Visualization

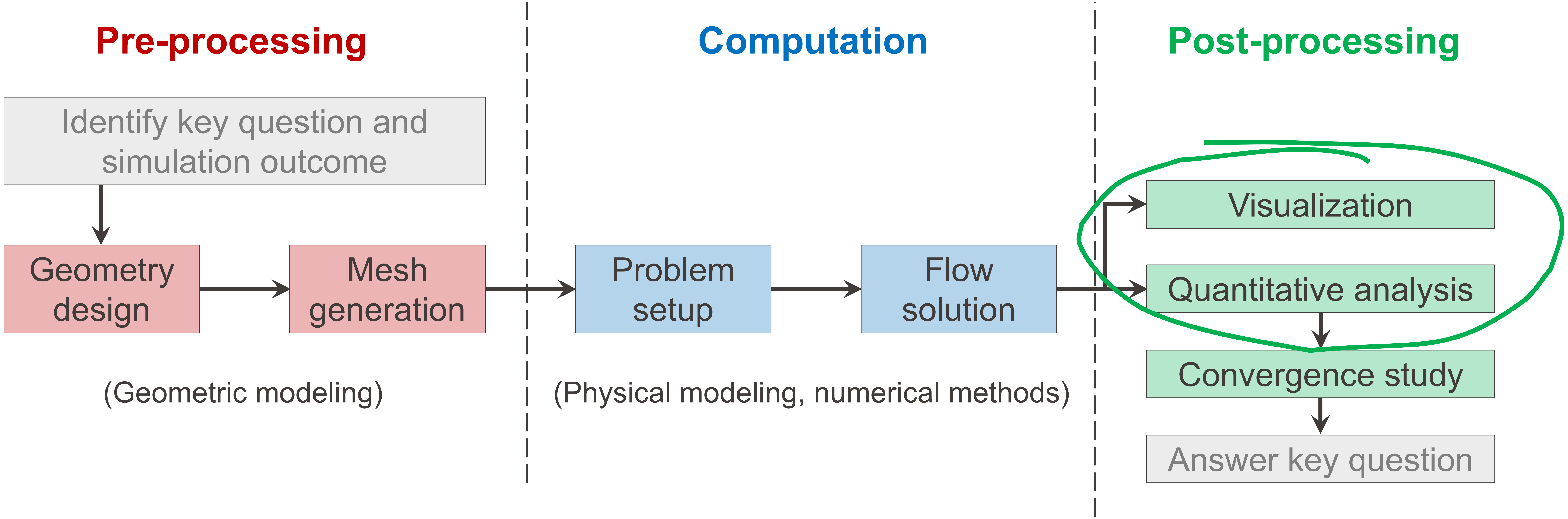
Numerical Flow Simulation

Edouard Boujo

Fall 2024

École
polytechnique
fédérale
de Lausanne

Numerical simulation workflow



Numerical Flow Simulation

Objectives

Why should we visualize scientific data (simulation & experiment results)?

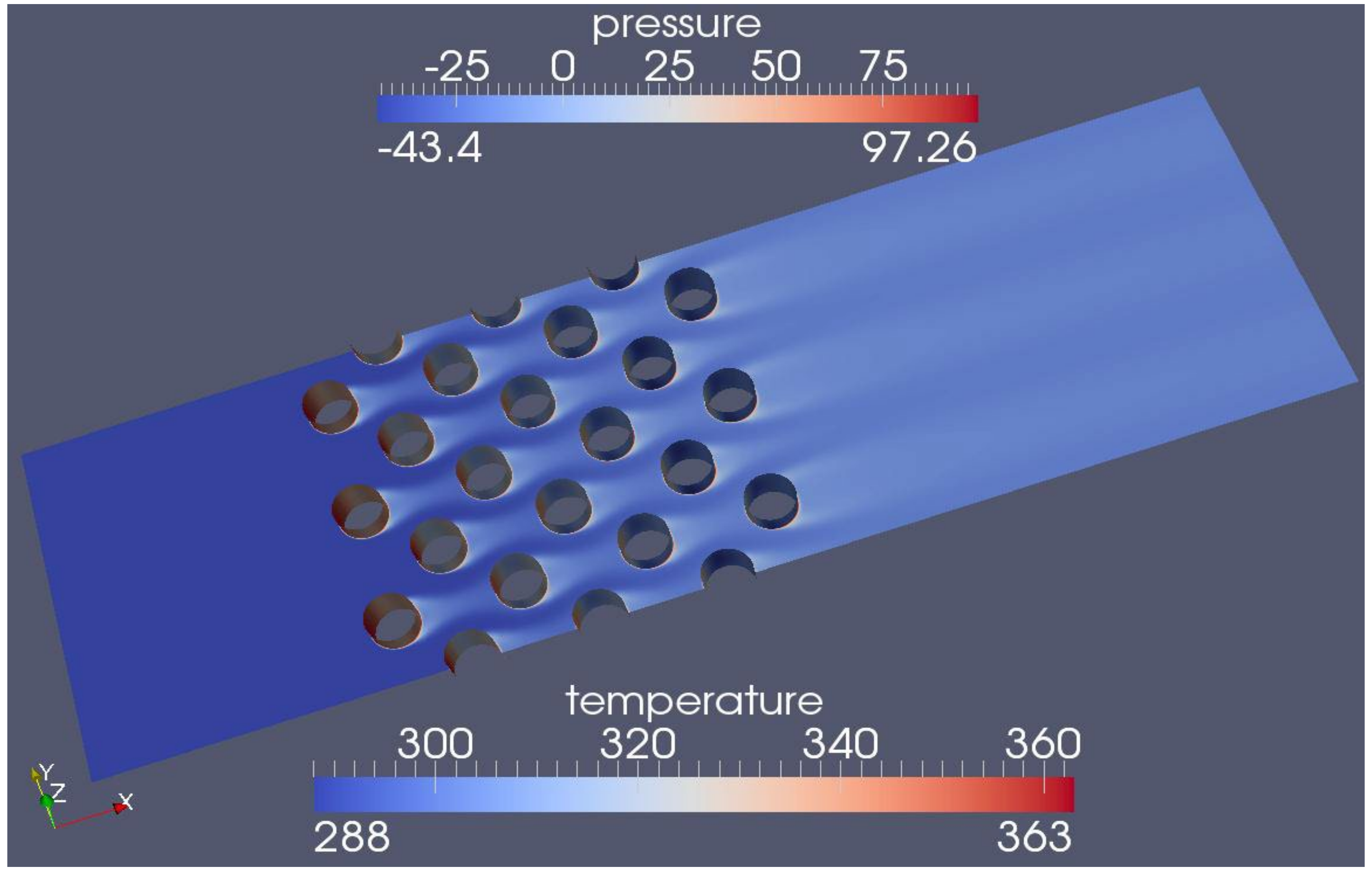
1. **Scientific visualization:**
   - help understand physical phenomena
   - enhance scientific discovery

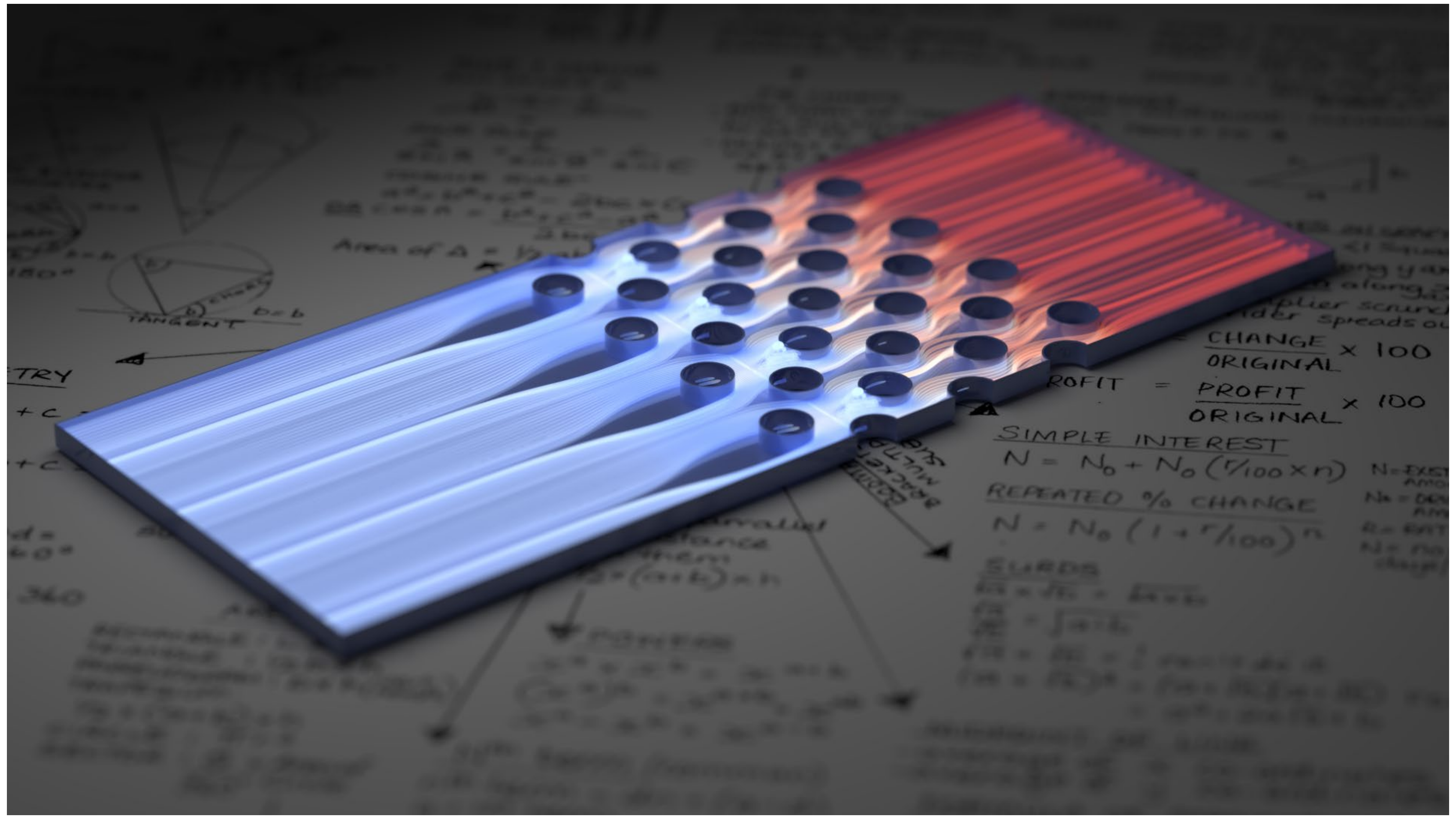
2. **Visual communication:**
   - communicate scientific results to peers (e.g. researchers, funding bodies)
   - pedagogy: improve student comprehension
   - outreach to general public, marketing

Objectives

1. **Scientific visualization**: often contains accurate and quantitative scientific information. “More scientific”.

2. **Visual communication**: often more “striking” (attracts attention) and “memorable” (easily remembered). “More aesthetic”.





Numerical Flow Simulation

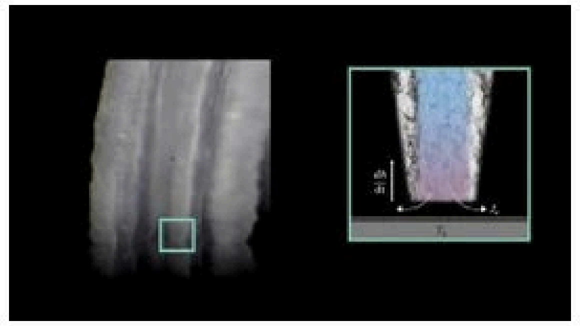
The two can be combined.

Objectives

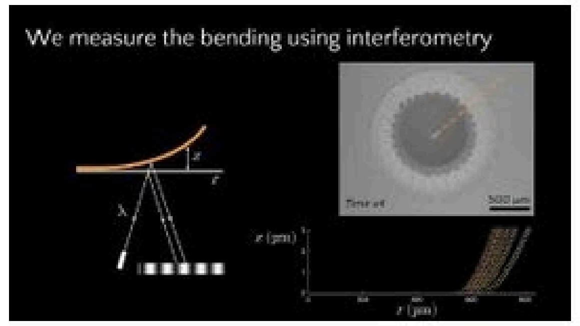
- Some examples of scientific visualization and/or visual communication:
  - APS DFD Gallery of Fluid Motion

2019 APS/DFD Milton van Dyke Award Winners (Video)

V0013: Impaled droplets: On the breakup of drops impacting singularities



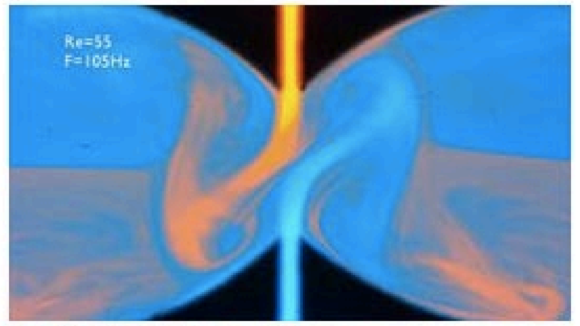
V0040: Crystal Critters



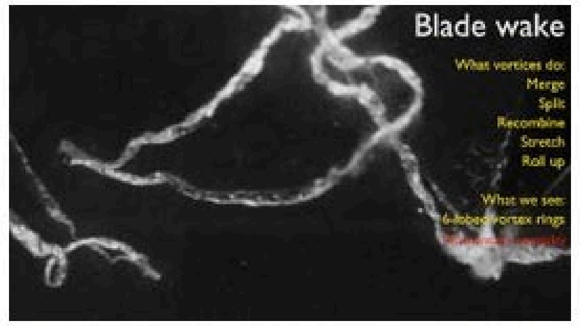
V0082: Blooming patterns in drying drops

2019 APS/DFD Gallery of Fluid Motion Award Winners (Video)

V0072: Gas Escape Behavior from Bursting Bubbles

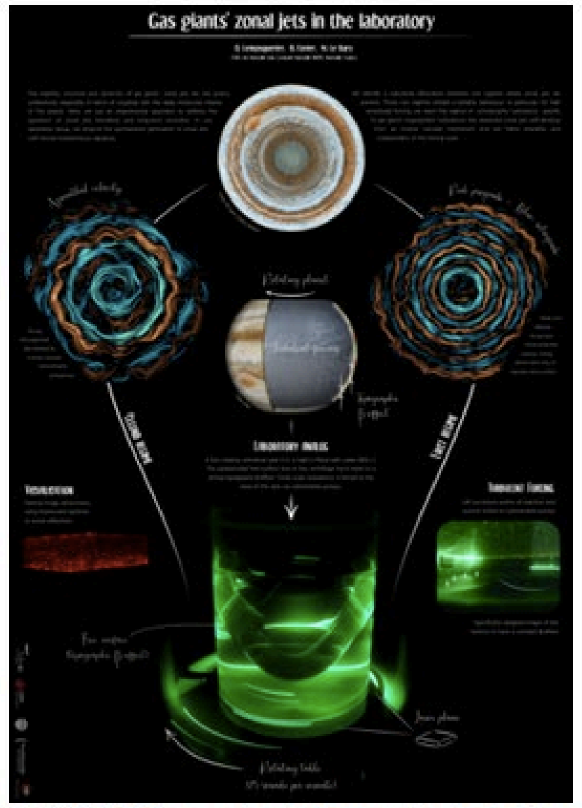


V0036: Swinging jets

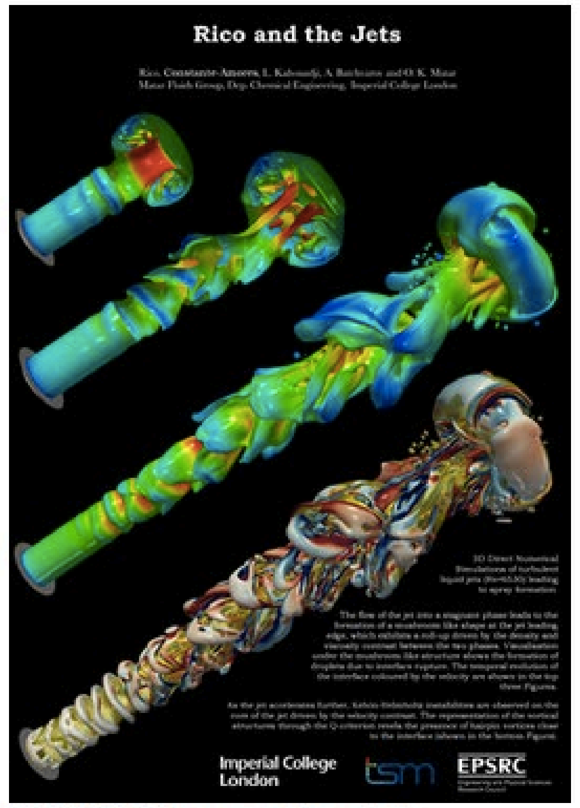


V0067: Unraveling the interplay of two counter-rotating helical vortices

2019 APS/DFD Milton van Dyke Award Winners (Poster)



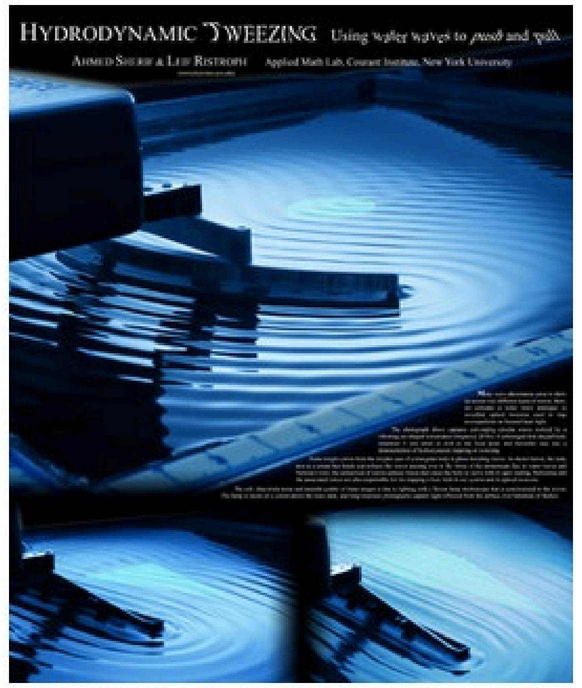
P0015: Gas giants' zonal jets in the laboratory



