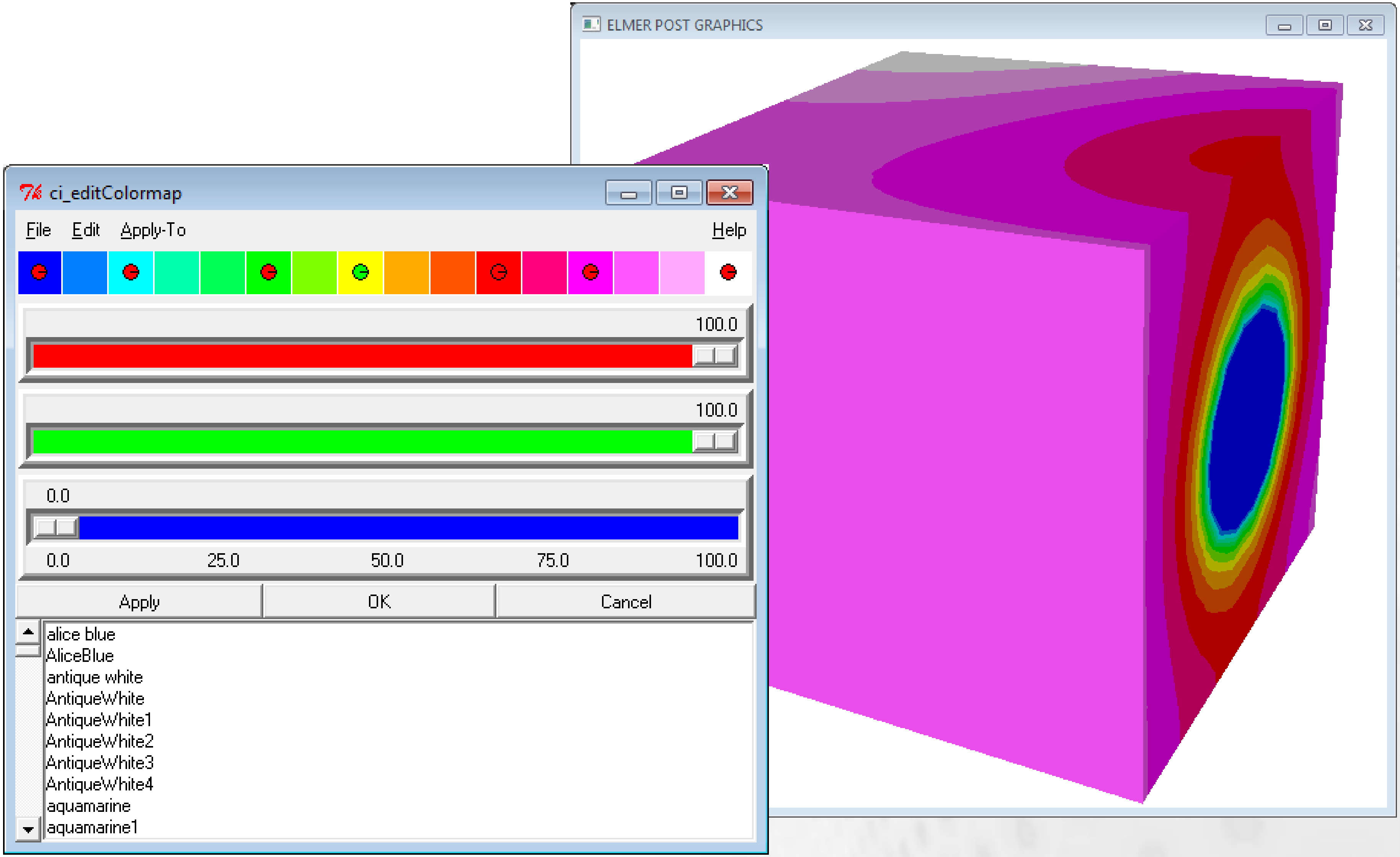
90.0

0.0 25.0 50.0 75.0 100.0

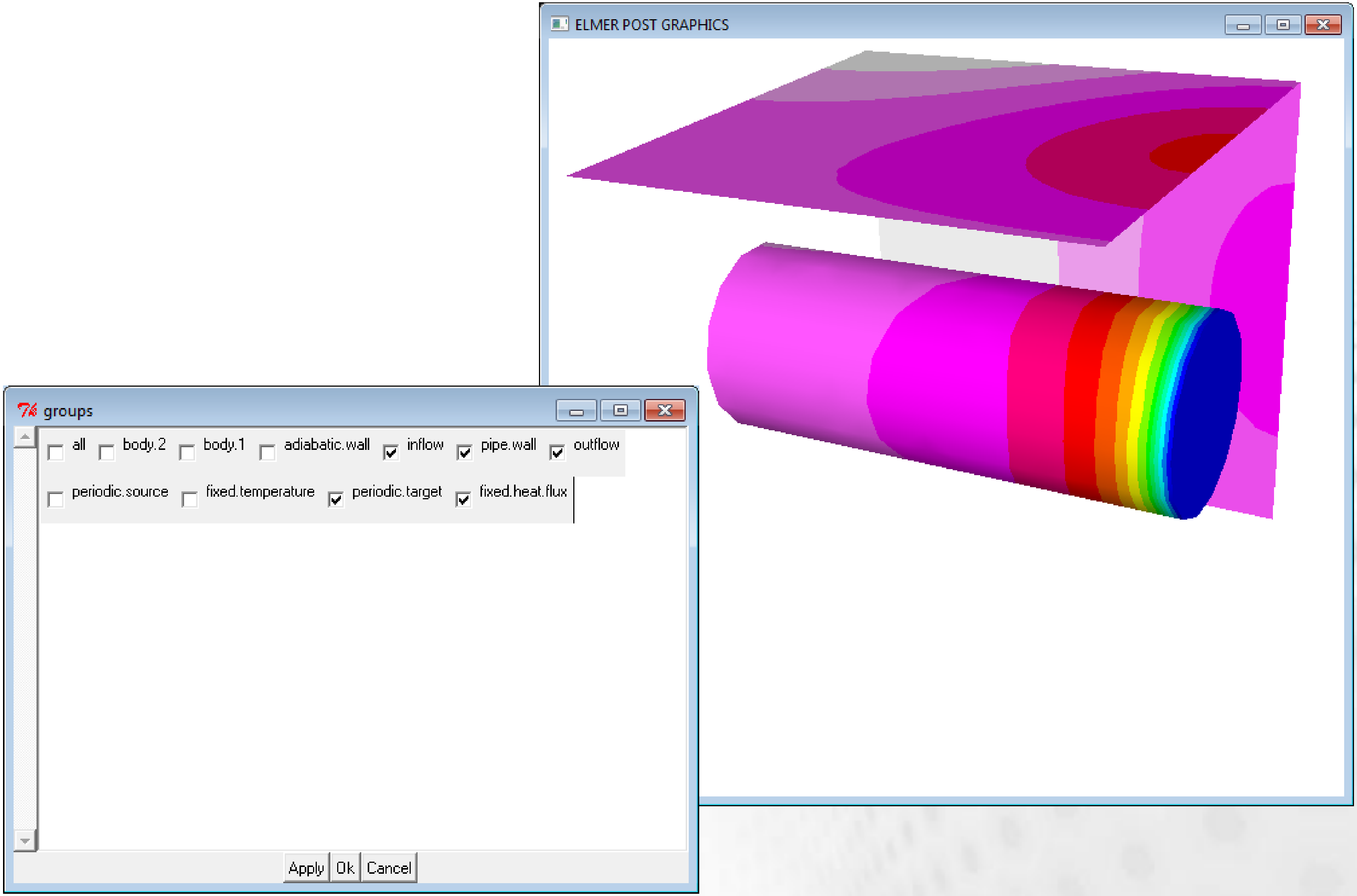
alice blue
AliceBlue
antique white
AntiqueWhite
AntiqueWhite1
AntiqueWhite2
AntiqueWhite3
AntiqueWhite4
aquamarine
aquamarine1

Apply OK Cancel

Change of colormap



Selecting active geometric entities

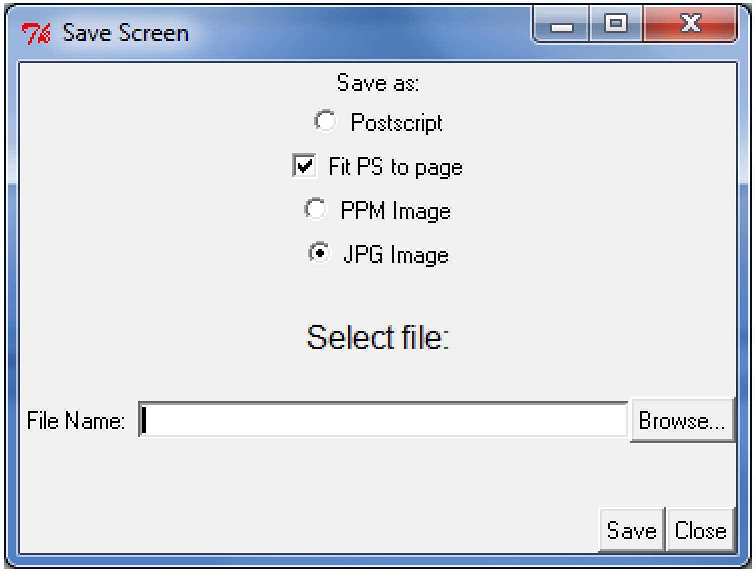


☐ all ☐ body.2 ☐ body.1 ☐ adiabatic.wall ☑ inflow ☑ pipe.wall ☑ outflow

☐ periodic.source ☐ fixed.temperature ☑ periodic.target ☑ fixed.heat.flux

Saving figures

- File -> Save Image -> jpg



Save as:

○ Postscript

☑ Fit PS to page

○ PPM Image

● JPG Image

Select file:

File Name:

Deformation in geometry

- Assume displacement field in variable `”Displacement”`
- Set in command windows:

```
math n0=nodes
math nodes=n0+Displacement
```

- Replot

Visualization with Paraview

Exporting 2D/3D data: ResultOutputSolve

By setting suffix for **`Post File`** to **`.vtu`** paraview format is saved automatically.