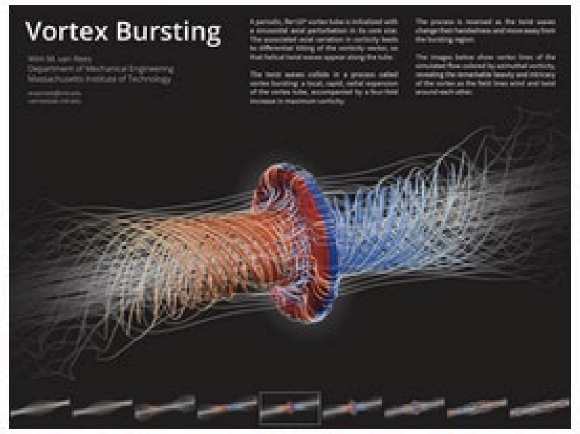
P0020: Rico and the Jets

P0034: Fluid Dynamics of Millefiori

2019 APS/DFD Gallery of Fluid Motion Award Winners (Poster)



P0031: Hydrodynamic Tweezing: Using water waves to push and pull



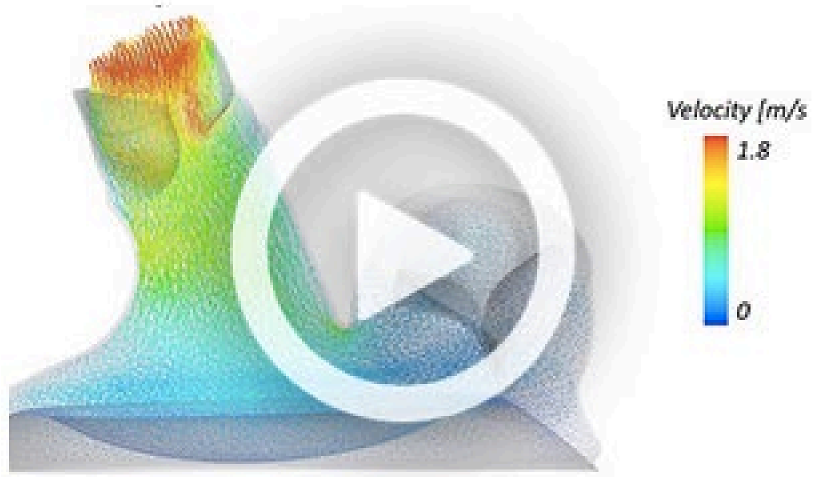
P0035: Vortex Bursting

https://gfm.aps.org/meetings

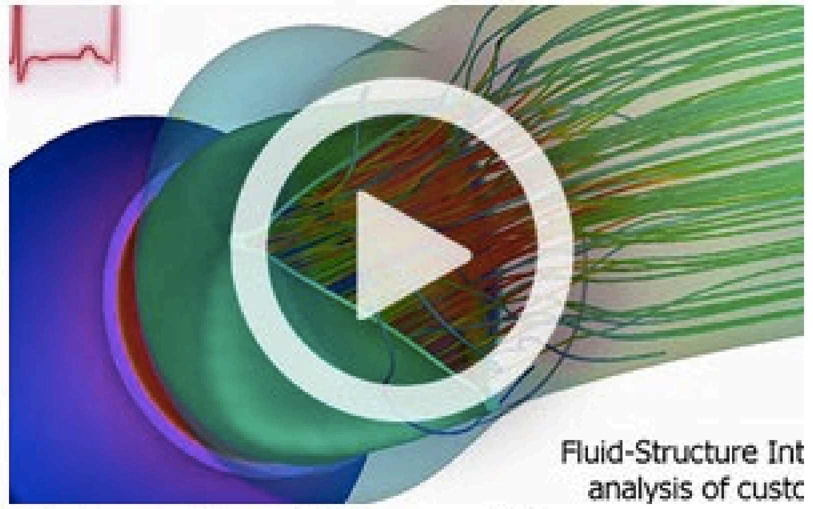
Objectives

- Some examples of scientific visualization and/or visual communication:
  - Ansys Hall of Fame competition

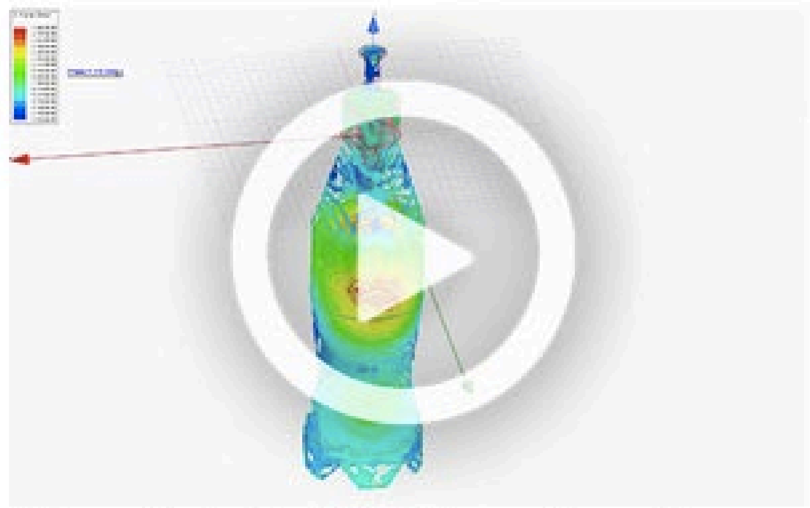
Top 3 Academic winners "Best in Class"



Politecnico di Milano, Italy



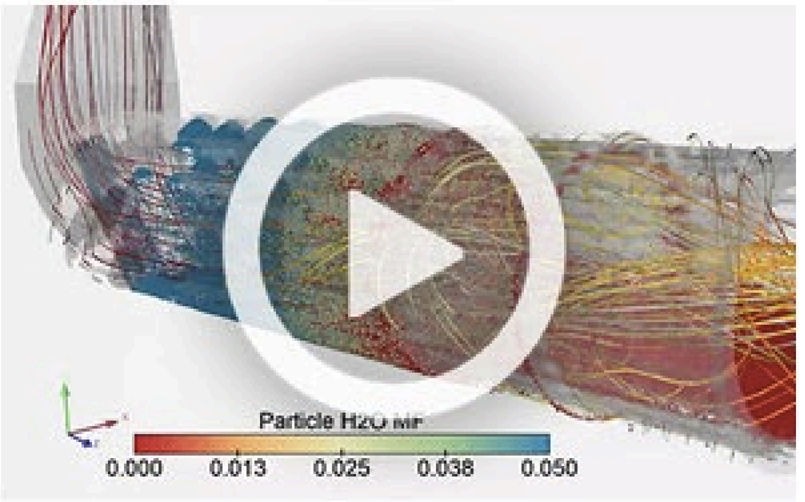
University of Rome "Tor Vergata," Italy



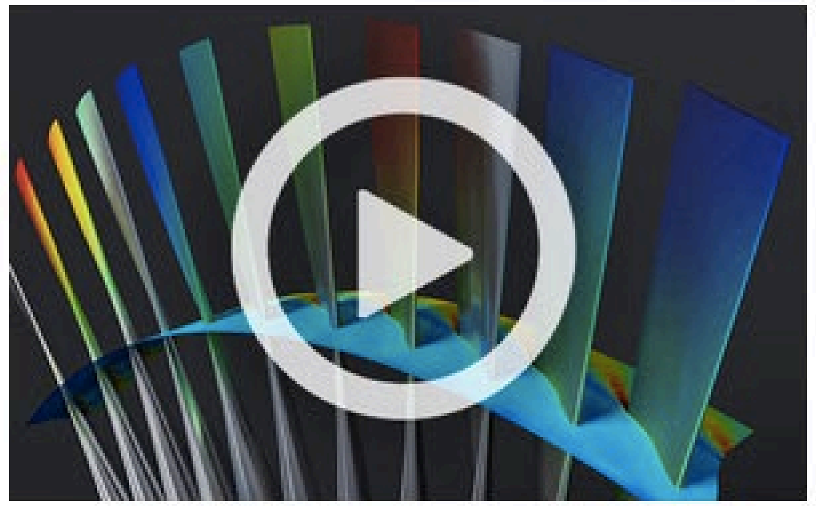
King Abdullah University of Science and Technology, Saudi Arabia

Top 3 Commercial winners "Best in Class"

Druids Process Technology S.L., Spain



Astec, Inc., U.S.A.

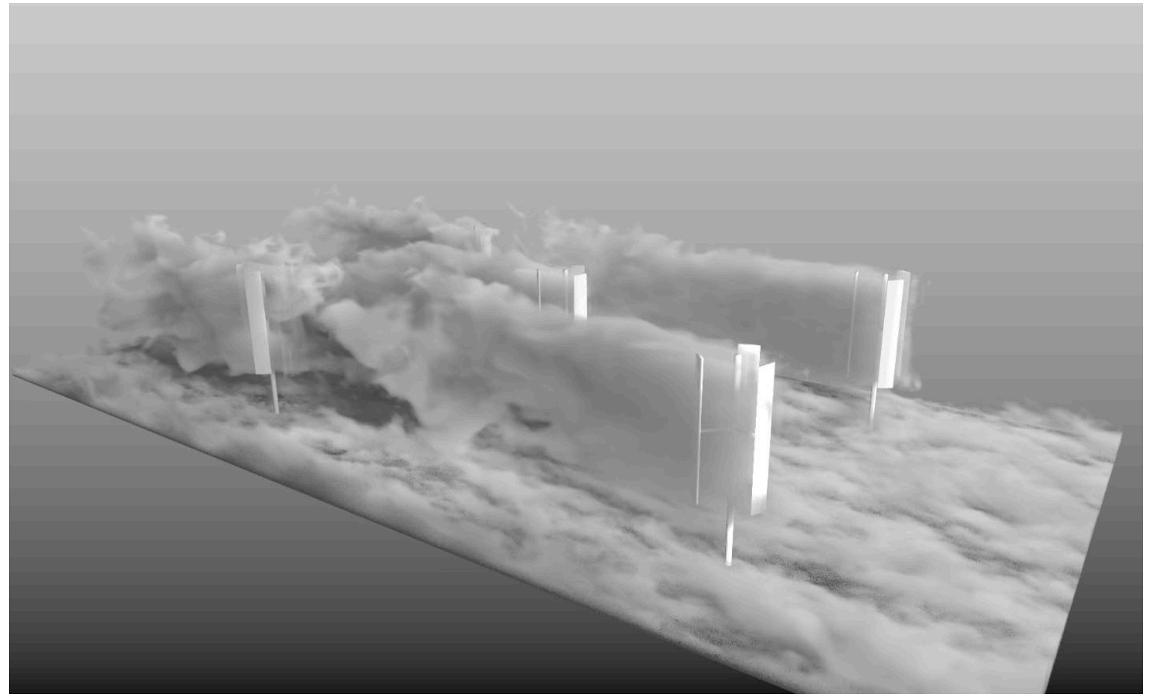
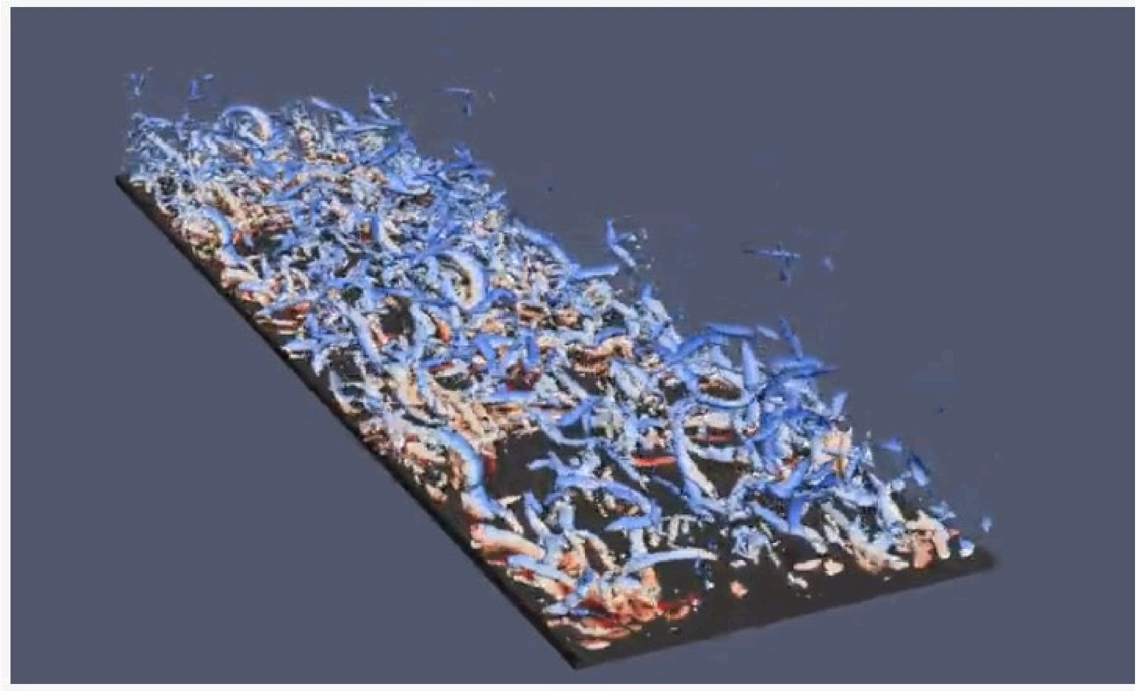


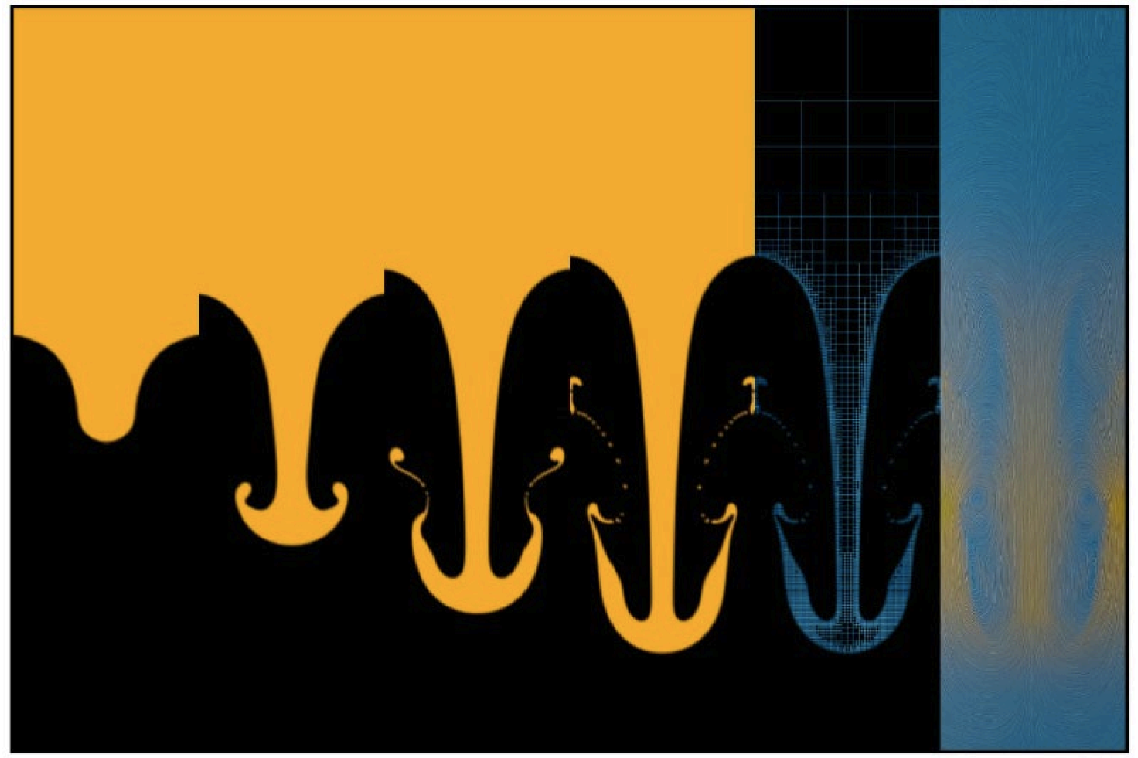
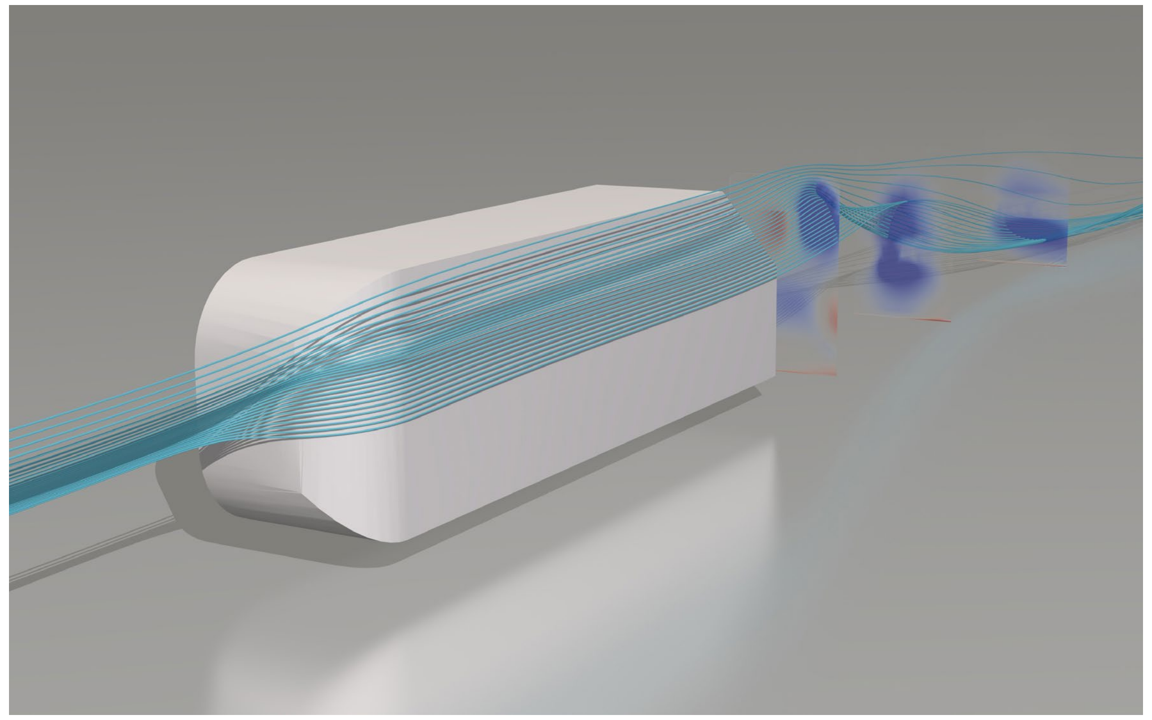
NUM solution, Czech Republic

https://www.ansys.com/other/hall-of-fame

Objectives

- Some examples of scientific visualization and/or visual communication:
  - EPFL ACCES Visualization Contest

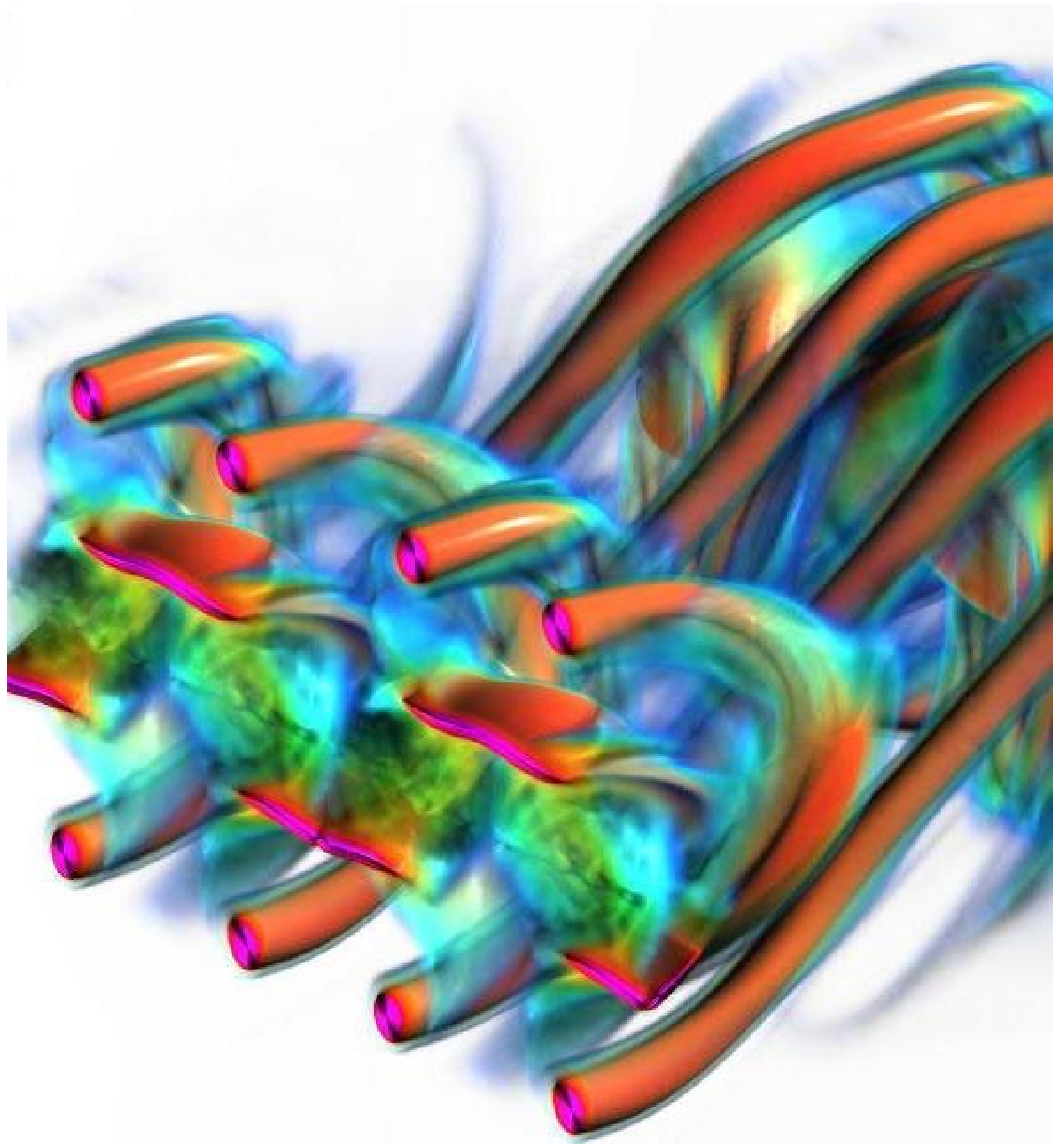



https://www.epfl.ch/labs/acces/

Techniques

- Visualization of scientific data is increasingly important as a result of:
  - Increased computation & storage: larger and more complex physical problems
  - Increased graphics: increased possibilities provided by modern graphics cards

- Advanced visualization relies on:
  - Computer graphics
  - Image processing
  - High-performance (parallel) computing

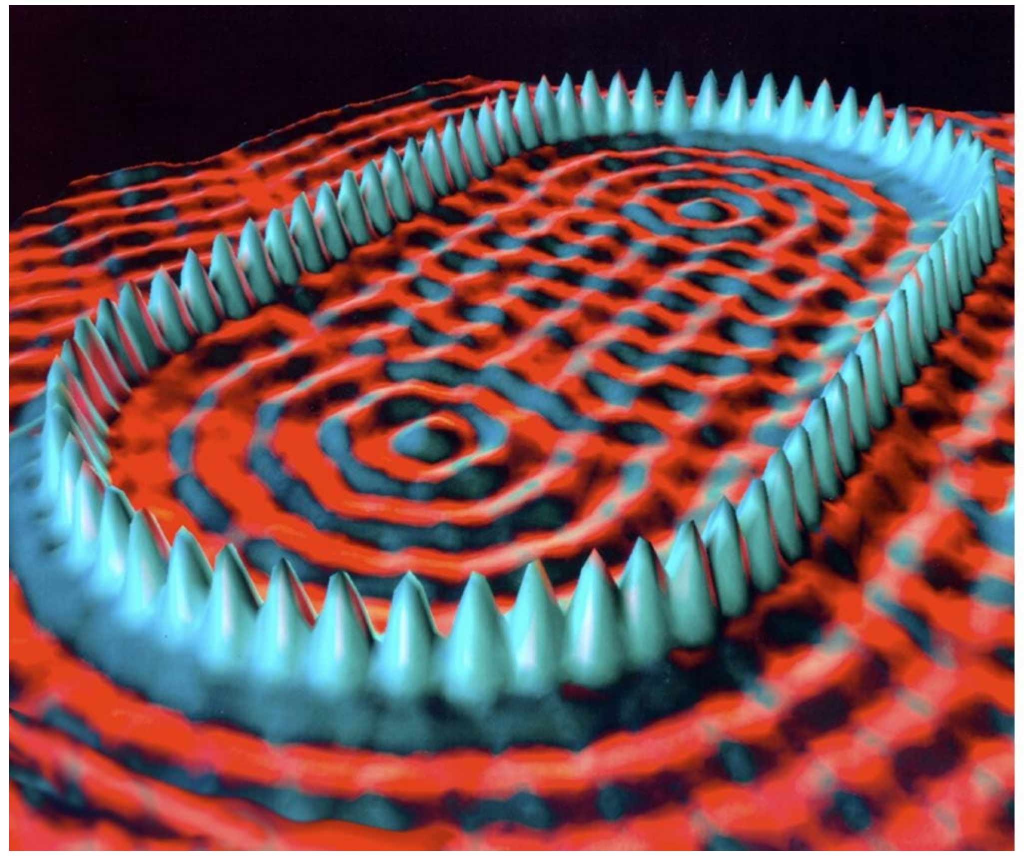


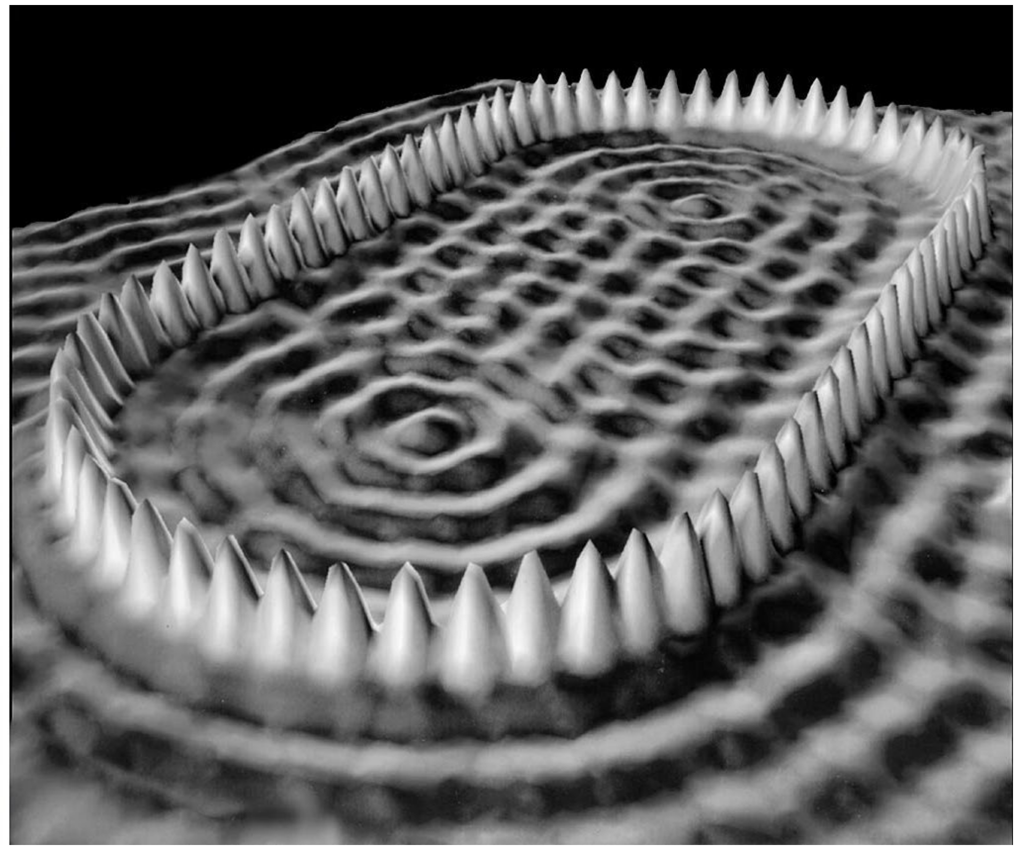
Techniques

- Visualization of scientific data is a combination of:
  - Plotting techniques for 2D data
  - Iso-surfaces and volume rendering for 3D data

  - Photorealism (ray tracing & texturing)
  - Stereoscopy (depth information)
  - Animation
  - Data mining & pattern discovery (e.g. coherent structures)
  - Color and human perception
  - Aesthetics (artistic aspects)
  - …

  - Software implementation
  - Hardware implementation





Sometimes, less techniques may be more effective (e.g. color can be distractive)

Techniques: traditional rendering ("rasterization")

- Scientific visualization software (e.g. Fluent, ParaView) generally use traditional rasterization to map 2D/3D data onto 2D screen.

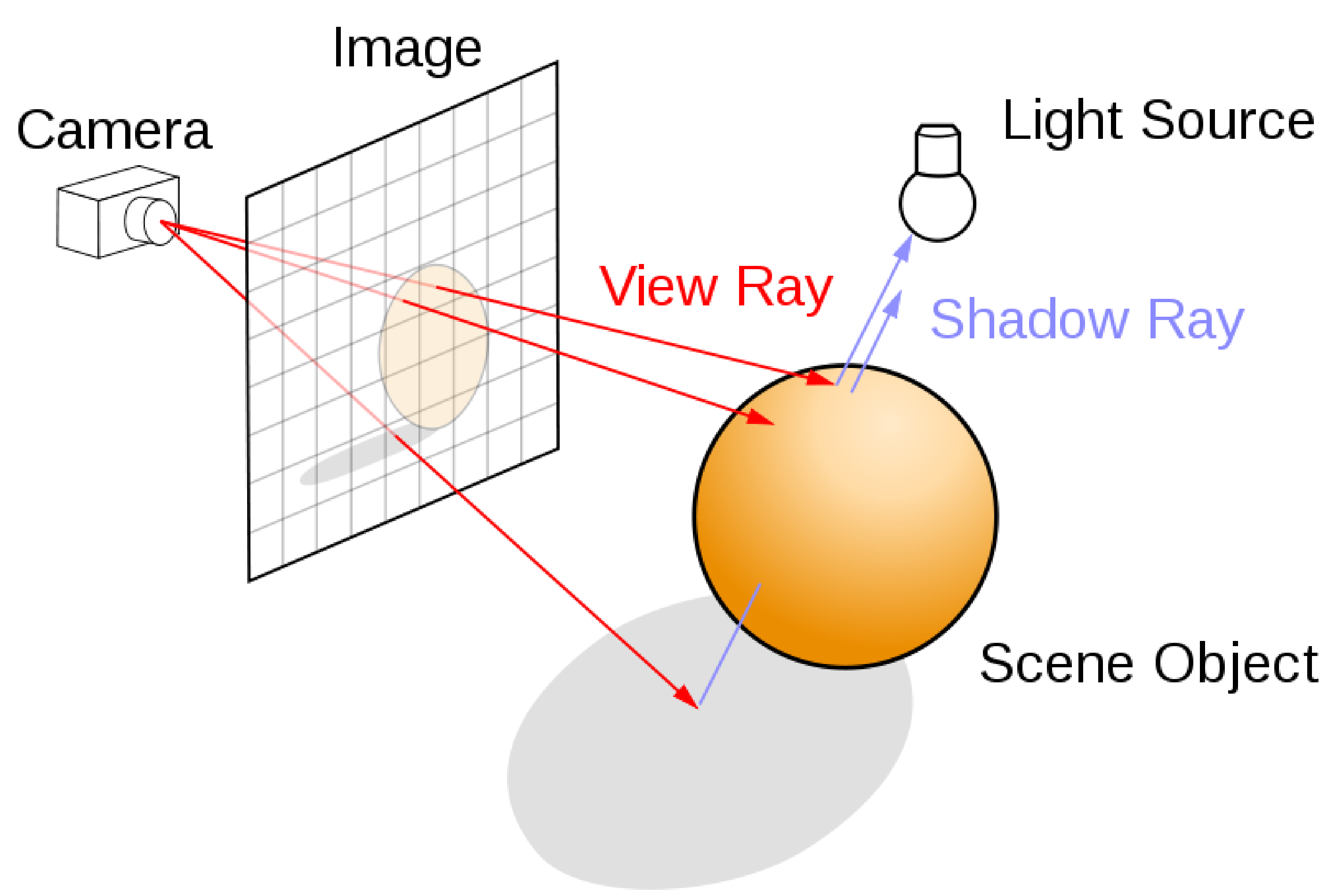
- Simplified steps:
  - Create objects as a mesh of geometric primitive objects (e.g. triangles)
  - Associate properties with mesh (e.g. position, normal, color, texture)
  - Convert 2D/3D triangles into pixels in 2D viewing plane
  - Add further pixel processing (e.g. shading) for greater realism

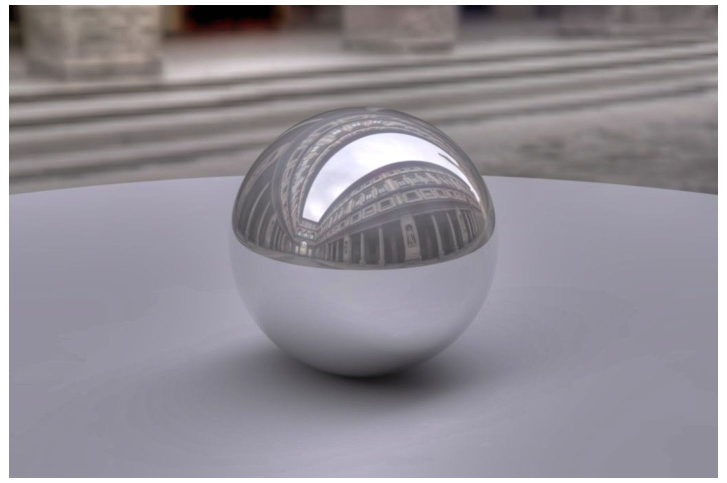
- Characteristics:
  - Extremely fast when using specialized hardware (GPU)
  - Allows real-time visualization (e.g. video games)
  - Limited in producing realistic details (e.g. physical light behavior)

Techniques: ray tracing

- For visual communication, can trace the path of light backwards through the pixels in the image plane, and simulate the interaction of light with objects.