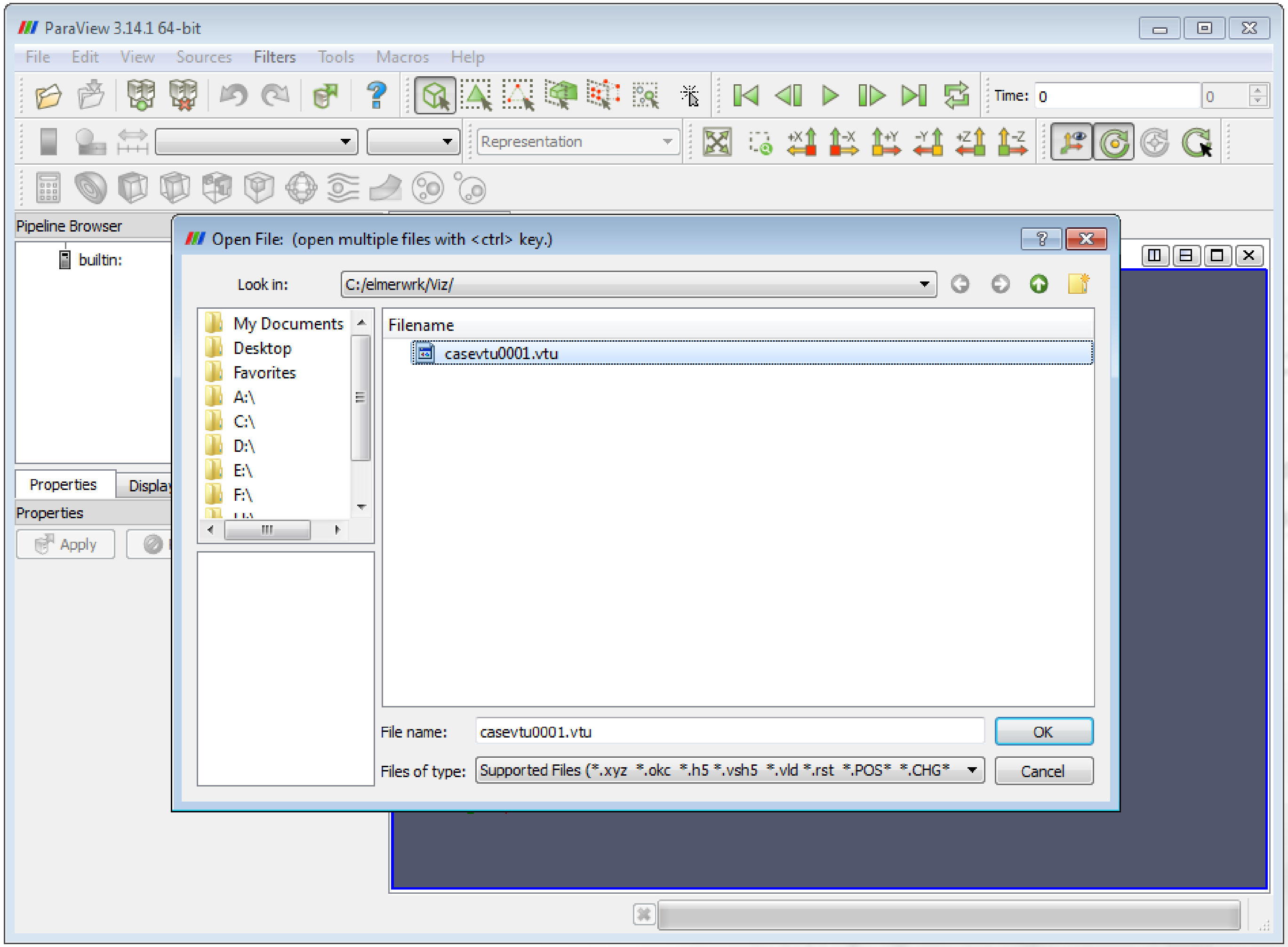
An example shows how to save data in unstructured XML VTK (.vtu) files to directory ”results” in single precision binary format.

```
Solver n
  Exec Solver = after timestep
  Equation = "result output"
  Procedure = "ResultOutputSolve‶ "ResultOutputSolver"
  Output File Name = "case"
  Output Format = String ”vtu”
  Binary Output = True
  Single Precision = True
  Save Geometry Ids = True
End
```

Filename conventions

- Suffix of unstructured XML based VTU file is `.vtu`
- Timesteps numbered `#step`
- Partitions numbered with `#part`par`#step`
- Holder for vtu files in parallel is `.pvtu`

Loading data



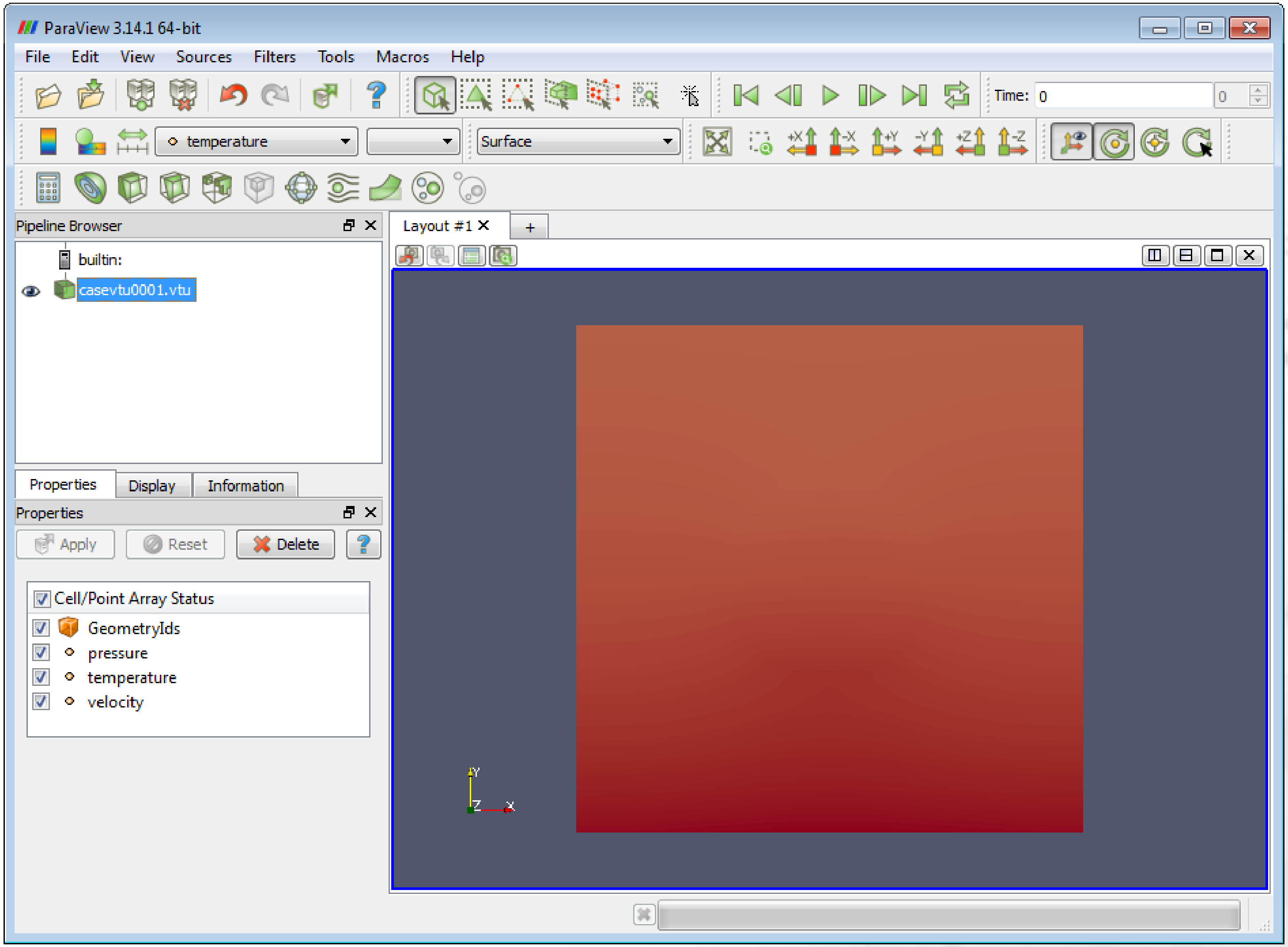
Look in: C:/elmerwrk/Viz/

File name: casevtu0001.vtu

Files of type: Supported Files (*.xyz *.okc *.h5 *.vsh5 *.vld *.rst *.POS* *.CHG*

Note: Paraview may have several datasets at the same time!

Solid color



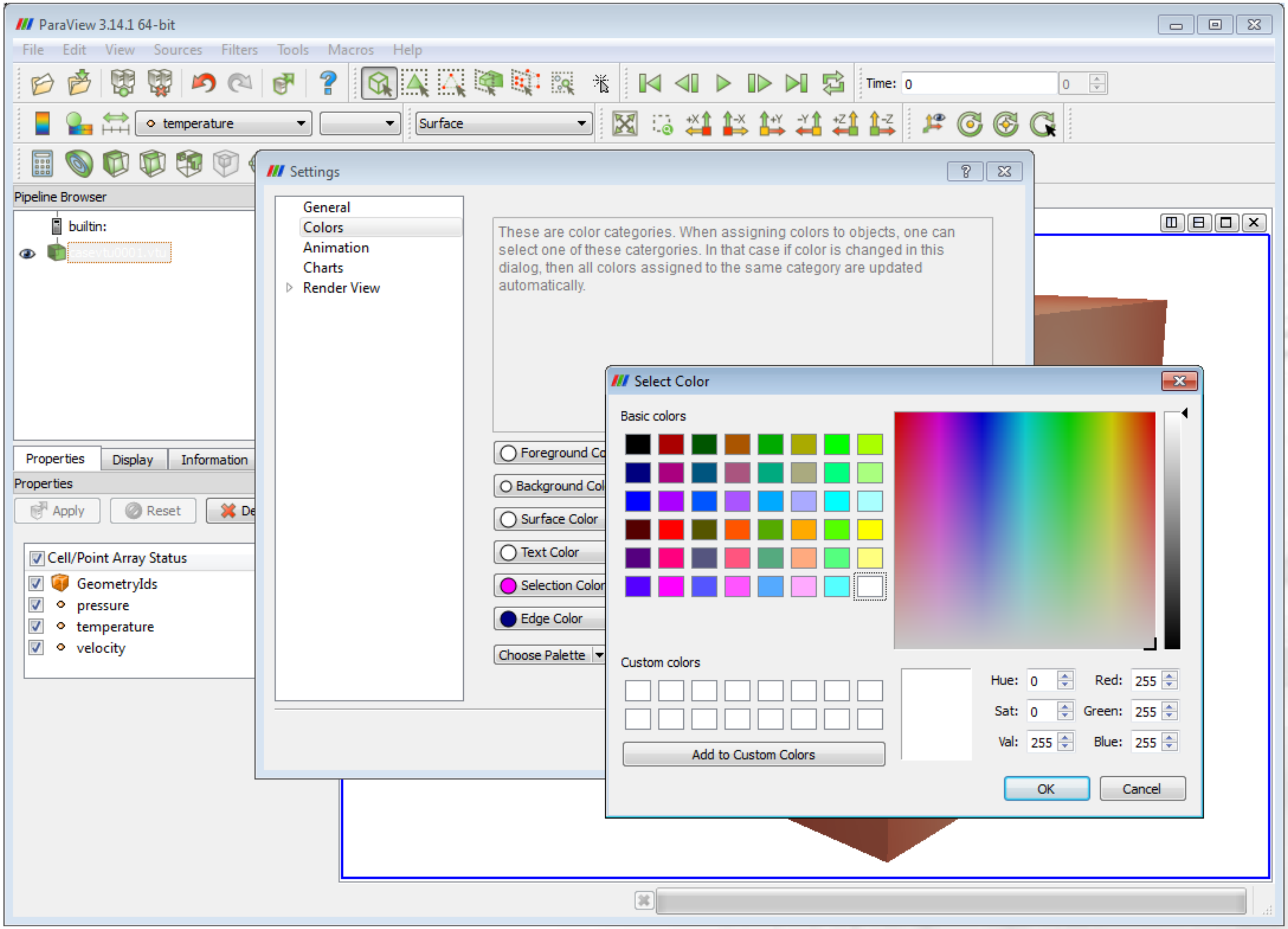
Properties

☑ Cell/Point Array Status
☑ GeometryIds
☑ pressure
☑ temperature
☑ velocity

Moving object in Paraview

- Rotate
  - Mouse: Left bottom
- Scale
  - Mouse: Right bottom
- Translate
  - Mouse: Center bottom

Setting background color



ParaView 3.14.1 64-bit

Settings

These are color categories. When assigning colors to objects, one can select one of these catergories. In that case if color is changed in this dialog, then all colors assigned to the same category are updated automatically.

Time: 0

Cell/Point Array Status

☑ GeometryIds

☑ pressure

☑ temperature

☑ velocity

Select Color

Hue: 0

Sat: 0

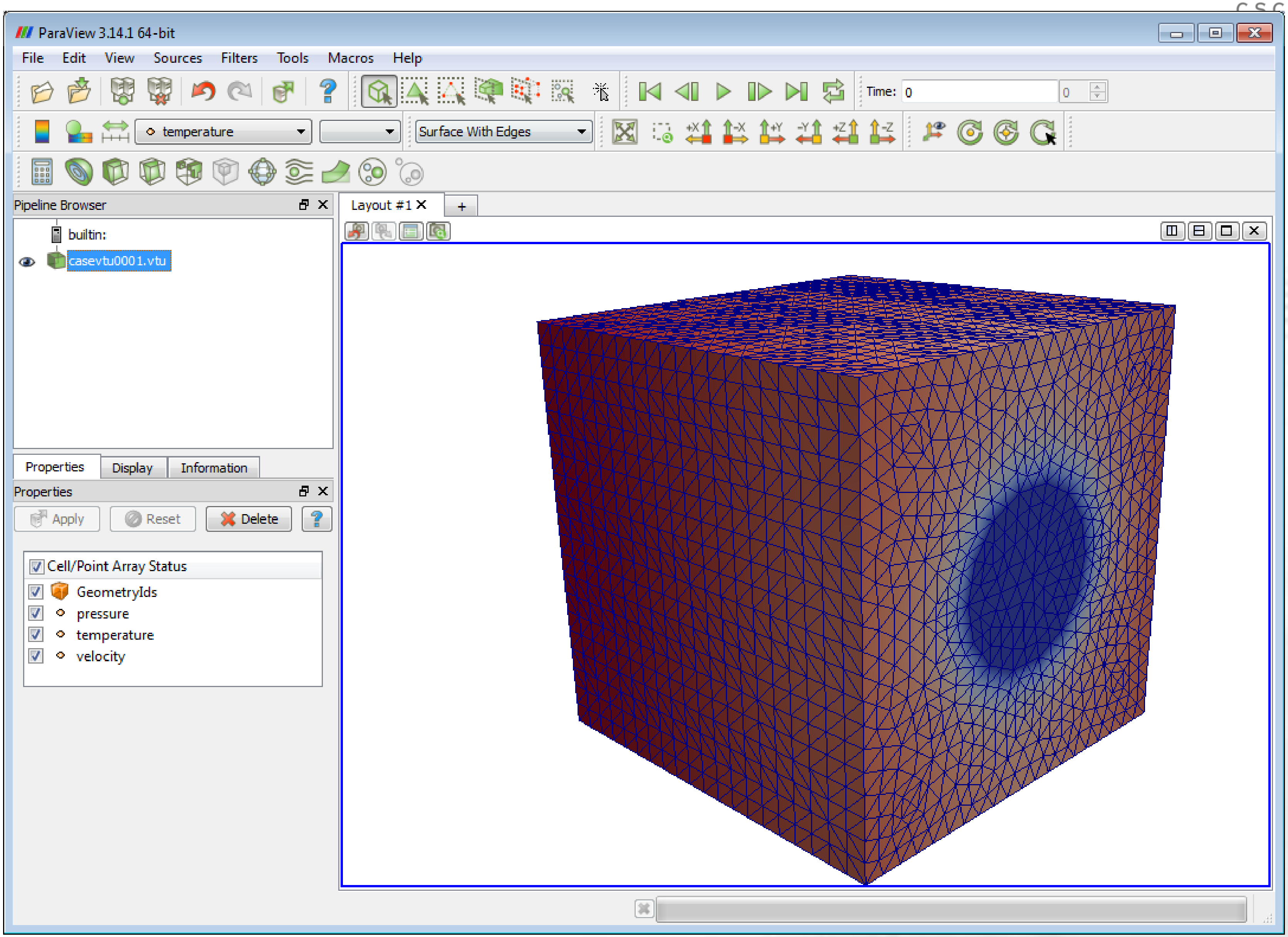
Val: 255

Red: 255

Green: 255

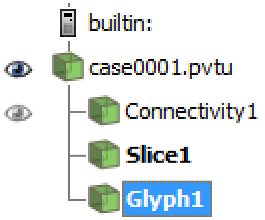
Blue: 255

Color mesh with surface + edges



☑ Cell/Point Array Status
- ☑ GeometryIds
- ☑ pressure
- ☑ temperature
- ☑ velocity