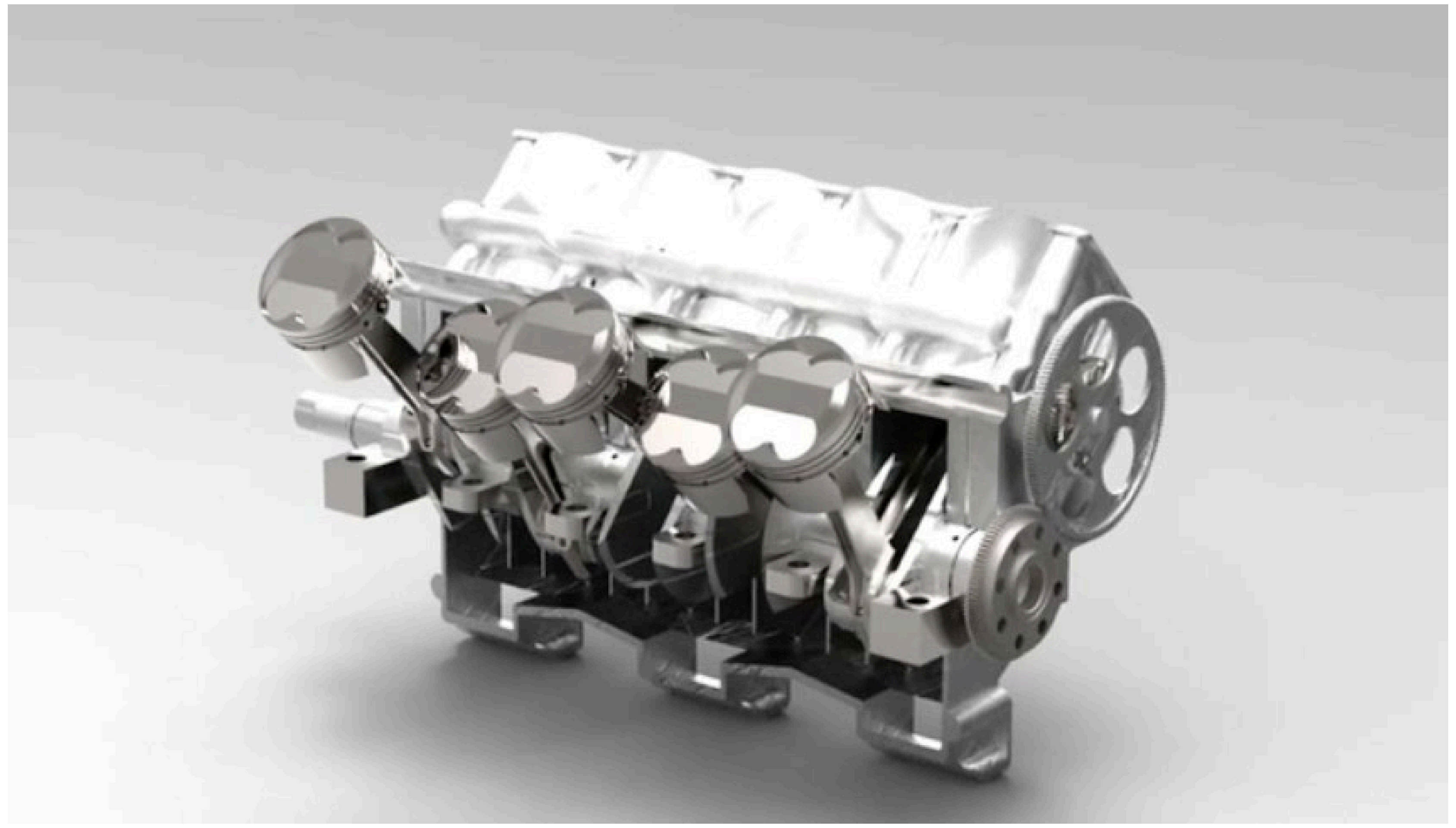
- Characteristics:
  - High degree of virtual realism
  - Can simulate wide variety of optical effects (reflection, refraction, scattering…)
  - Computationally expensive. Generally not fast enough for real-time visualization (ok for movies, not video games)

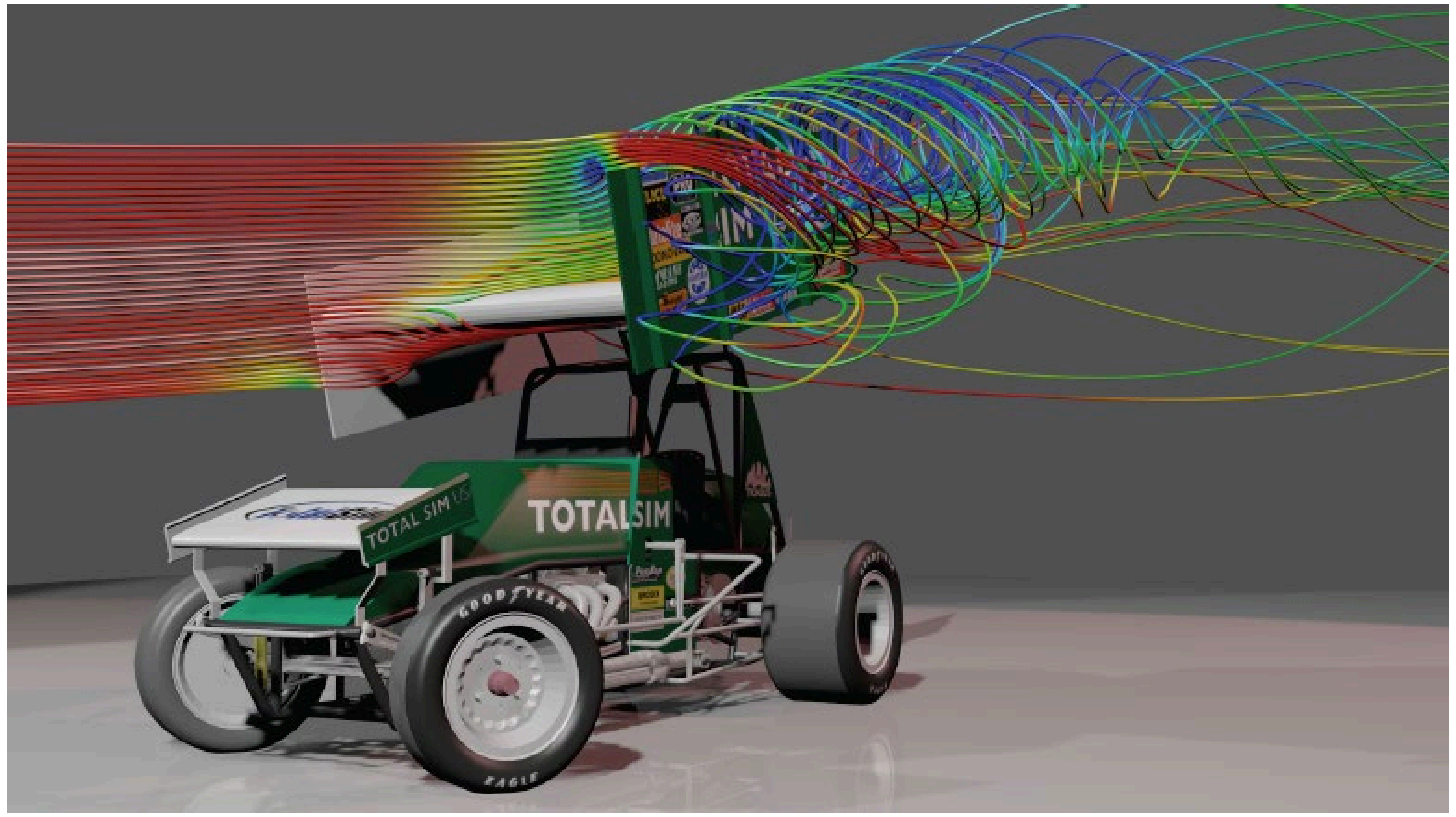




Techniques: photorealism

- Ray tracing + surface texturing → realistic (photographic) view
- Incorporated in some CAD / CFD software
- Specialized software:
  - Open source: Blender, POV-ray…
  - Commercial: Maya, 3ds Max…



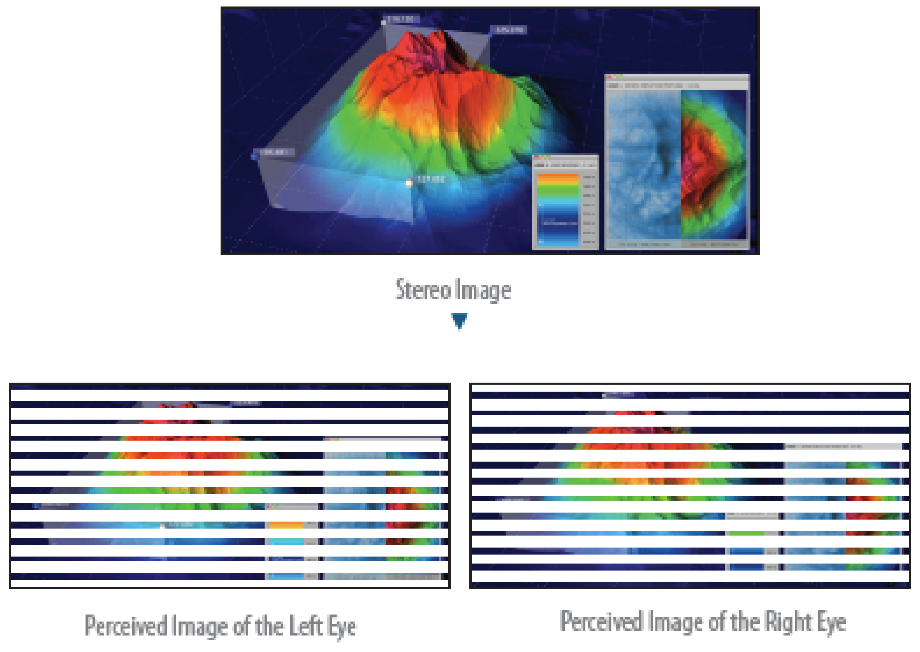


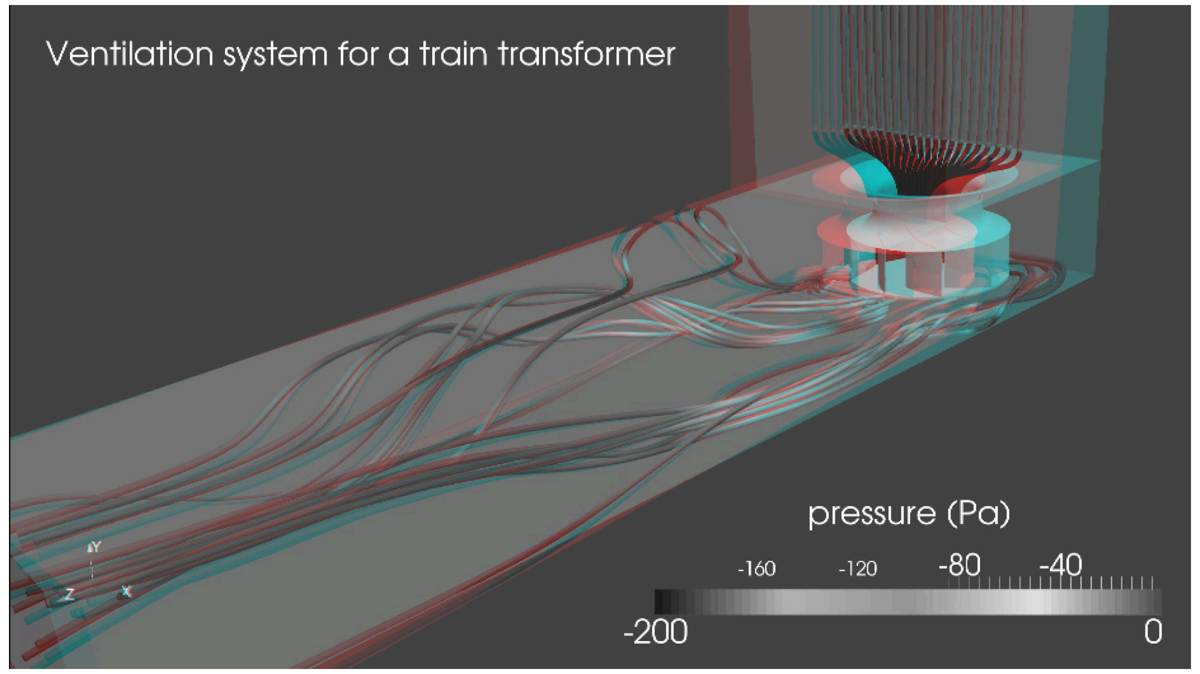
Numerical Flow Simulation

Techniques: stereoscopy

- All images of the 3D world are 2D, but the human brain can (partially) reconstruct a 3D representation if different images in left & right eyes.
- Three main categories of stereoscopic images:
  1. Active: switch L/R images in time (good quality but need expensive synchronized liquid crystal shutter glasses)

  2. Passive: polarized system (inexpensive glasses, good quality) or color anaglyph (inexpensive but lower quality)



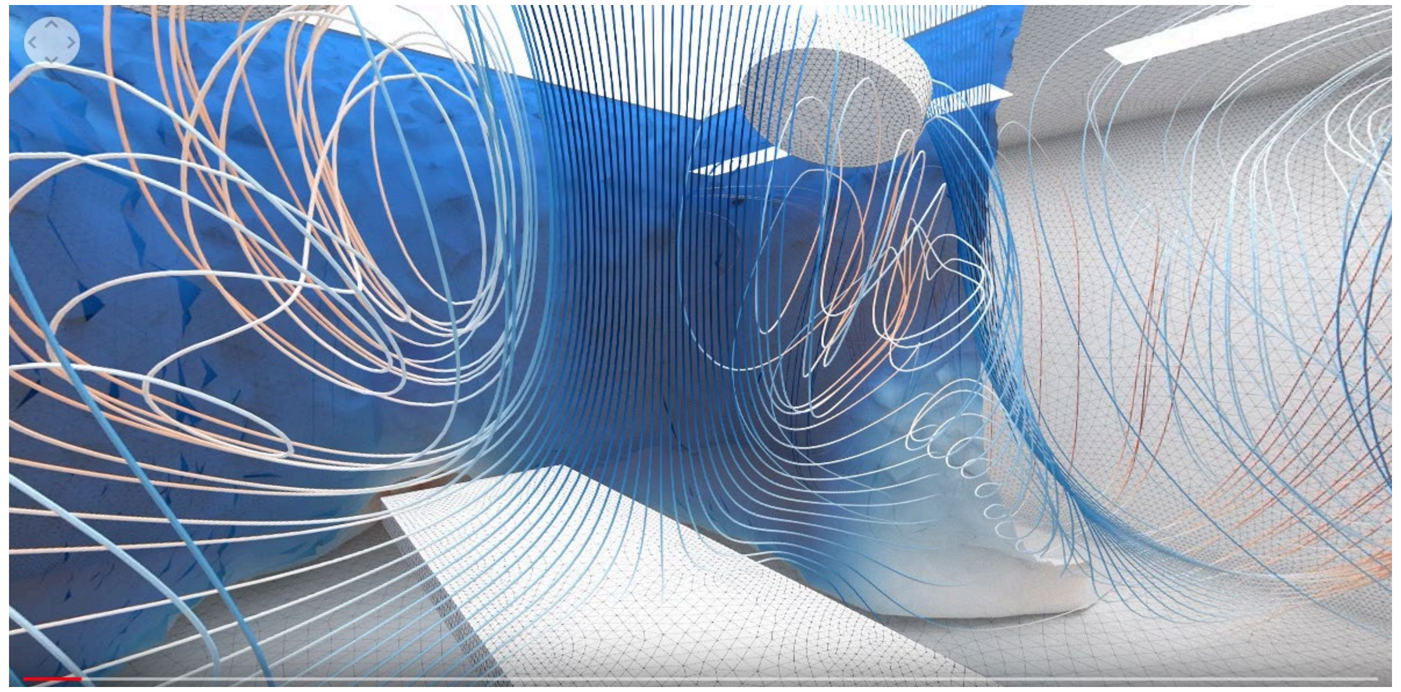


  3. Autostereoscopy ("glasses-free"): L/R images split by optical components

Techniques: virtual reality

- Stereoscopy + 360°. Immerse viewer within data field.
- Two main approaches:
  1. Projection on multiple walls: more than one viewer, expensive
  2. Headset: single viewer
     - Entry-level (inexpensive): passive headset + smartphone with gyroscope + VR app/website

     - High-level (expensive): active headset + computer with GPU + library of applications