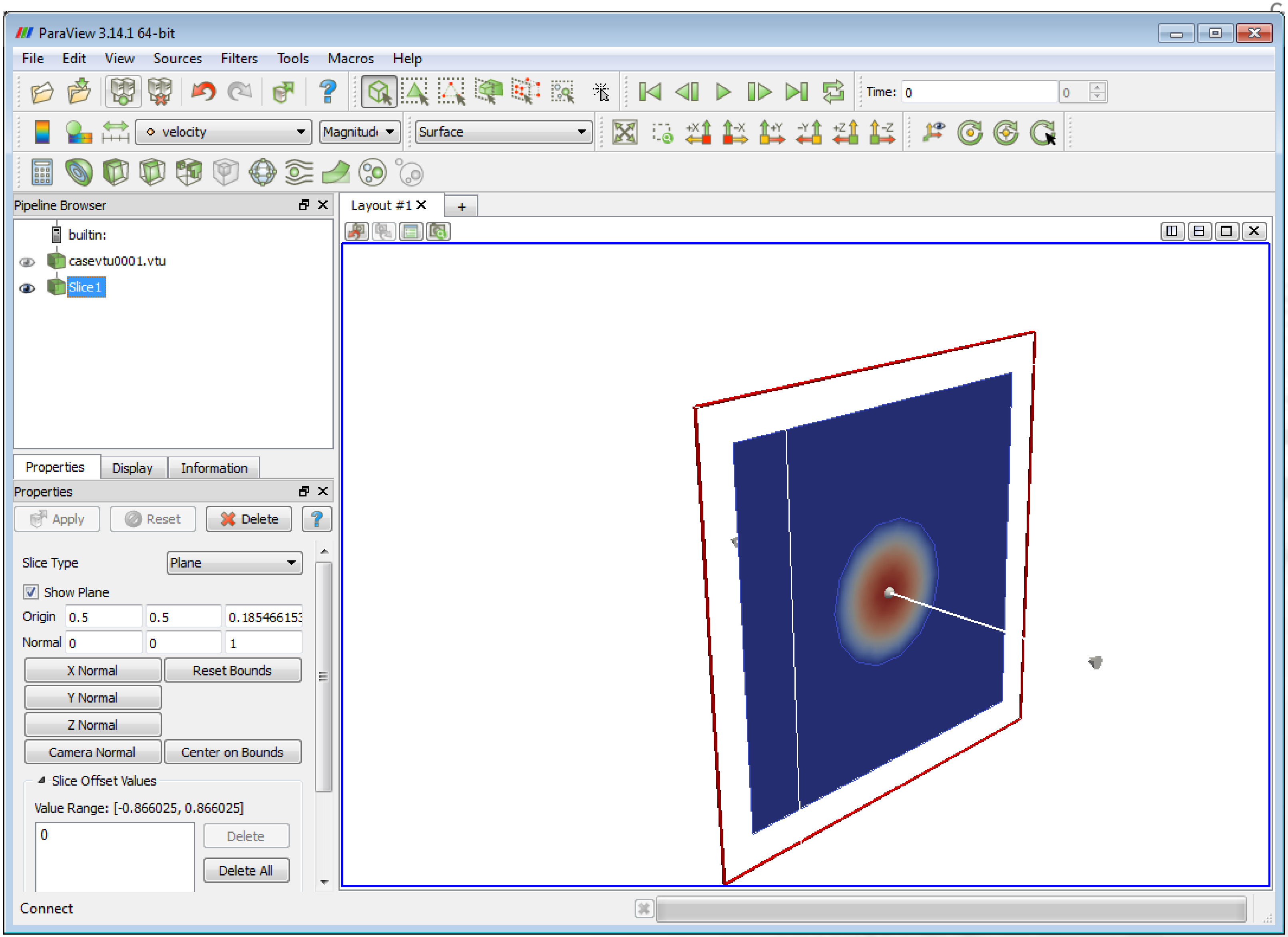
AMR Contour
AMR Dual Clip
Annotate Time Filter
Append Attributes
Append Datasets
Append Geometry
Block Scalars
Calculator
Cell Centers
Cell Data to Point Data
Clean
Clean Cells to Grid
Clean to Grid
Clip
Clip Closed Surface
Clip Generic Dataset
Compute Derivatives
Connectivity
Contingency Statistics
Contour
Contour Generic Dataset
Curvature
D3
Decimate
Delaunay 2D
Delaunay 3D
Descriptive Statistics
Elevation
Extract AMR Blocks
Extract Block
Extract CTH Parts
Extract Cells By Region
Extract Edges
Extract Generic Dataset Surface
Extract Level
Extract Selection
Extract Subset
Extract Surface
FFT Of Selection Over Time
FOF/SOD Halo Finder
Feature Edges
Gaussian Resampling
Generate Ids
Generate Quadrature Points
Generate Quadrature Scheme Dictionary
Generate Surface Normals
Glyph
Glyph With Custom Source
Gradient
Gradient Of Unstructured DataSet
Grid Connectivity
Group Datasets
Histogram
Image Data to Point Set
Integrate Variables
Interpolate to Quadrature Points
Intersect Fragments
Iso Volume
K Means
Level Scalars
Linear Extrusion
Loop Subdivision
Mask Points
Material Interface Filter
Median
Merge Blocks
Mesh Quality
Multicorrelative Statistics
Normal Glyphs
Octree Depth Limit
Octree Depth Scalars
Outline
Outline Corners
Outline Curvilinear DataSet
Particle Pathlines
ParticleTracer
Plot Data
Plot Global Variables Over Time
Plot On Intersection Curves
Plot On Sorted Lines
Plot Over Line
Plot Selection Over Time
Point Data to Cell Data
Principal Component Analysis
Probe Location
Process Id Scalars
Programmable Filter
Python Calculator
Quadric Clustering
Random Vectors
Rectilinear Data to Point Set
Rectilinear Grid Connectivity
Reflect
Resample With Dataset
Ribbon
Rotational Extrusion
Scatter Plot
Shrink
Slice
Slice Generic Dataset
Smooth
Stream Tracer
Stream Tracer For Generic Datasets
Stream Tracer With Custom Source
Subdivide
Surface Flow
Surface Vectors
Table To Points
Table To Structured Grid
Temporal Cache
Temporal Interpolator
Temporal Shift Scale
Temporal Snap-to-Time-Step
Temporal Statistics
Tessellate
Tetrahedralize
Texture Map to Cylinder
Texture Map to Plane
Texture Map to Sphere
Threshold
Transform
Triangle Strips
Triangulate
Tube
Warp By Scalar
Warp By Vector
Youngs Material Interface



builtin:
case0001.pvtu
Connectivity1
Slice1
Glyph1

- Paraview uses extensively *filters* to create new datasets
- Filters and datasets may be set active or passive by clicking the eye
- Several datasets may be visualized at the same time

Plotting a slice



Pipeline Browser:
- builtin:
- casevtu0001.vtu
- Slice1

Properties:

Slice Type: Plane

☑ Show Plane

Origin: 0.5 | 0.5 | 0.18546615

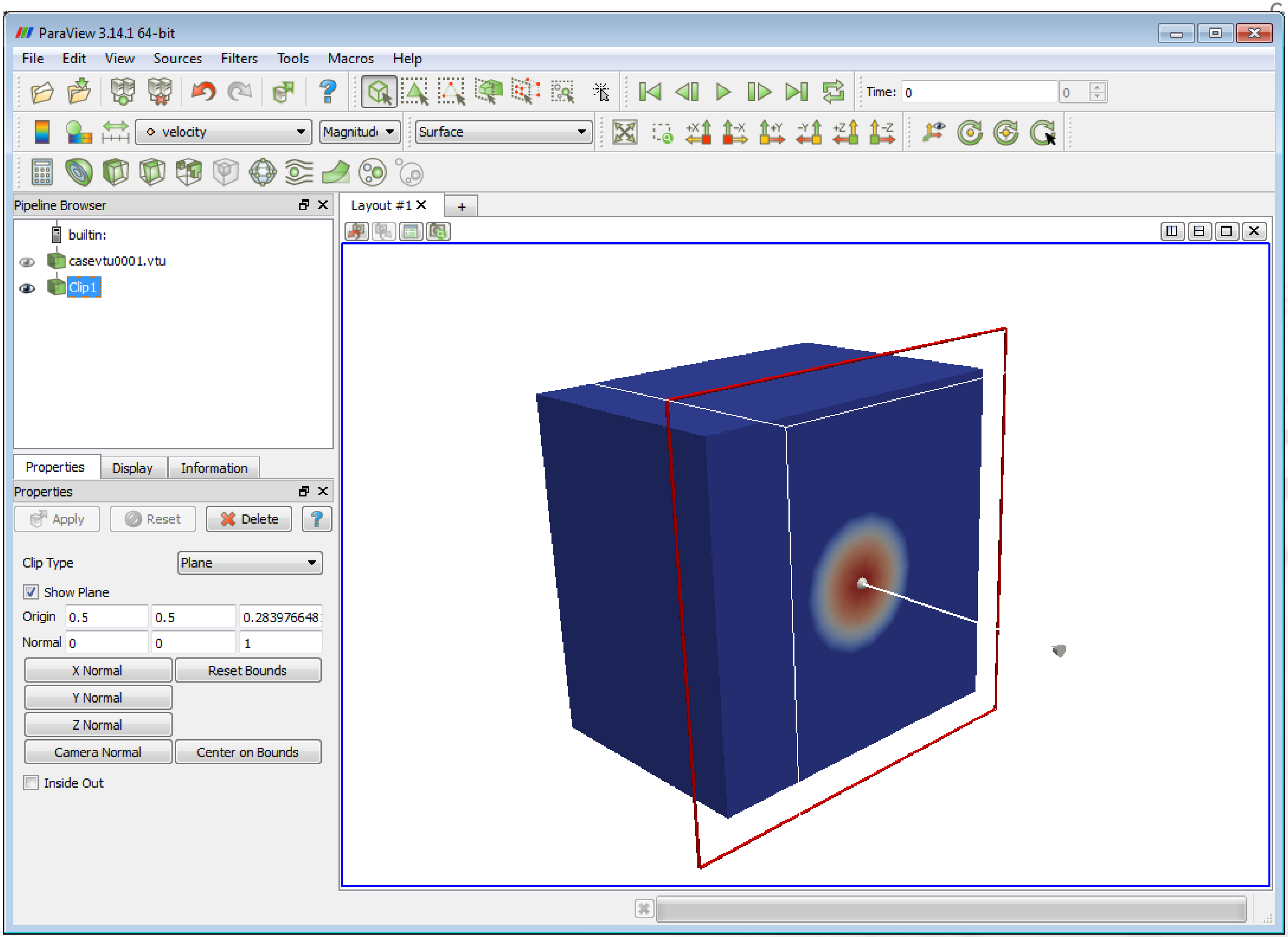
Normal: 0 | 0 | 1

Slice Offset Values

Value Range: [-0.866025, 0.866025]