Example: CFD ventilation study of an operating room
https://youtu.be/By3L6SeSnJ0

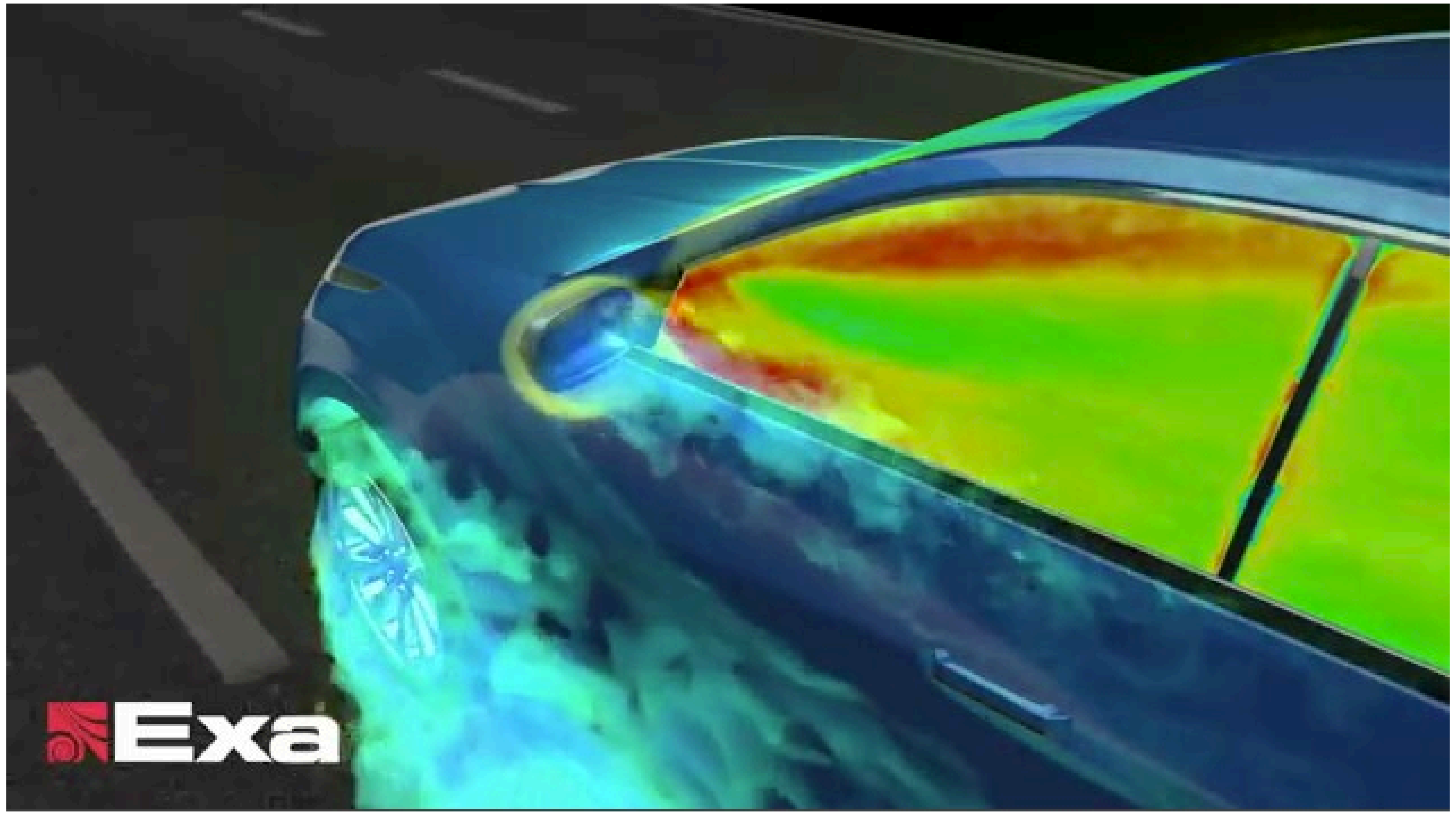
Techniques: animations

- Animations
  - Several types:
    - Move camera (stationary data)
    - Vary time (time-dependent data)
    - Can combine the above two
  - Choices to make:
    - Image size (number of pixels)
    - Frame rate
    - File type: either video (AVI, QuickTime, MPEG…), or series of images to be animated with another software (e.g. QuickTime Pro, Adobe Premiere, ImageMagick…)

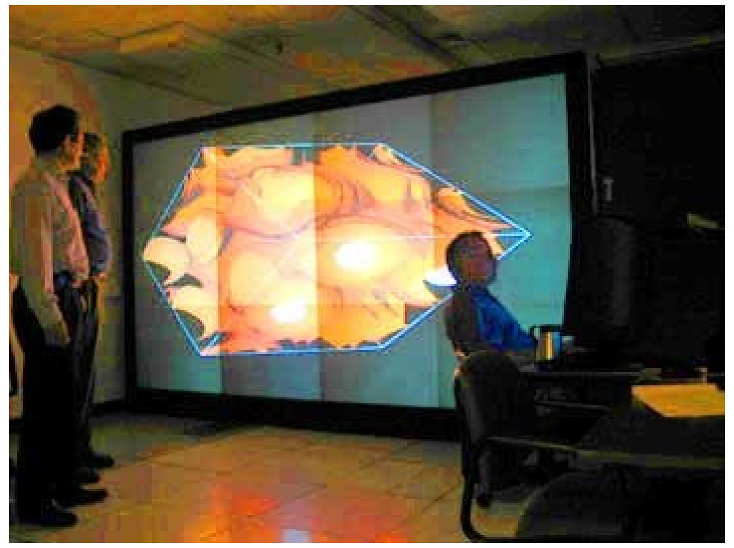
Techniques: auralization

- "Acoustic virtual reality"
  - Use of sound in addition to visual information
  - Compute sound generated at source and transmitted to receiver

Example: aero-acoustic noise from side mirror (video on Moodle)

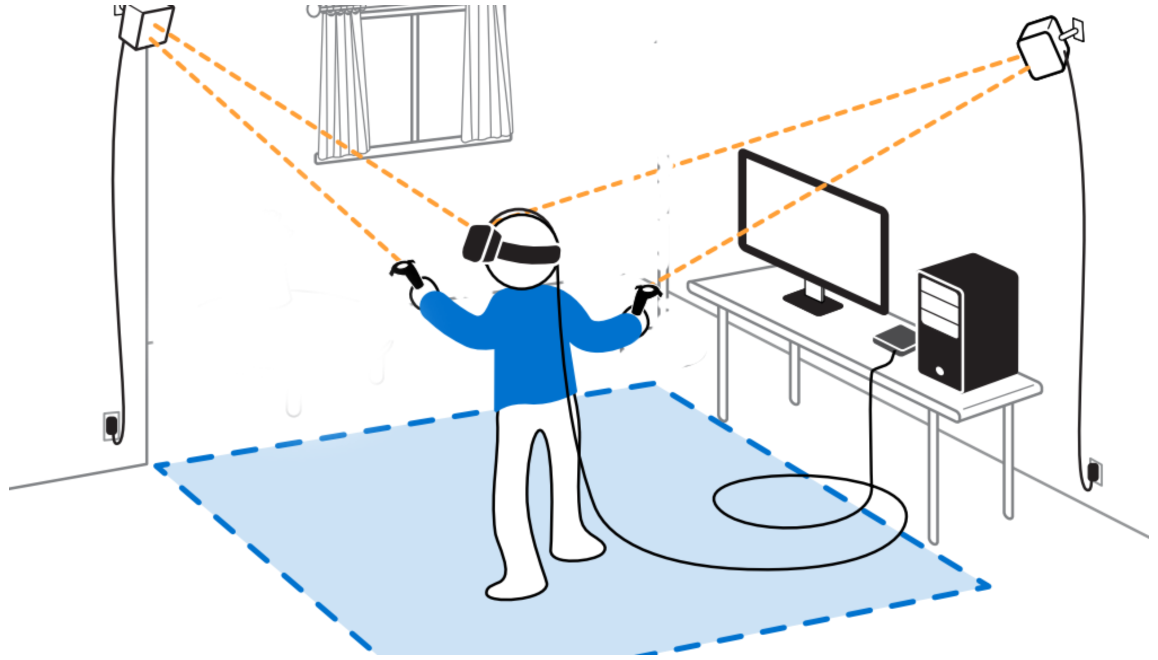


Specialized hardware



- High-end scientific visualization:
  - Parallel processing: data analysis, rendering
  - Display: large tiled display walls, stereoscopic (e.g. 3D monitors), interactive touch screen, immersive (e.g. headset)

- At EPFL: ACCES (joint STI & ENAC initiative) has several collaborative visualization facilities that extend visualization capabilities:
  - CoViz1: high resolution display, stereoscopy
  - CoViz3: virtual reality, immersion & interaction



https://www.epfl.ch/labs/acces/

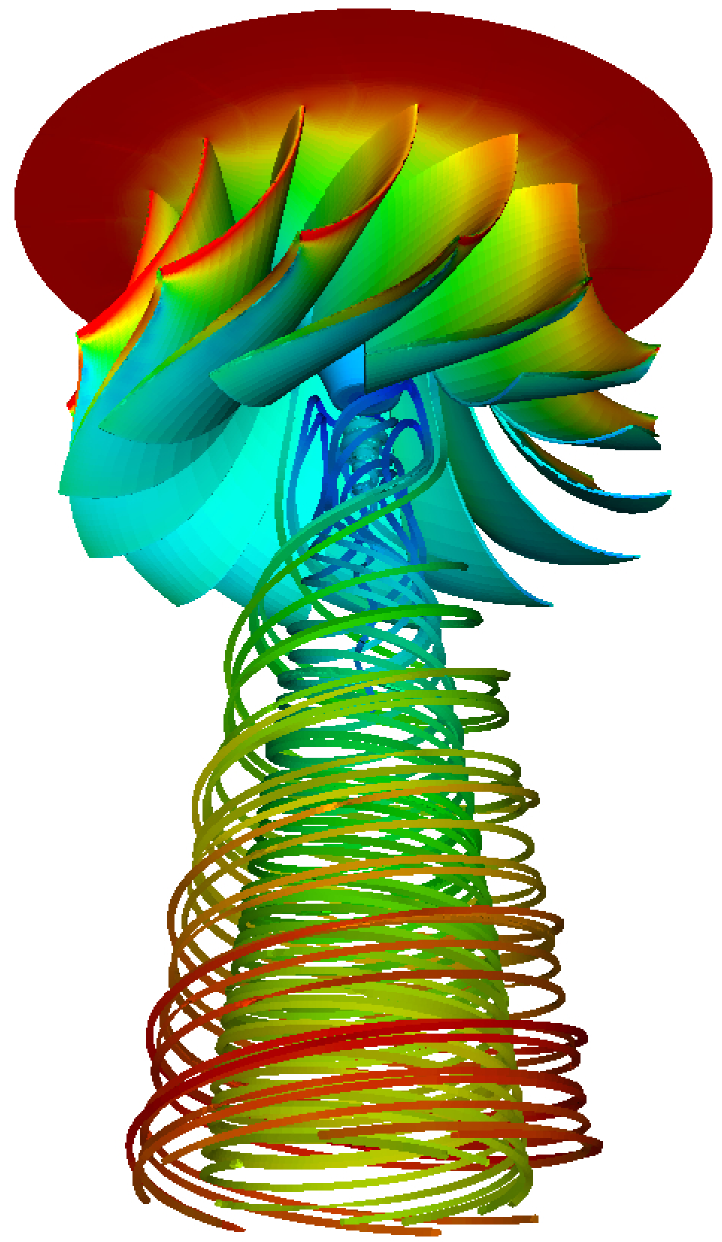
Specialized software

- Why use **integrated visualization tools** (e.g. Fluent)?
  - Can quicky review simulation results.
  - Can stop simulation, visualize, and modify / continue calculation.

- Why use **specialized visualization software** (e.g. Ensight, Paraview...)?
  - Advantages:
    - Flexible and extensible
    - More powerful (e.g. memory requirements, speed, functionality)
    - Enable importing of different datasets (e.g. coupled simulations)
    - If open source, general availability on different computer systems

  - Disadvantages:
    - Not integrated in the simulation environment
    - Require appropriate import/export capabilities

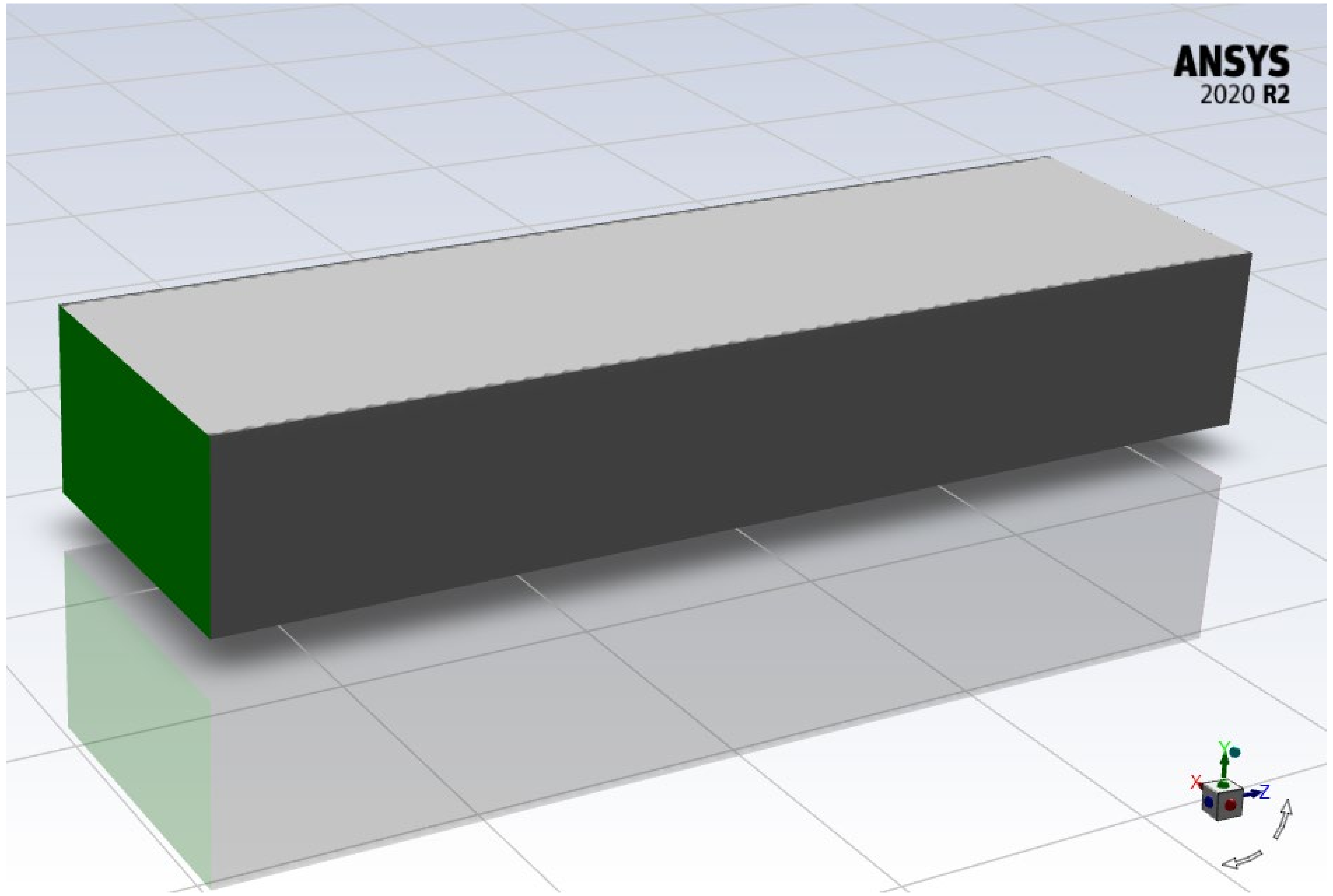
Specialized software

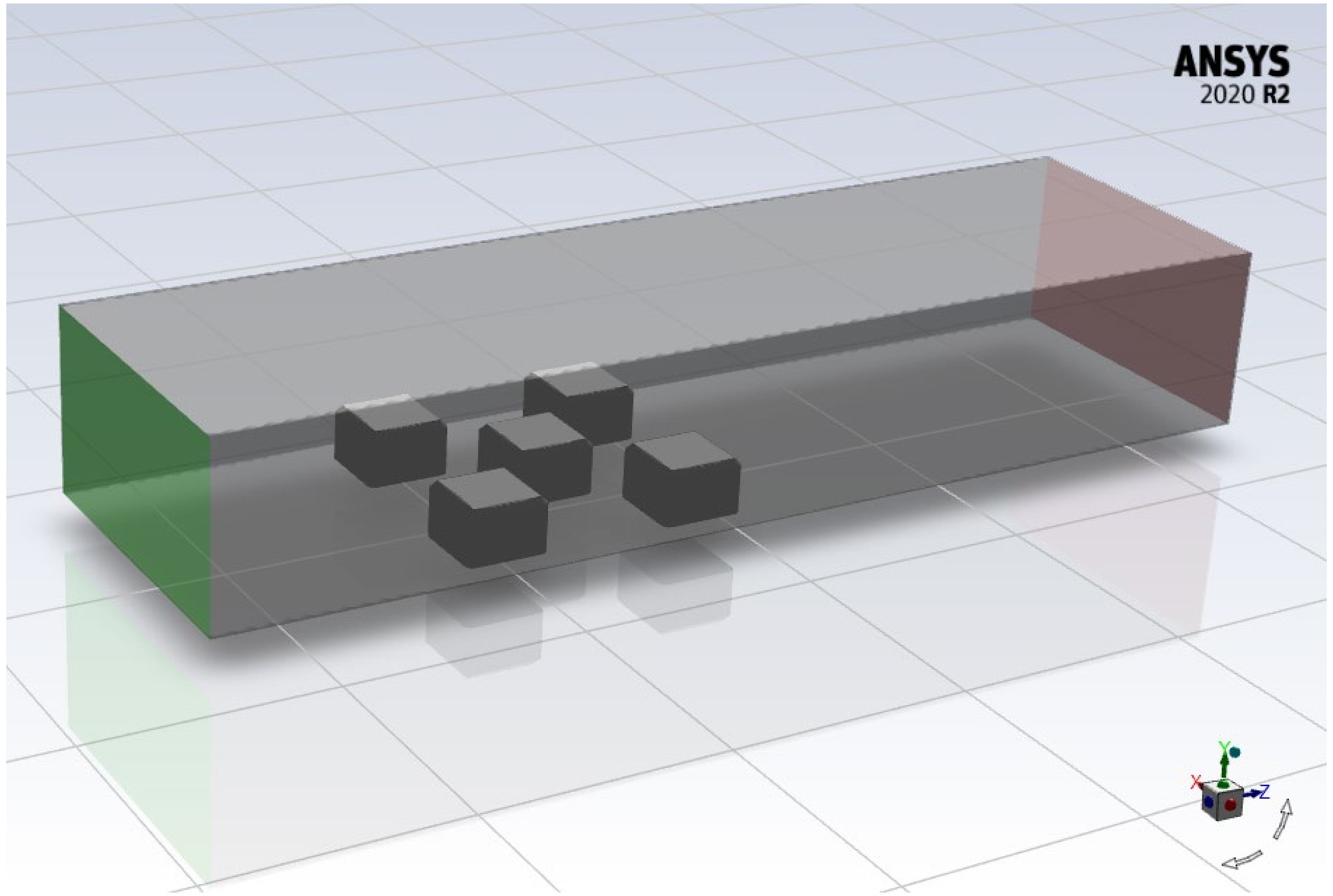


- Examples of software for CFD scientific visualization:
  - Fully integrated (black box with GUI, minimal extension possible by user)
    - FieldView (Intelligent Light - www.ilight.com)
    - EnSight (Ansys - www.ensight.com)
    - Tecplot (Tecplot Inc. - www.tecplot.com)
  - Library-based (graphics library with GUI, extensible by user)
    - ParaView (open source - www.paraview.org)
    - VisIt (open source - wci.llnl.gov/codes/visit)
  - Modular Visualization Environment (programming environment, extensible by user)
    - AVS/Express (AVS Inc. - www.avs.com)