0

Plotting a clip



Properties

Clip Type: Plane

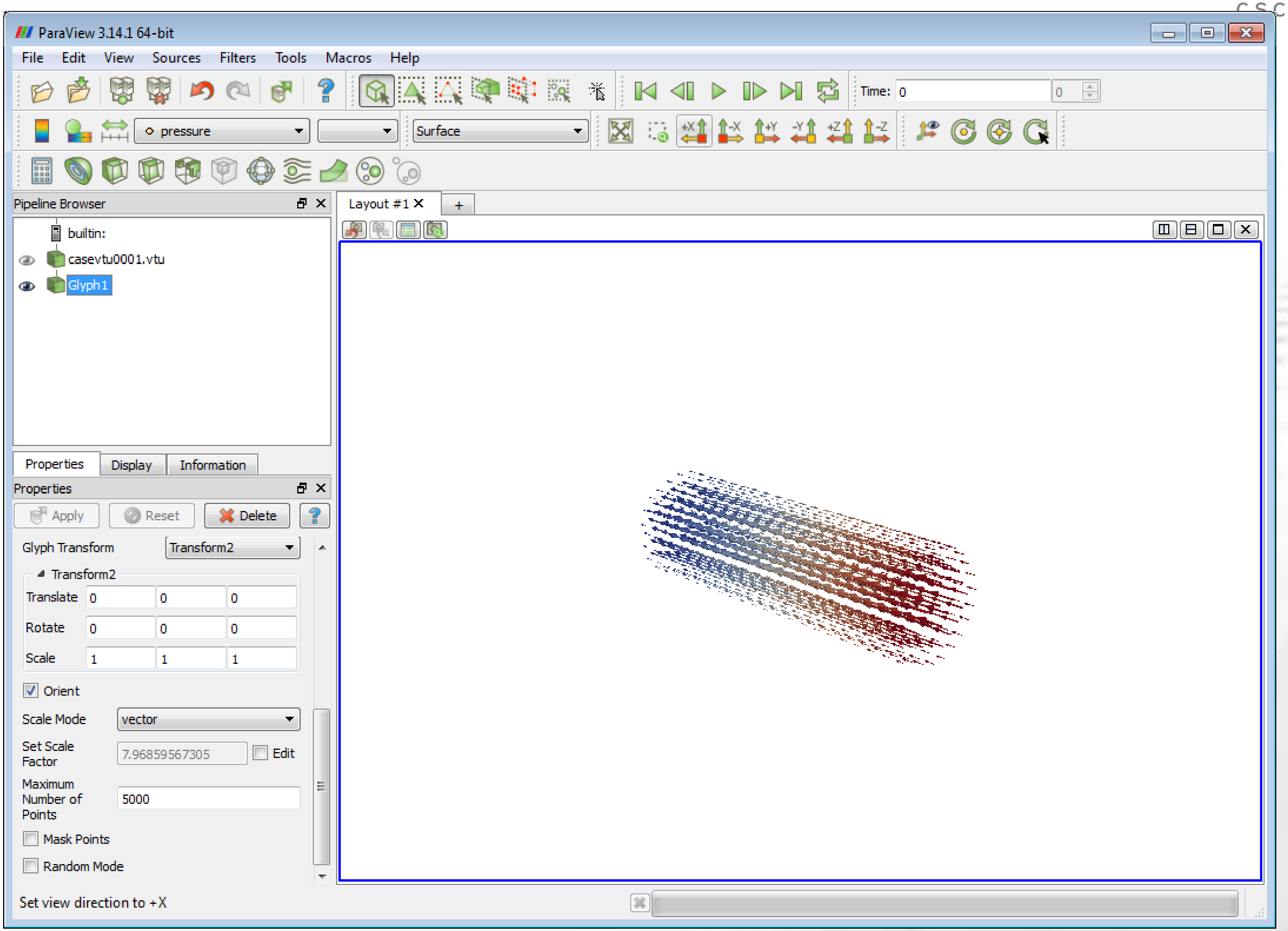
☑ Show Plane

Origin: 0.5 | 0.5 | 0.283976648

Normal: 0 | 0 | 1

☐ Inside Out

Vector plot



Time: 0

builtin:
casevtu0001.vtu
Glyph1

Glyph Transform: Transform2

Transform2

Translate	0	0	0
Rotate	0	0	0
Scale	1	1	1

☑ Orient

Scale Mode: vector

Set Scale Factor: 7.96859567305 ☐ Edit

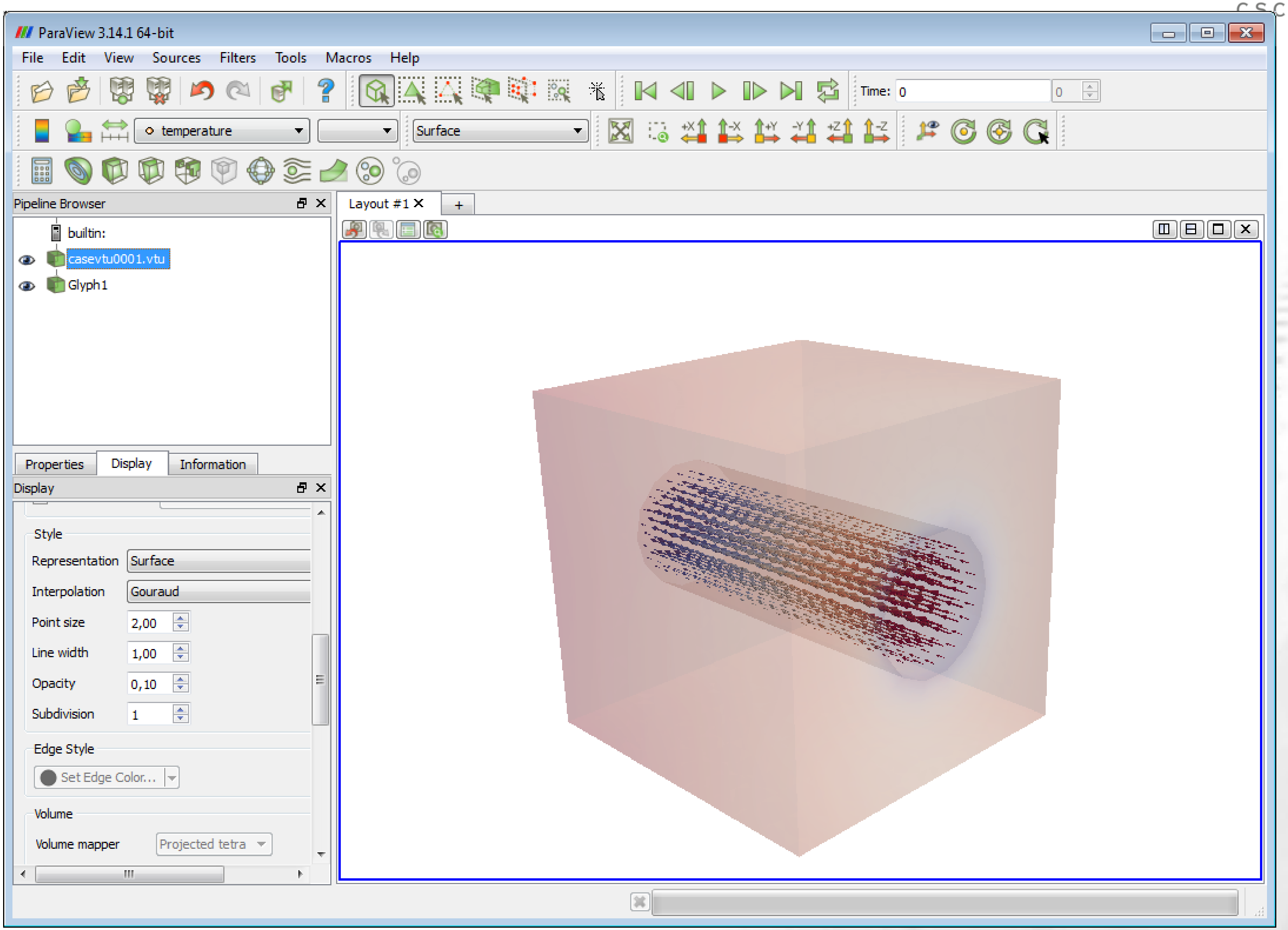
Maximum Number of Points: 5000

☐ Mask Points

☐ Random Mode

Set view direction to +X

Vector plot + opaque solid surface



Style

Representation Surface

Interpolation Gouraud

Point size 2,00

Line width 1,00

Opacity 0,10

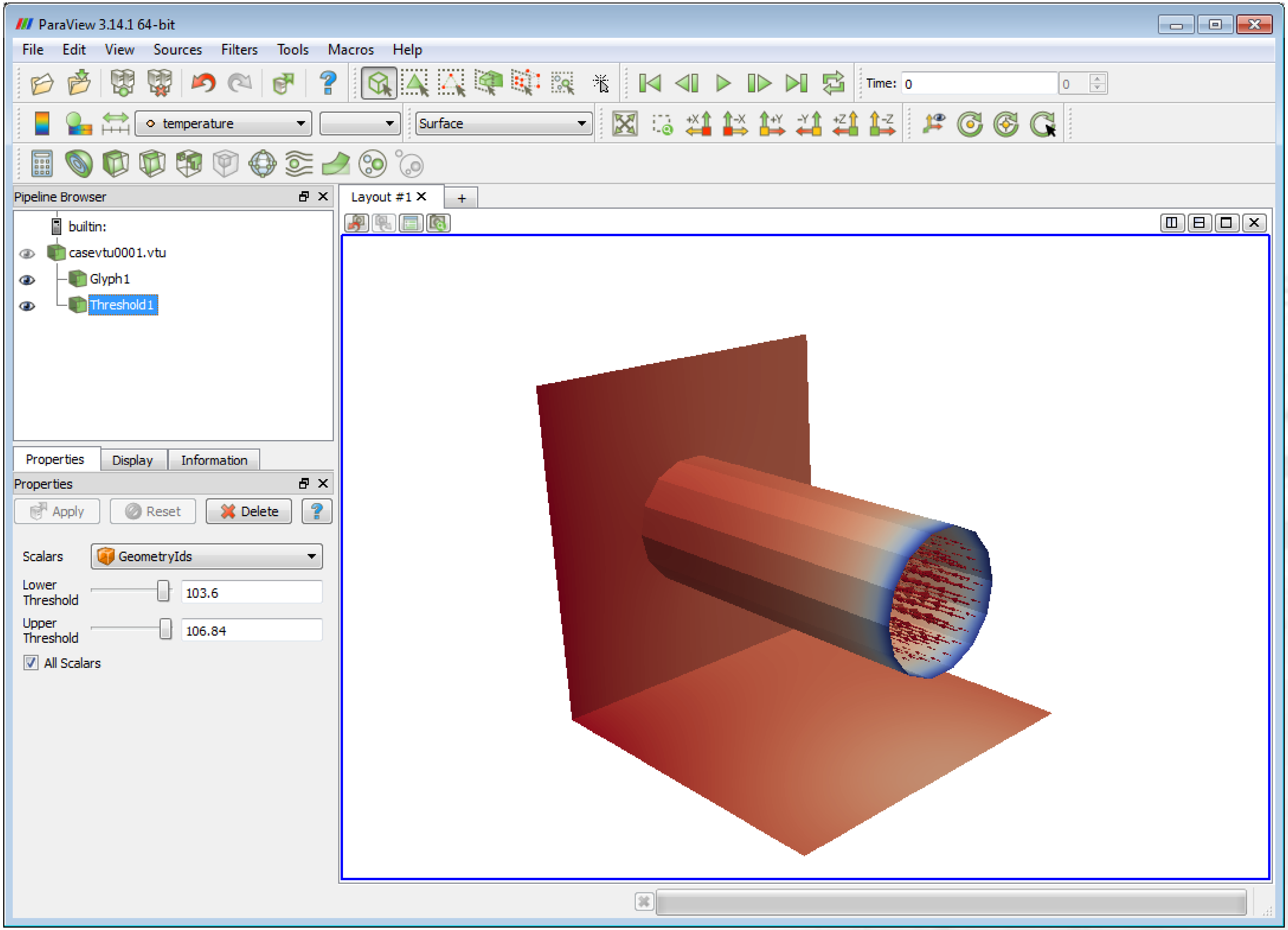
Subdivision 1

Edge Style