Basic visualization in Fluent

- Geometry and mesh

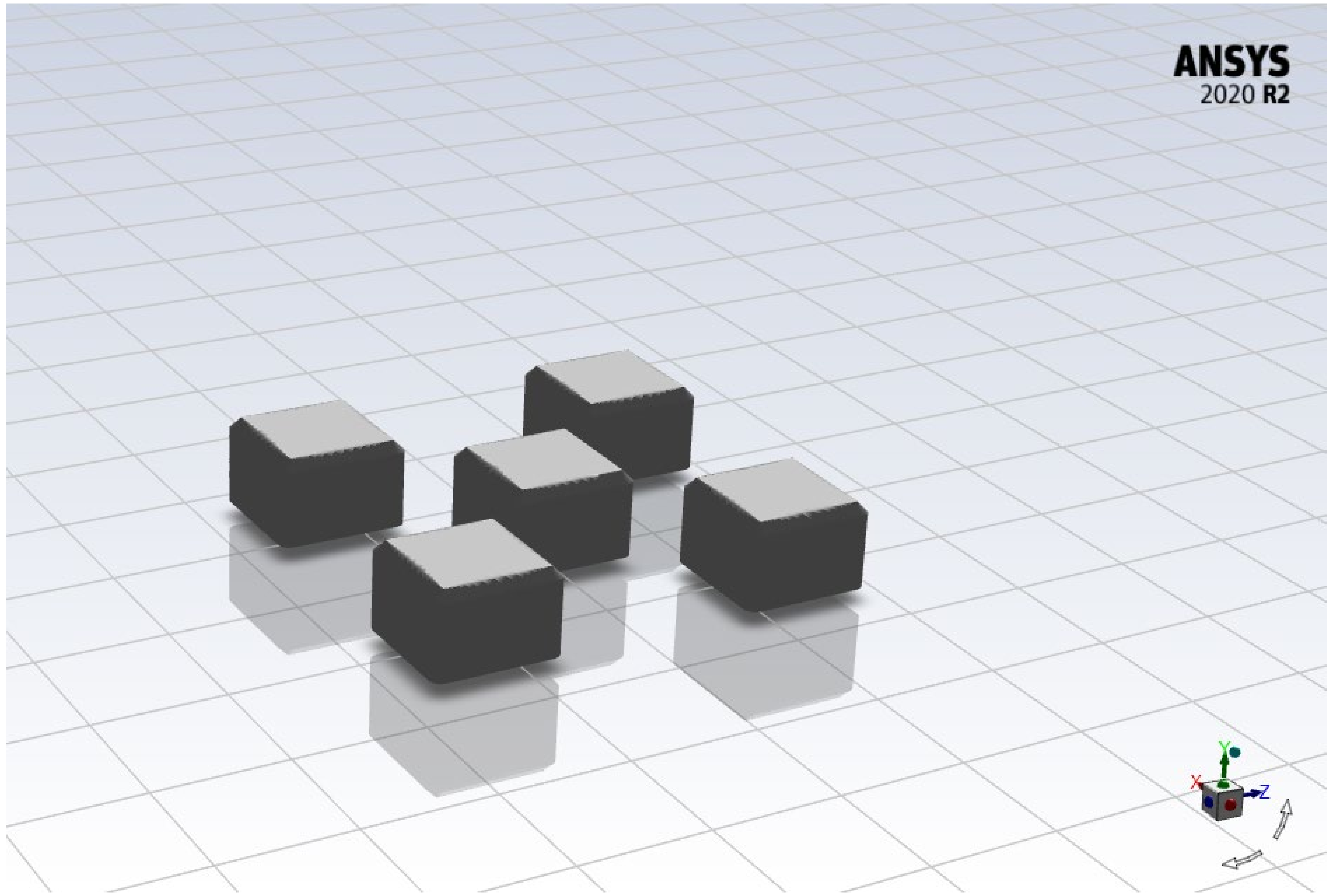




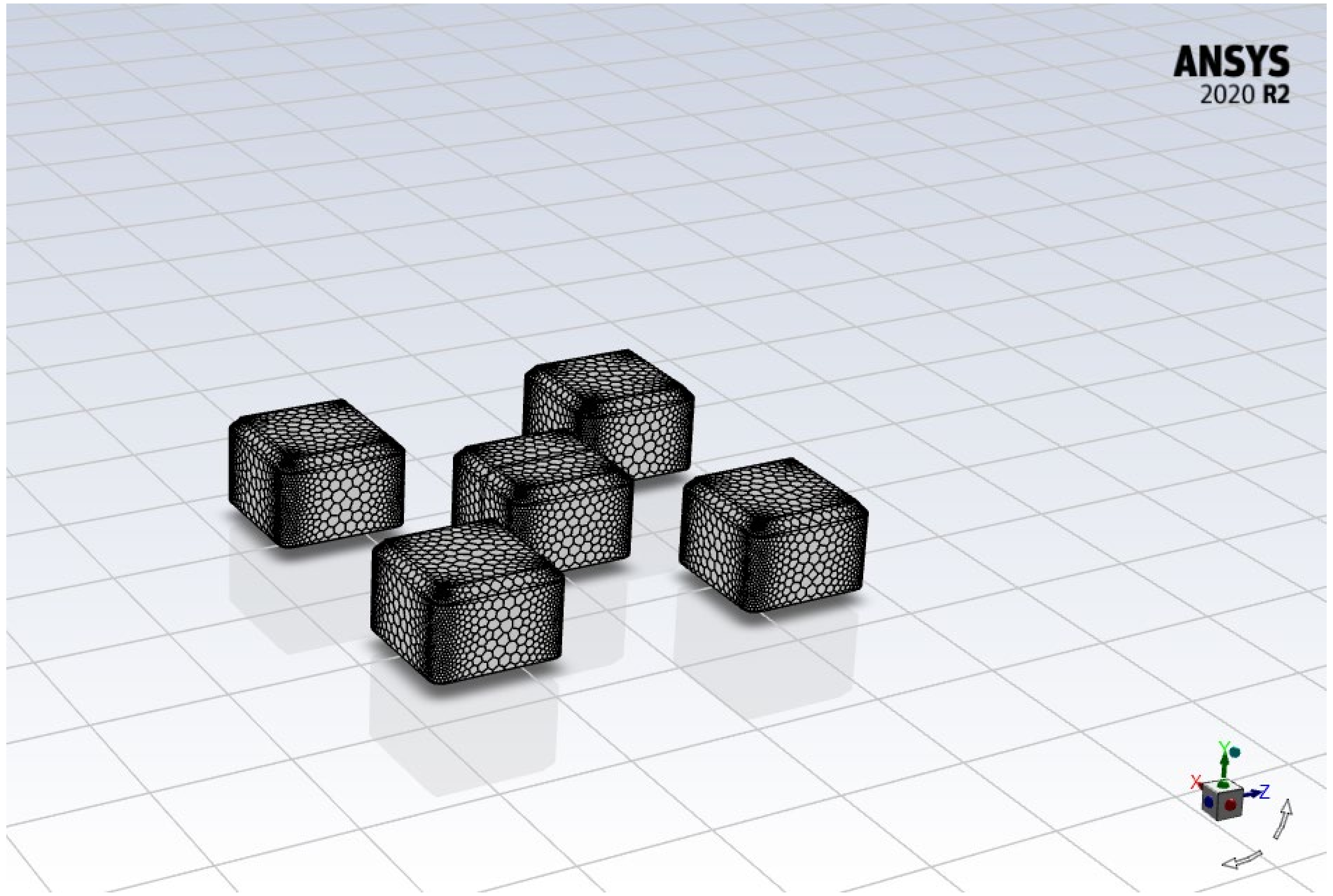
Transparency

Basic visualization in Fluent

- Geometry and mesh



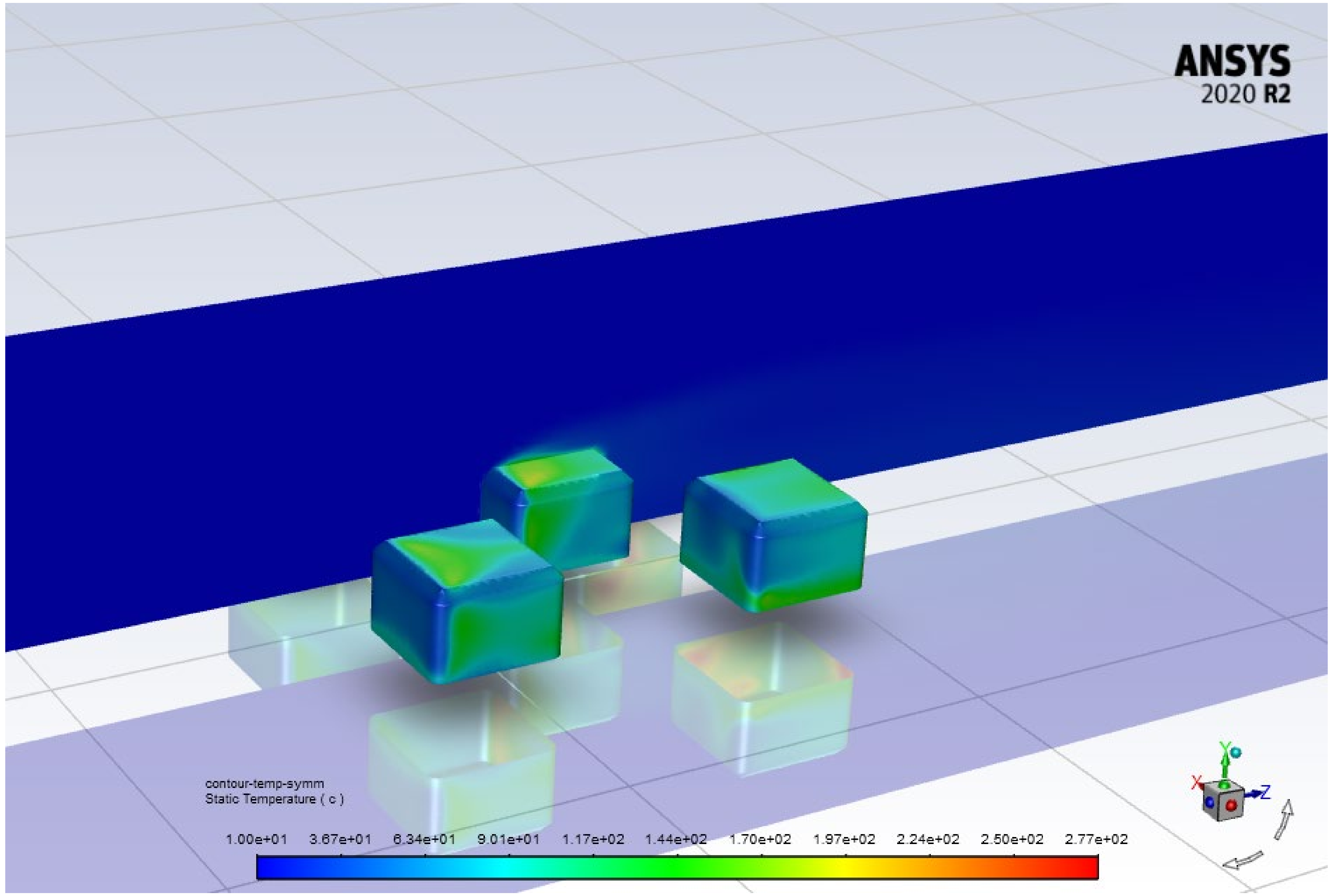
Faces



Faces and edges

Basic visualization in Fluent

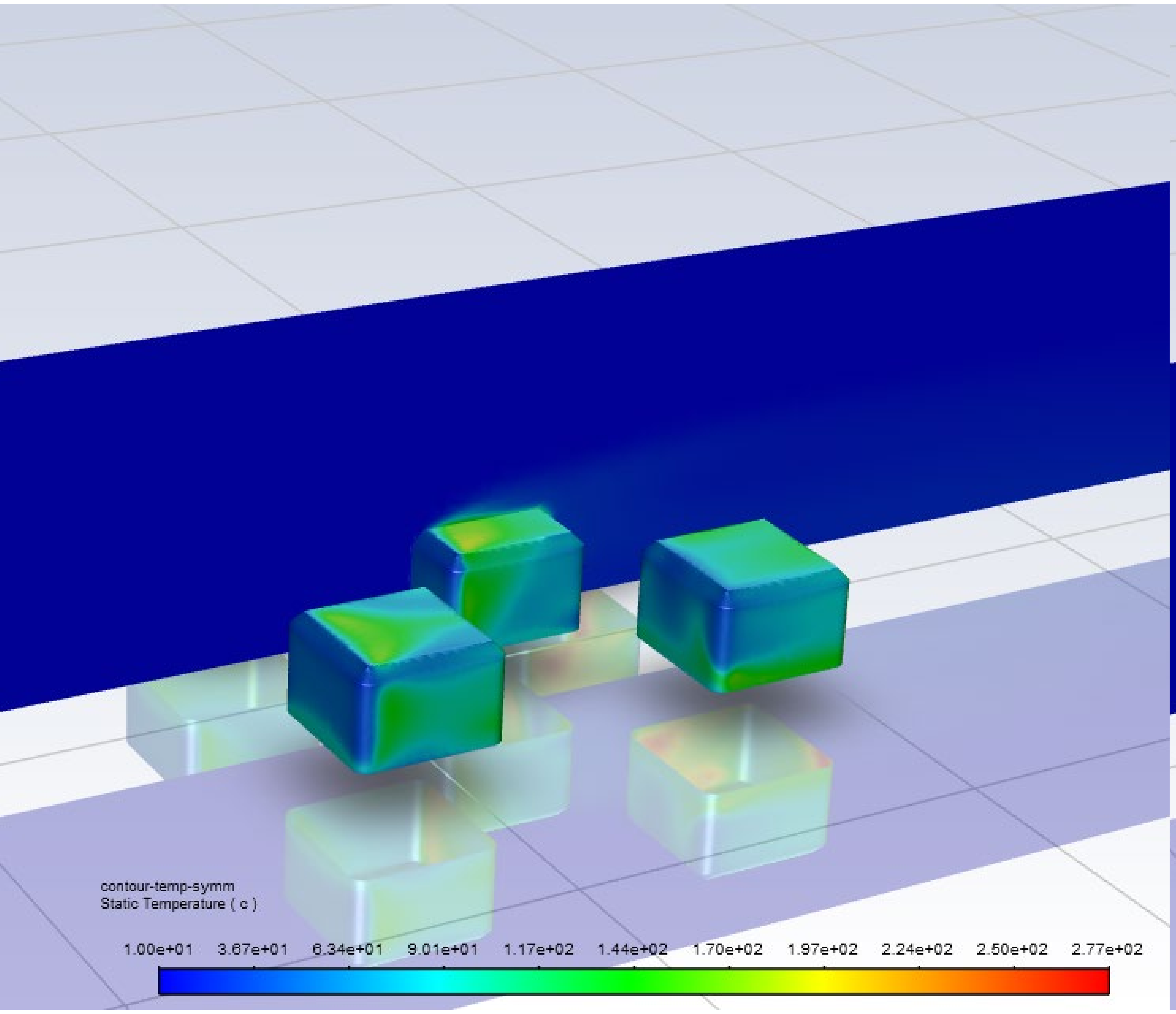
- Contours on surfaces: either on the geometry, or on objects created especially for visualization (do not belong to the geometry) such as lines, planes, iso-surfaces, etc.



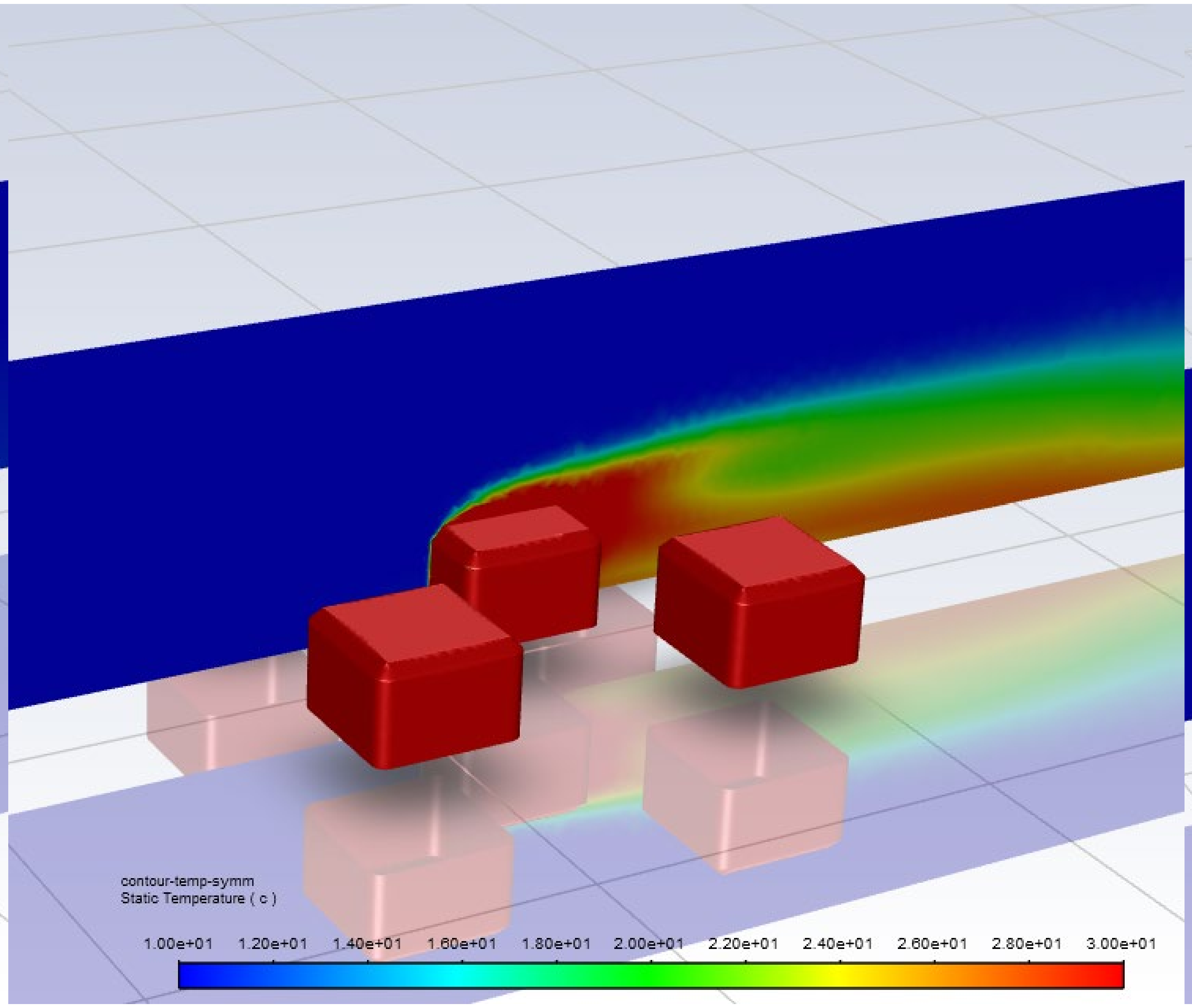
Contours of temperature

Basic visualization in Fluent

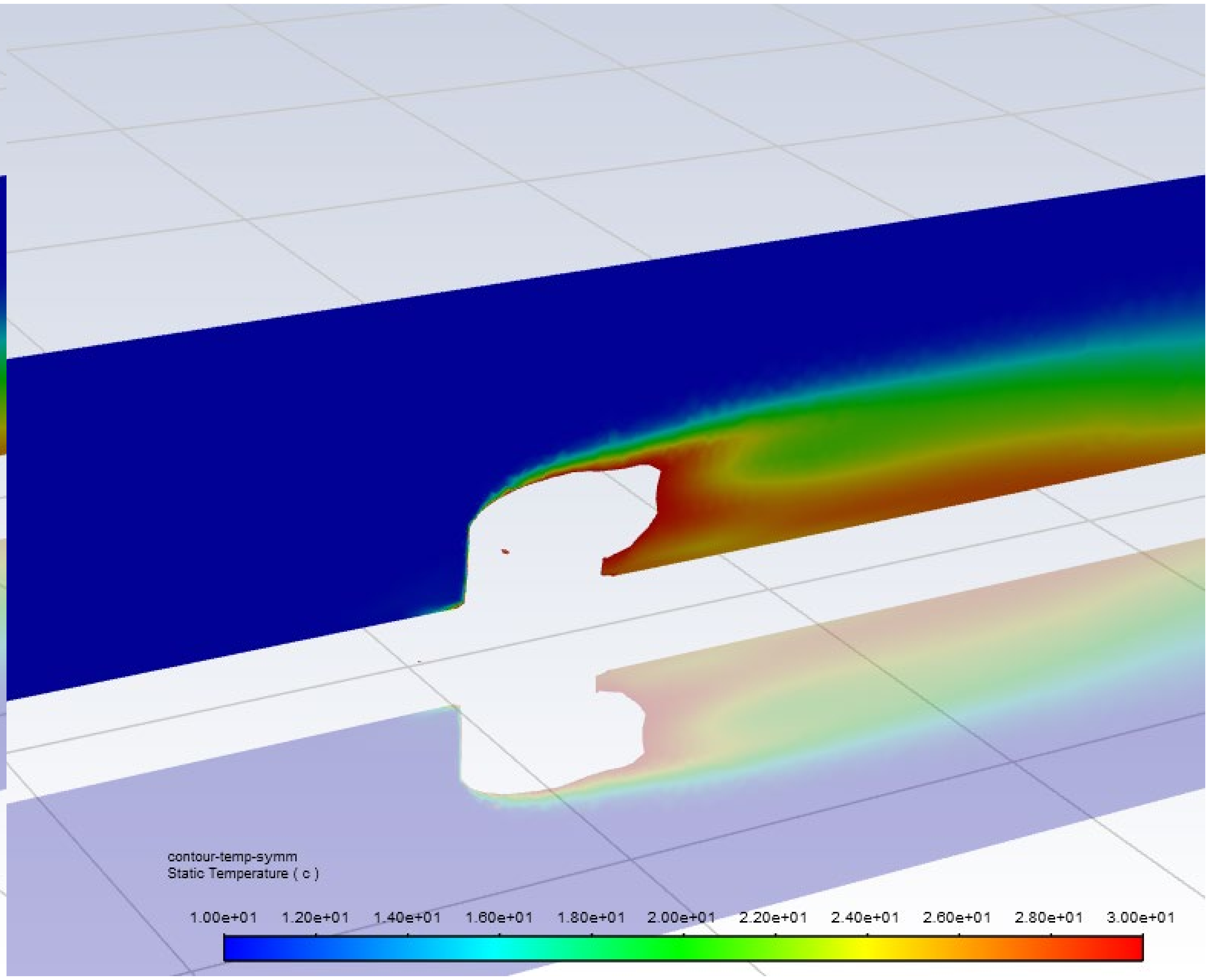
- Contours on surfaces: either on the geometry, or on objects created especially for visualization (do not belong to the geometry) such as lines, planes, iso-surfaces, etc.



Auto range (10-277 °C)



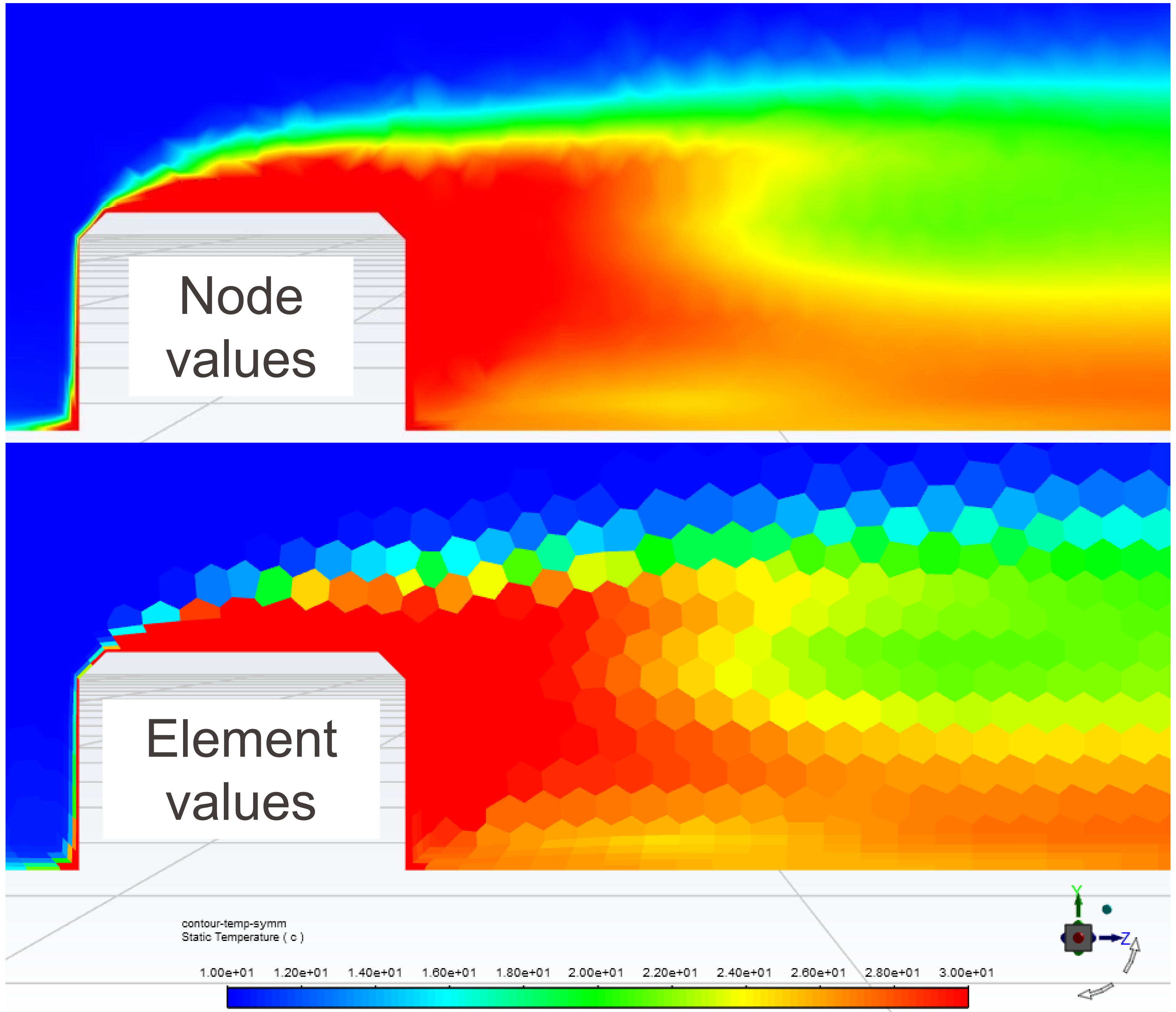
User-defined range 10-30 °C



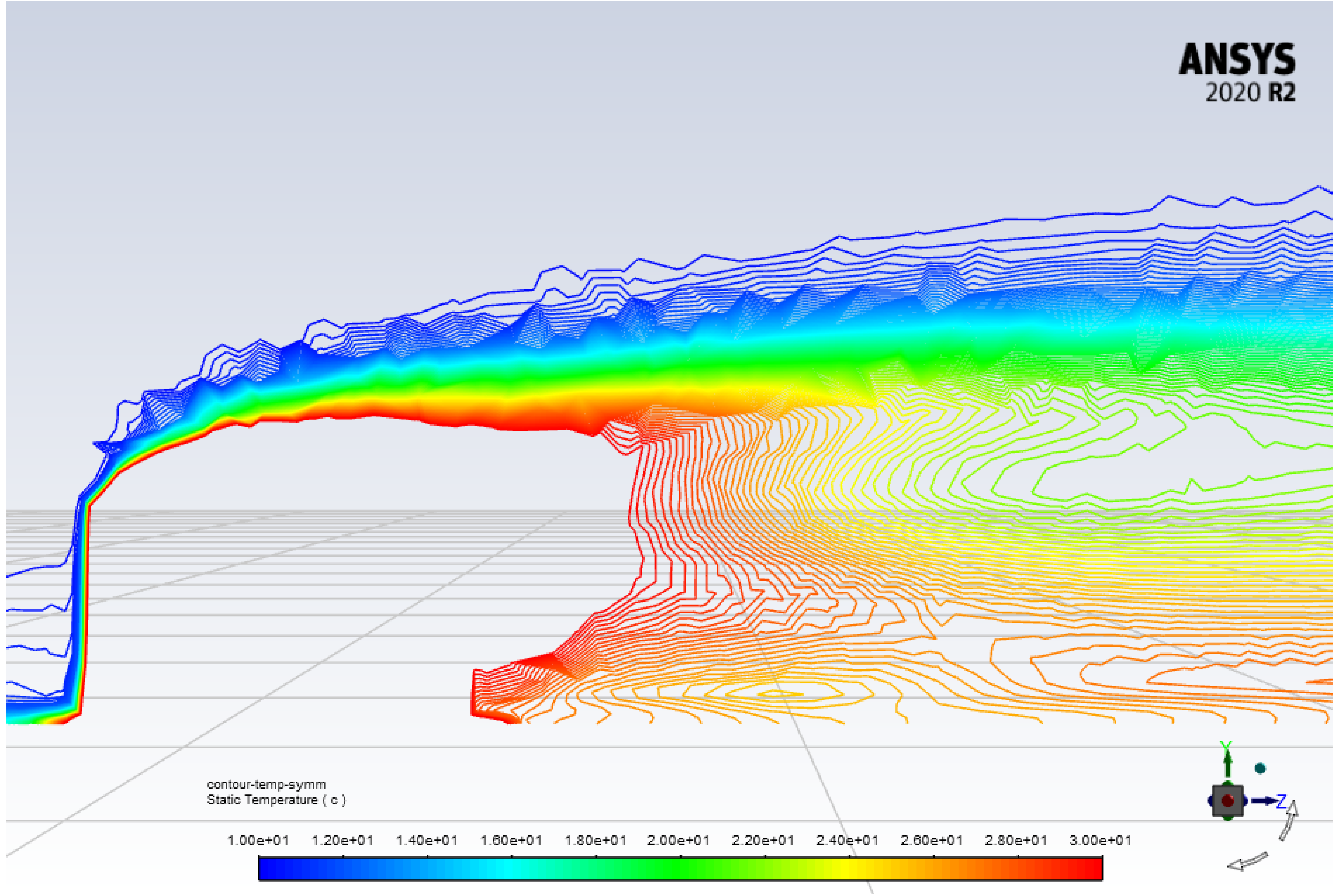
Clip to range 10-30 °C

Basic visualization in Fluent

- Contours on surfaces: either on the geometry, or on objects created especially for visualization (do not belong to the geometry) such as lines, planes, iso-surfaces, etc.