Volume

Volume mapper Projected tetra

Vector plot + solid surface with Id treshold



Pipeline Browser

- builtin:
  - casevtu0001.vtu
    - Glyph1
    - Threshold1

Time: 0

temperature

Surface

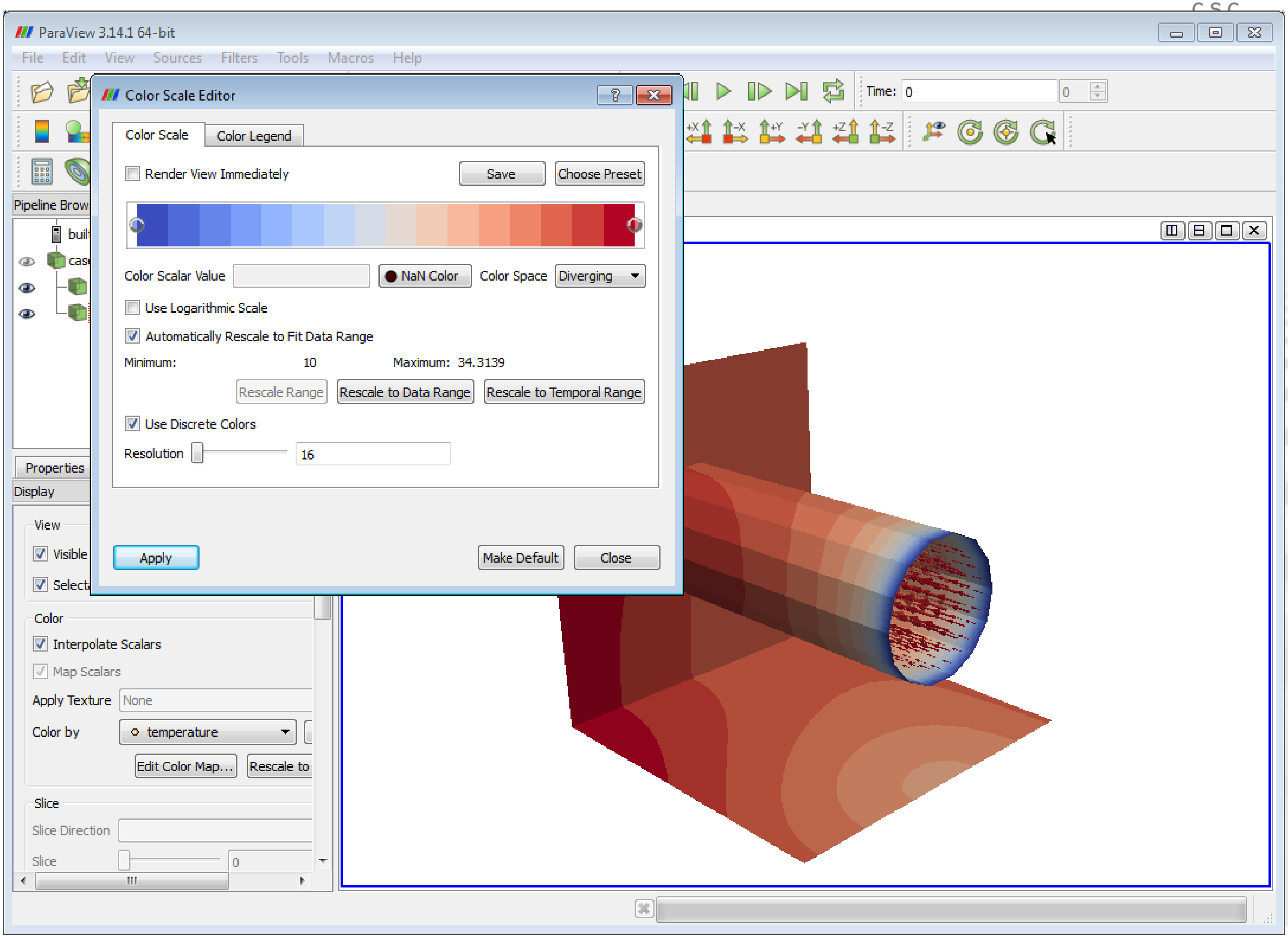
Scalars: GeometryIds

Lower Threshold: 103.6

Upper Threshold: 106.84

☑ All Scalars

Change of colormap



☐ Render View Immediately

Color Scalar Value

Color Space: Diverging

☐ Use Logarithmic Scale

☑ Automatically Rescale to Fit Data Range

Minimum: 10   Maximum: 34.3139

☑ Use Discrete Colors

Resolution: 16

View

☑ Visible

☑ Selectable

Color

☑ Interpolate Scalars

☑ Map Scalars

Apply Texture: None

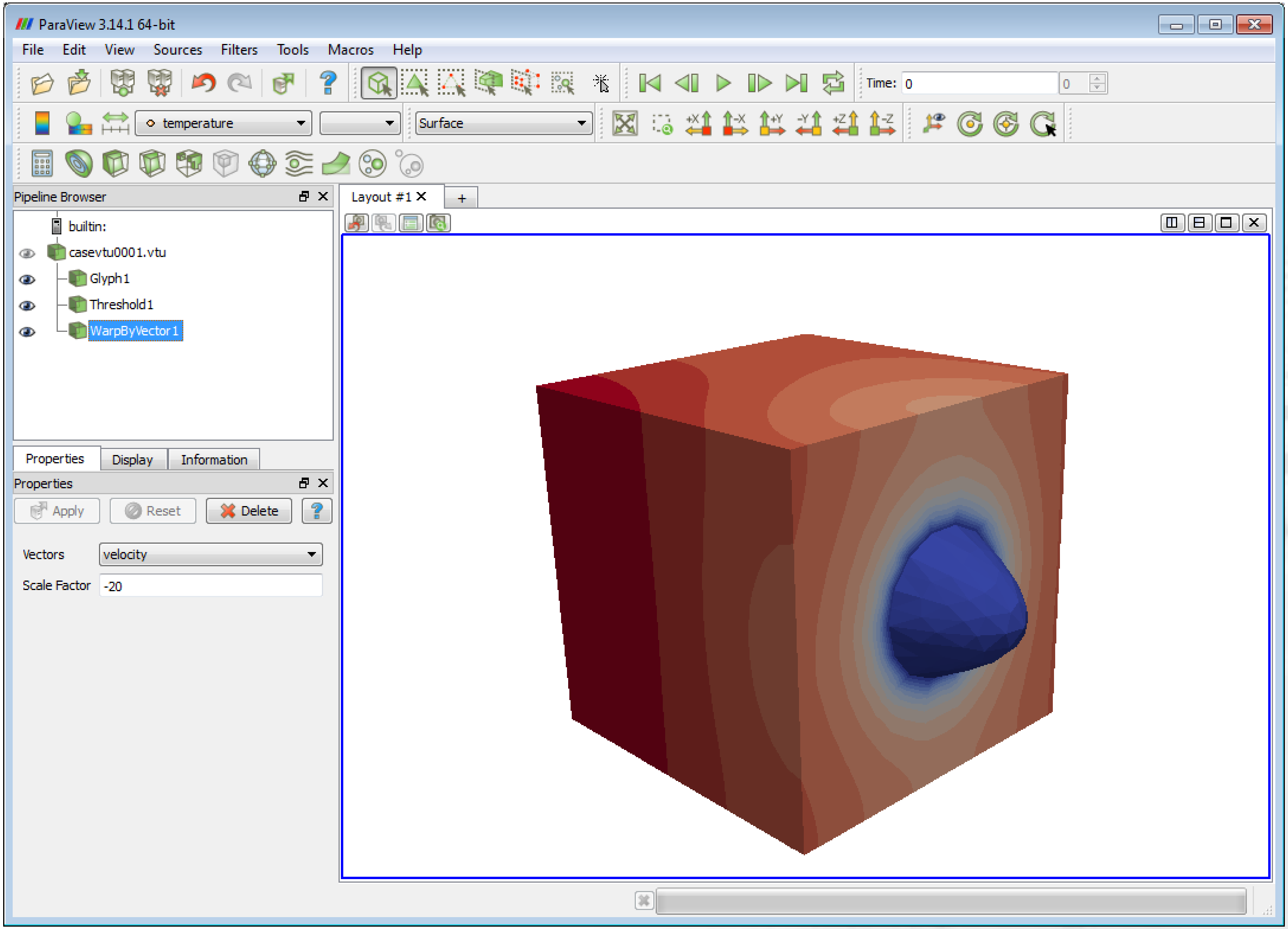
Color by: temperature

Slice

Slice Direction

Slice: 0

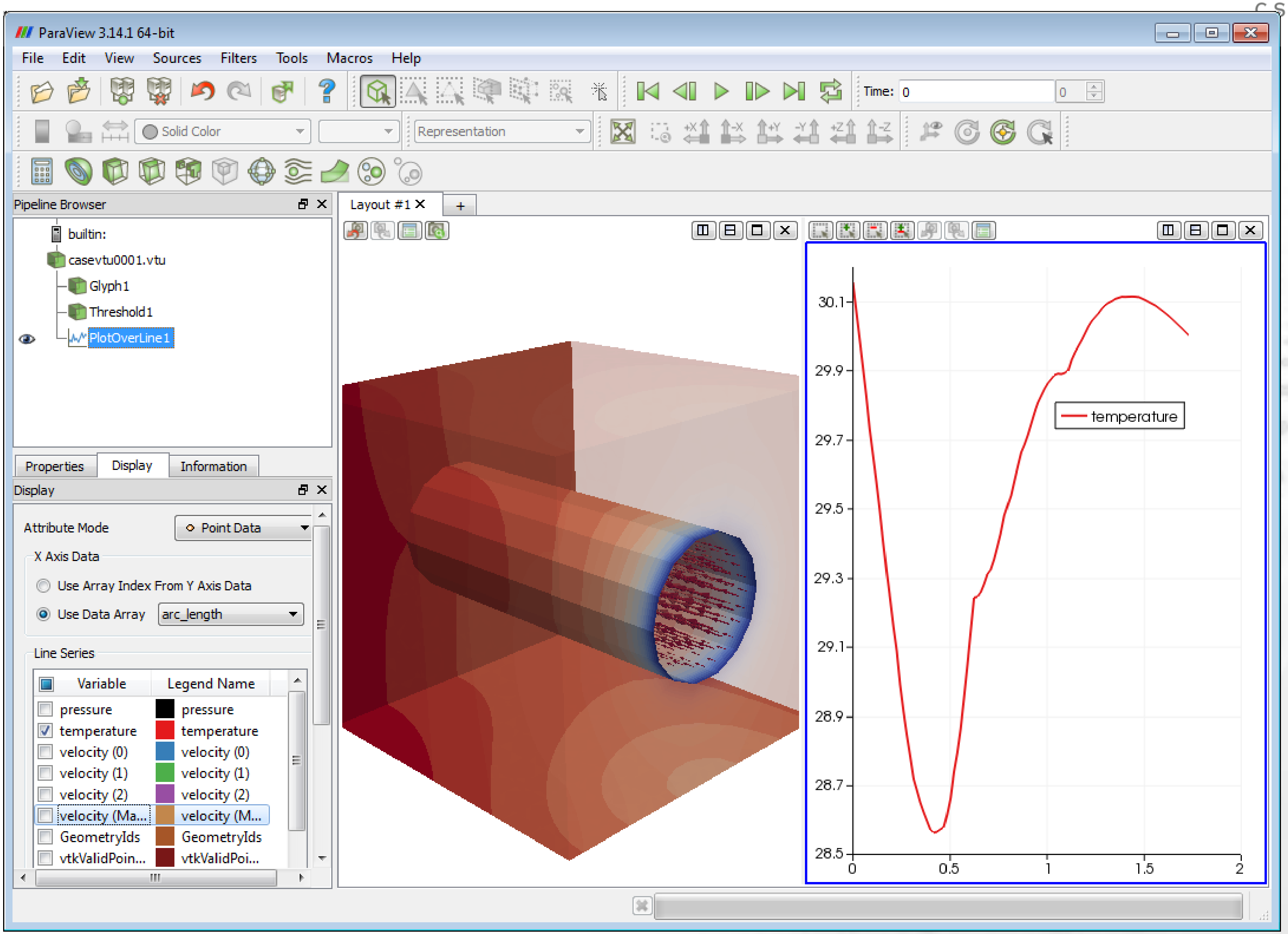
Deformation – WarpByVector filter



Vectors velocity

Scale Factor -20

Plot line – PlotOverLine filter



builtin:
casevtu0001.vtu
Glyph1
Threshold1
PlotOverLine1

Properties | Display | Information

Display

Attribute Mode: Point Data

X Axis Data

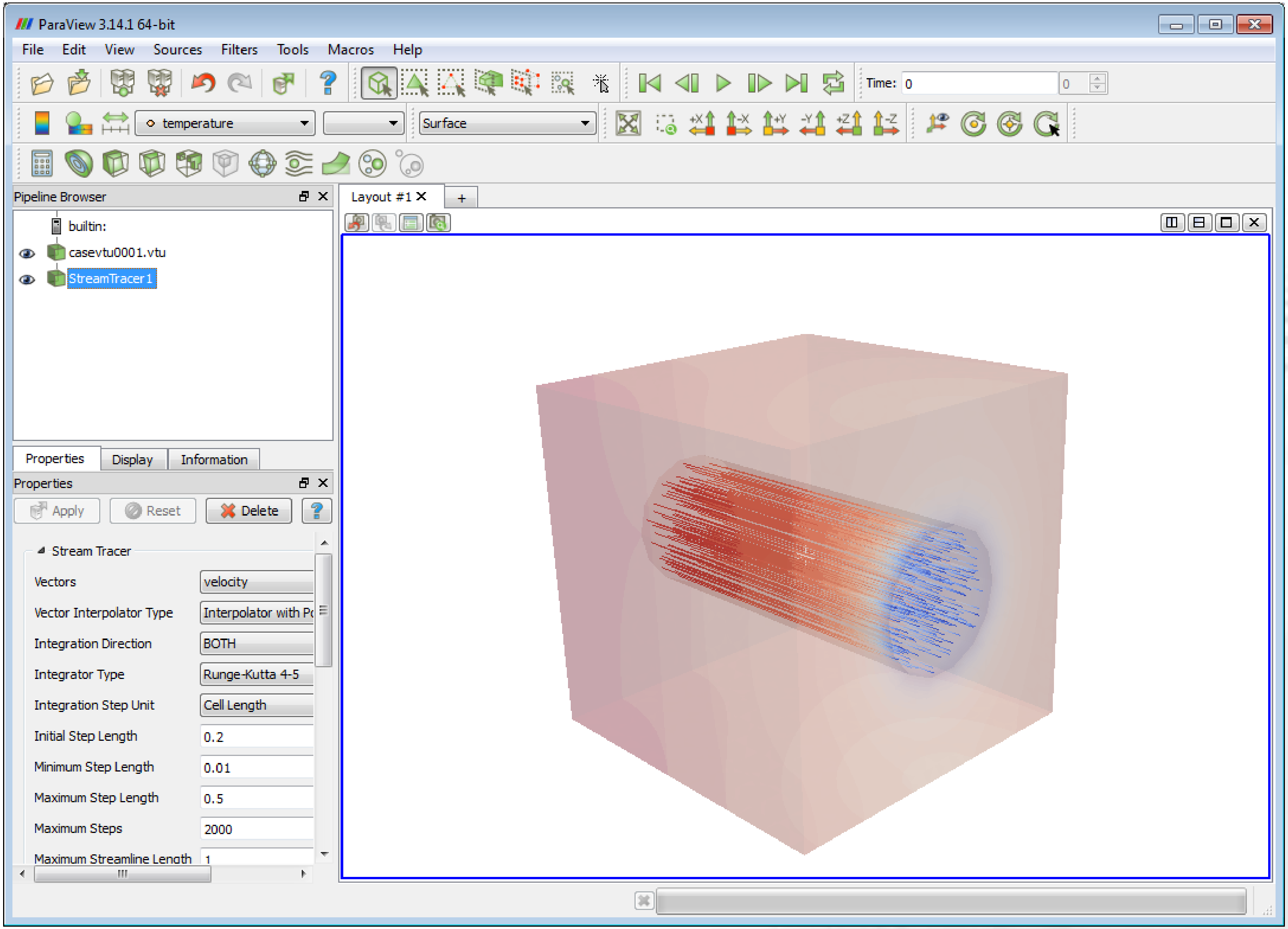
○ Use Array Index From Y Axis Data

● Use Data Array: arc_length

Line Series

☐ Variable	Legend Name
☐ pressure	■ pressure
☑ temperature	🟥 temperature
☐ velocity (0)	🟦 velocity (0)
☐ velocity (1)	🟩 velocity (1)
☐ velocity (2)	🟪 velocity (2)
☐ velocity (Ma...	🟫 velocity (M...
☐ GeometryIds	🟫 GeometryIds
☐ vtkValidPoin...	🟫 vtkValidPoi...

Streamlines – Filter StreamTracer



Vectors velocity

Vector Interpolator Type Interpolator with P

Integration Direction BOTH

Integrator Type Runge-Kutta 4-5

Integration Step Unit Cell Length

Initial Step Length 0.2

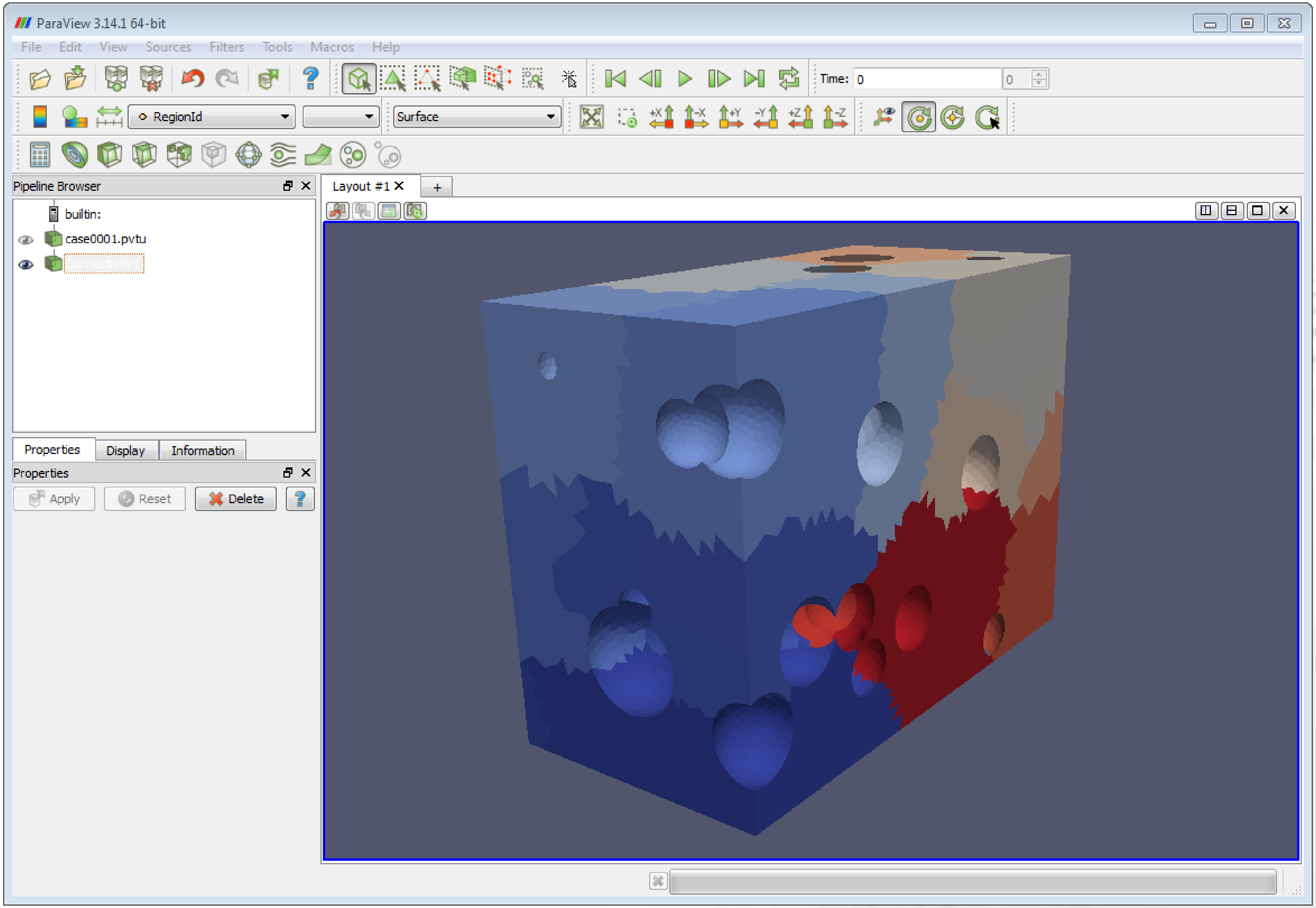
Minimum Step Length 0.01

Maximum Step Length 0.5

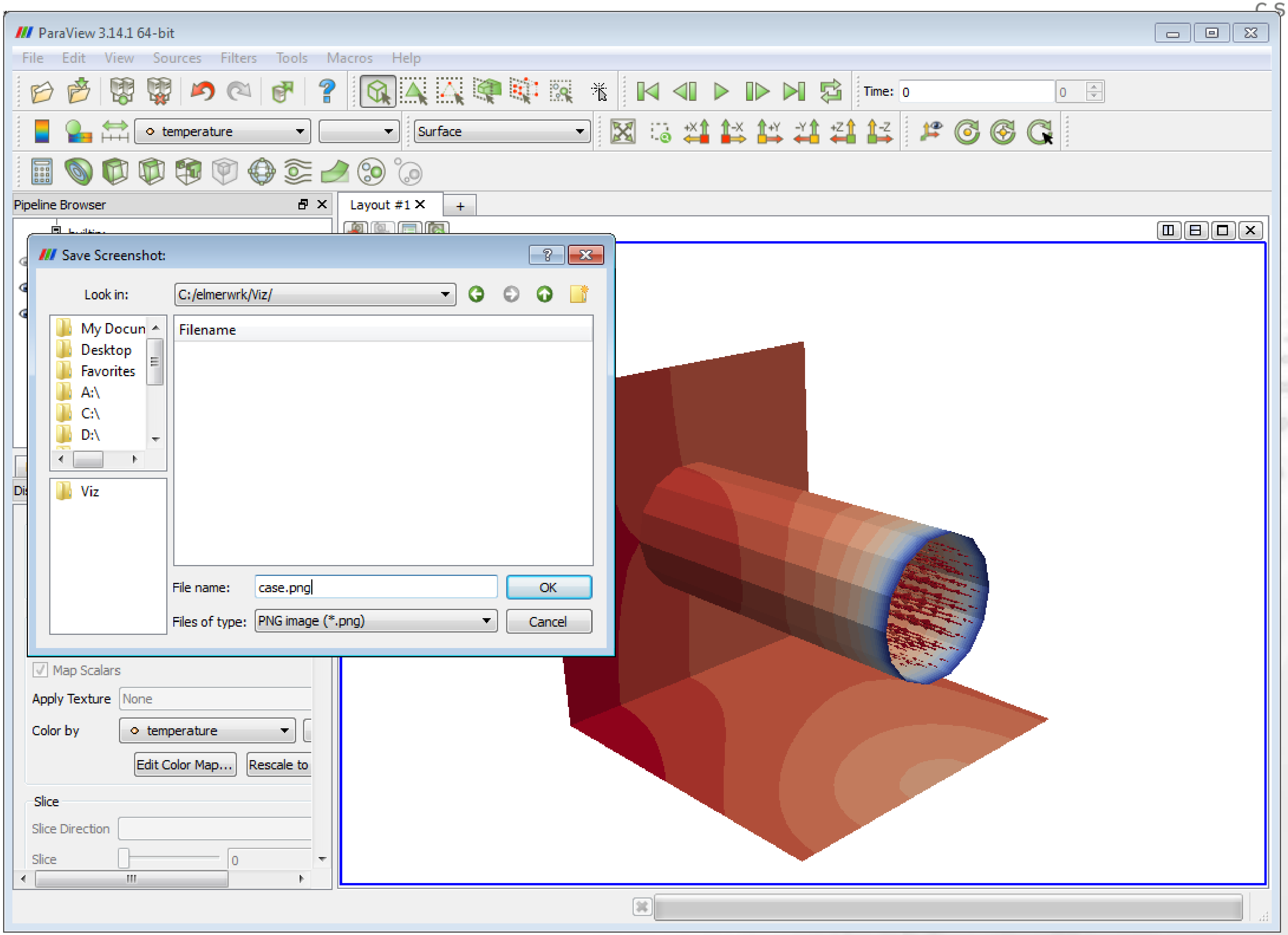
Maximum Steps 2000

Maximum Streamline Length 1

Partitioning – Connectivity filter



Saving figures



Time: 0

Color by: temperature

Representation: Surface

Look in: C:/elmerwrk/Viz/

File name: case.png

Files of type: PNG image (*.png)

Apply Texture: None

Slice: 0

Saving animations with Paraview

- The only packing method that comes with Paraview by default is motion AVI
- It is advicable to save the animation as separate files
- You may use ElmerClips to make mpg animations of the separate png figures