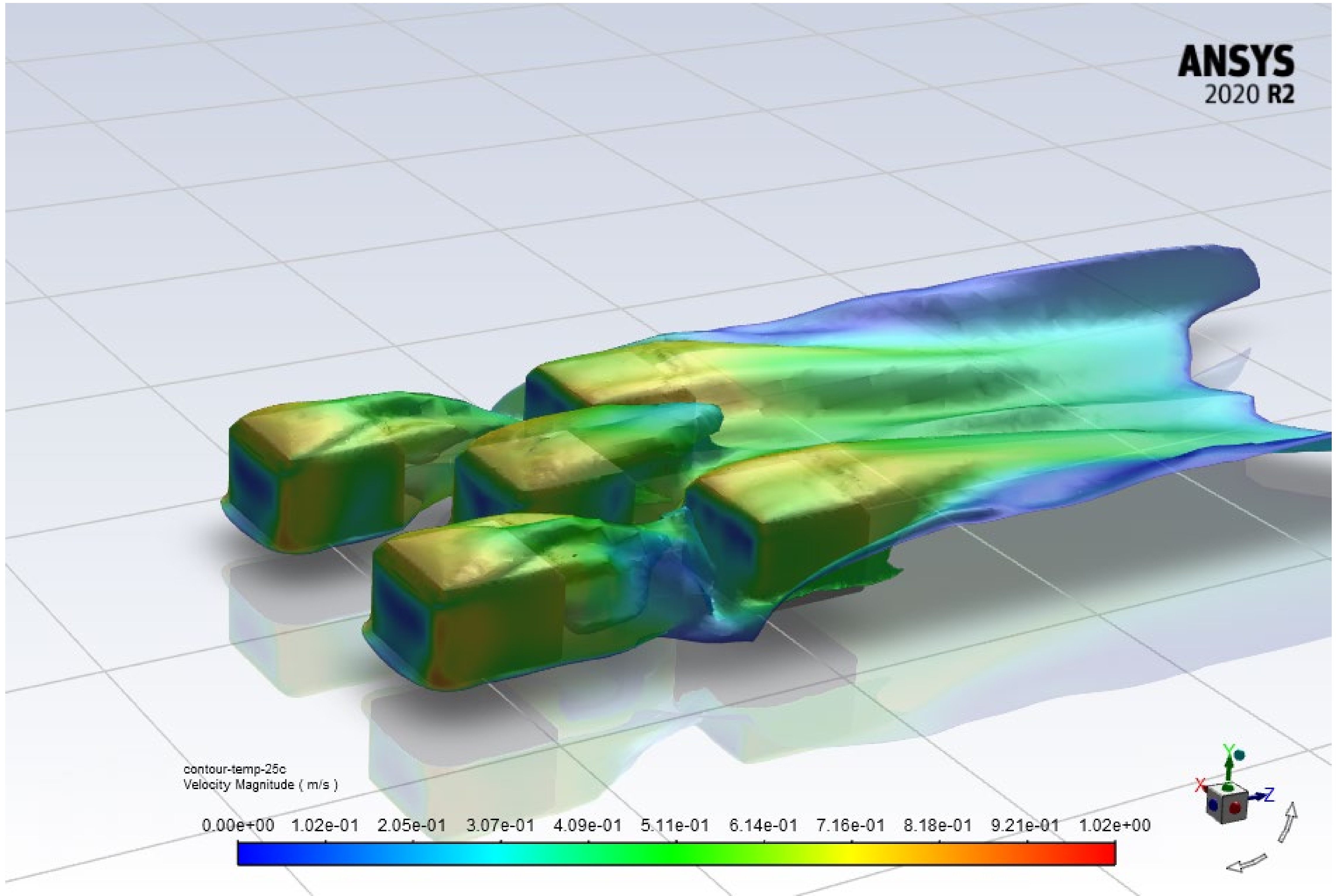
Filled contours



Unfilled contours

Basic visualization in Fluent

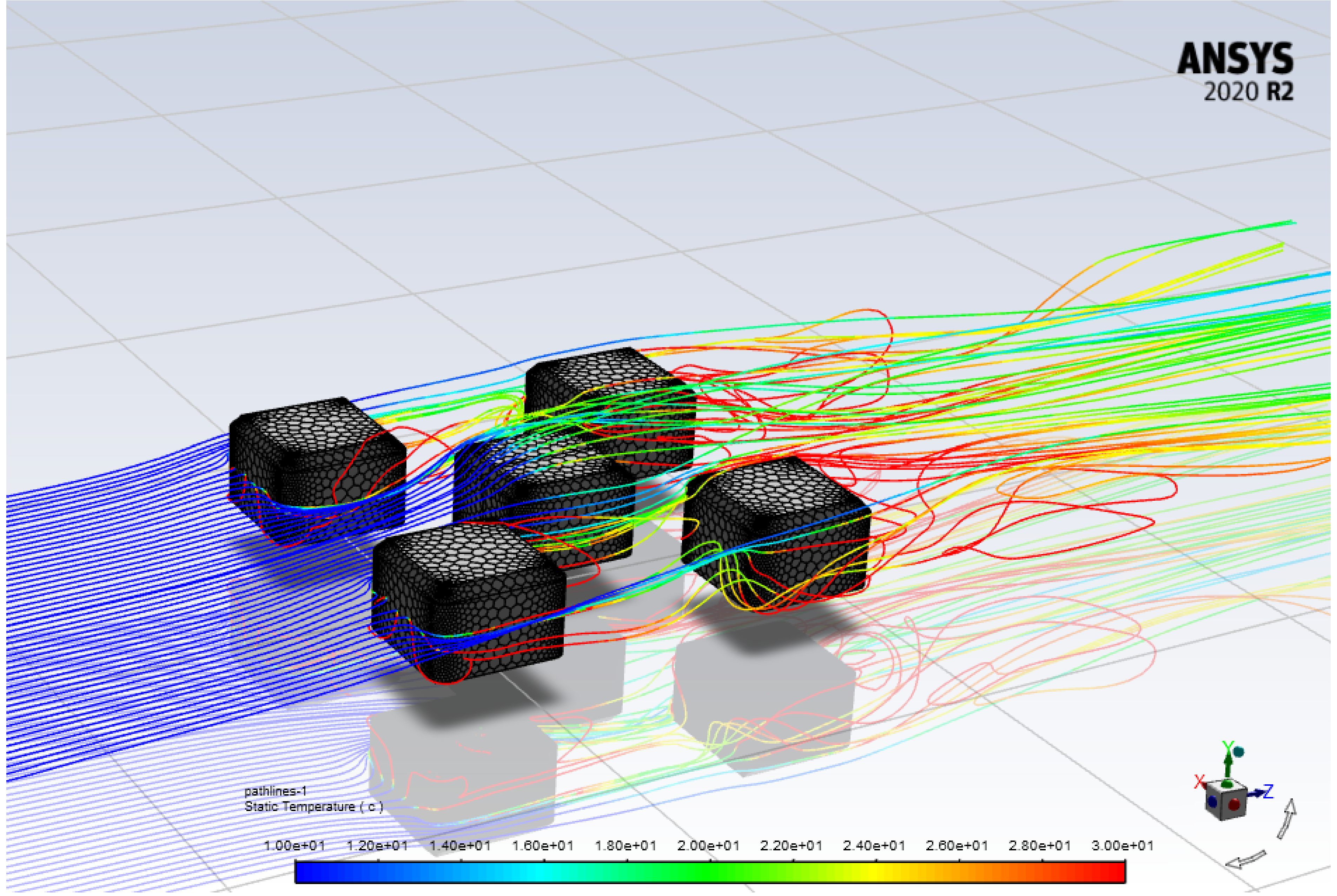
- Contours on surfaces: either on the geometry, or on objects created especially for visualization (do not belong to the geometry) such as lines, planes, iso-surfaces, etc.



Iso-surface of temperature *T*=25 °C colored by velocity magnitude

Basic visualization in Fluent

- Streamlines (called “pathlines” in Fluent, but remember that in unsteady flows **streamlines** (tangent to velocity field) ≠ **streaklines** (emission lines) ≠ **pathlines** (trajectories)).

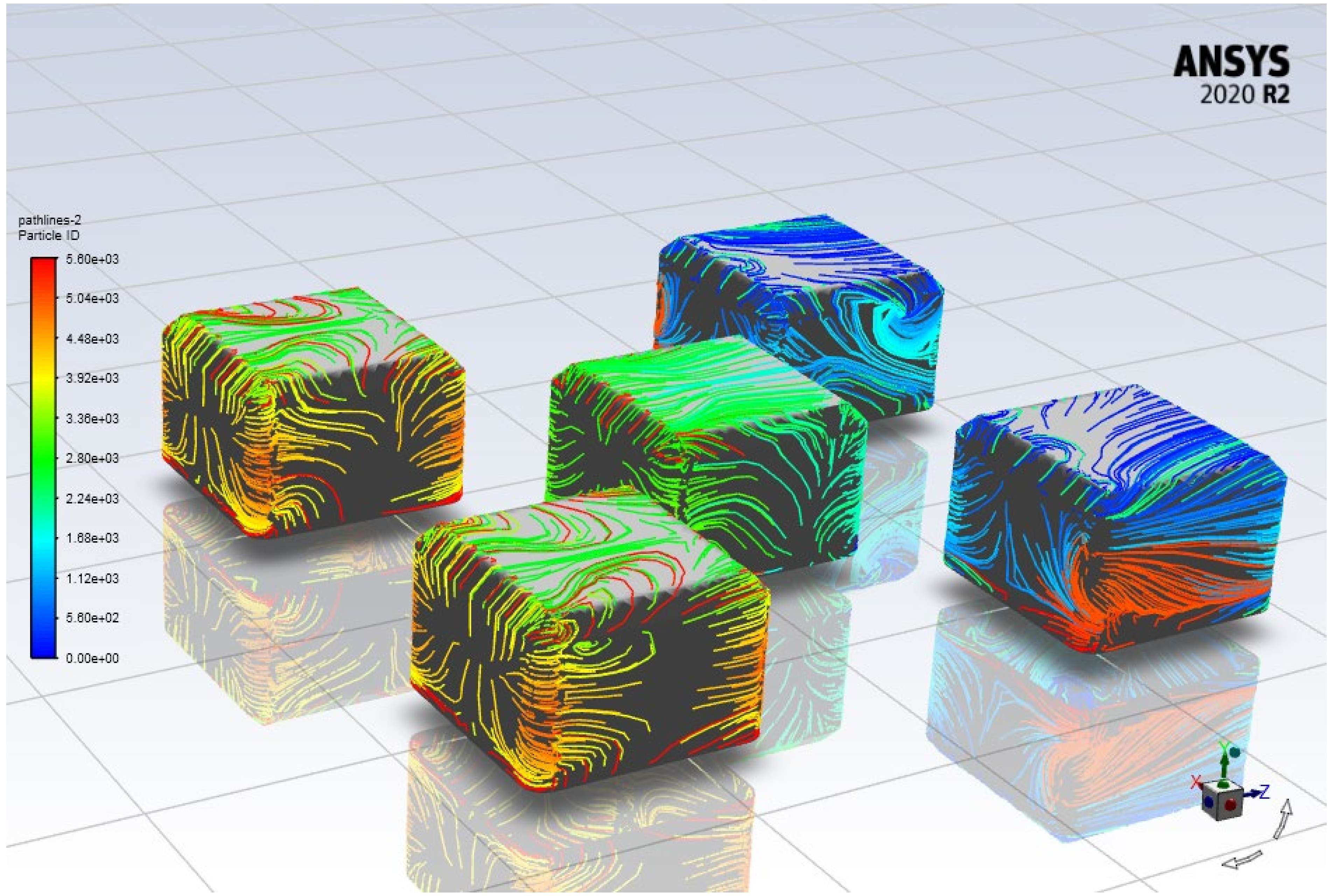


From a user-defined line located upstream, and colored by temperature.

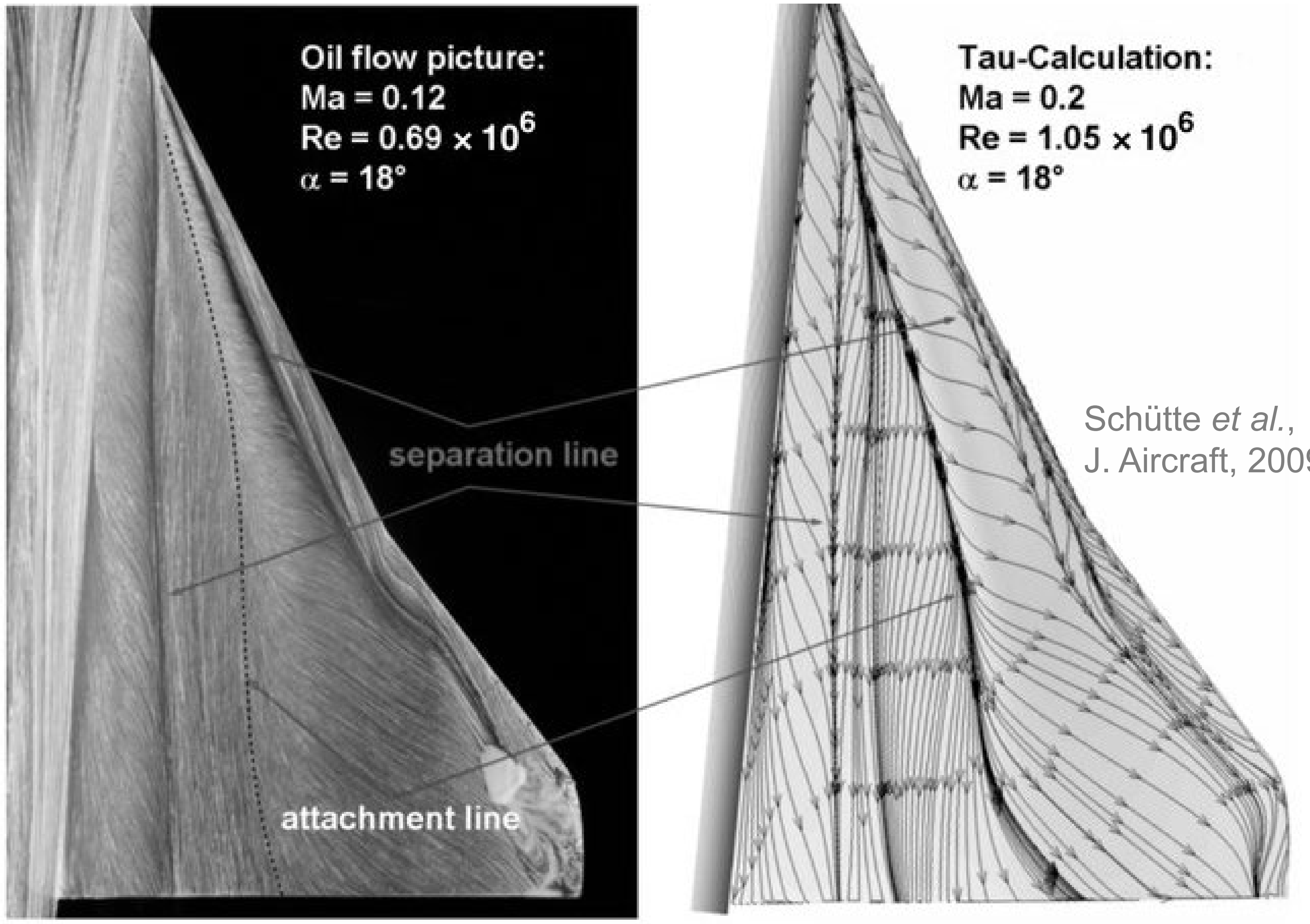
In 3D, may quickly become crowded.

Basic visualization in Fluent

- Streamlines (called “pathlines” in Fluent, but remember that in unsteady flows **streamlines** (tangent to velocity field) ≠ **streaklines** (emission lines) ≠ **pathlines** (trajectories)).



Constrained to a surface (“oil flow”).
Colored by “particle ID”.



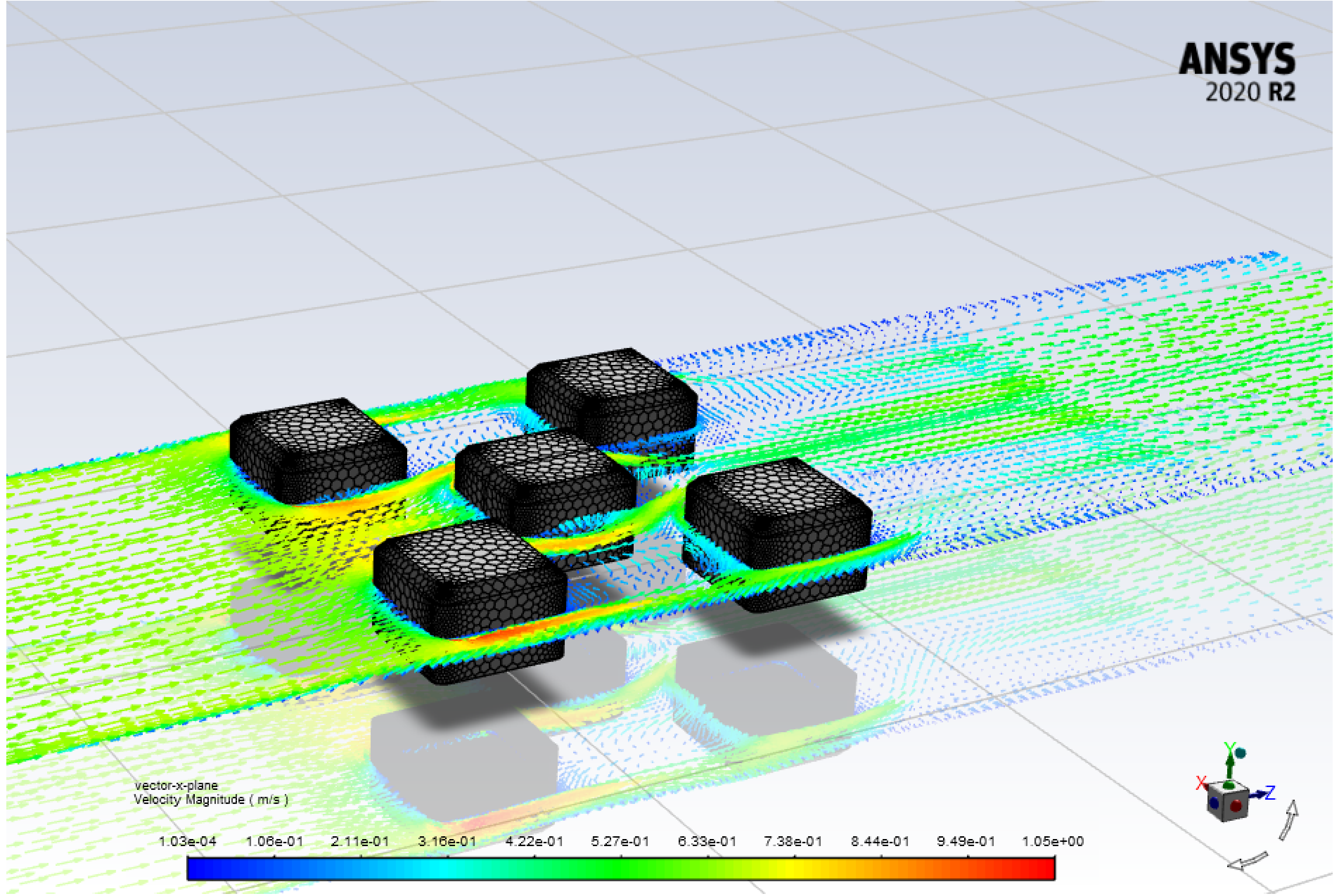
Schütte *et al.*, J. Aircraft, 2009

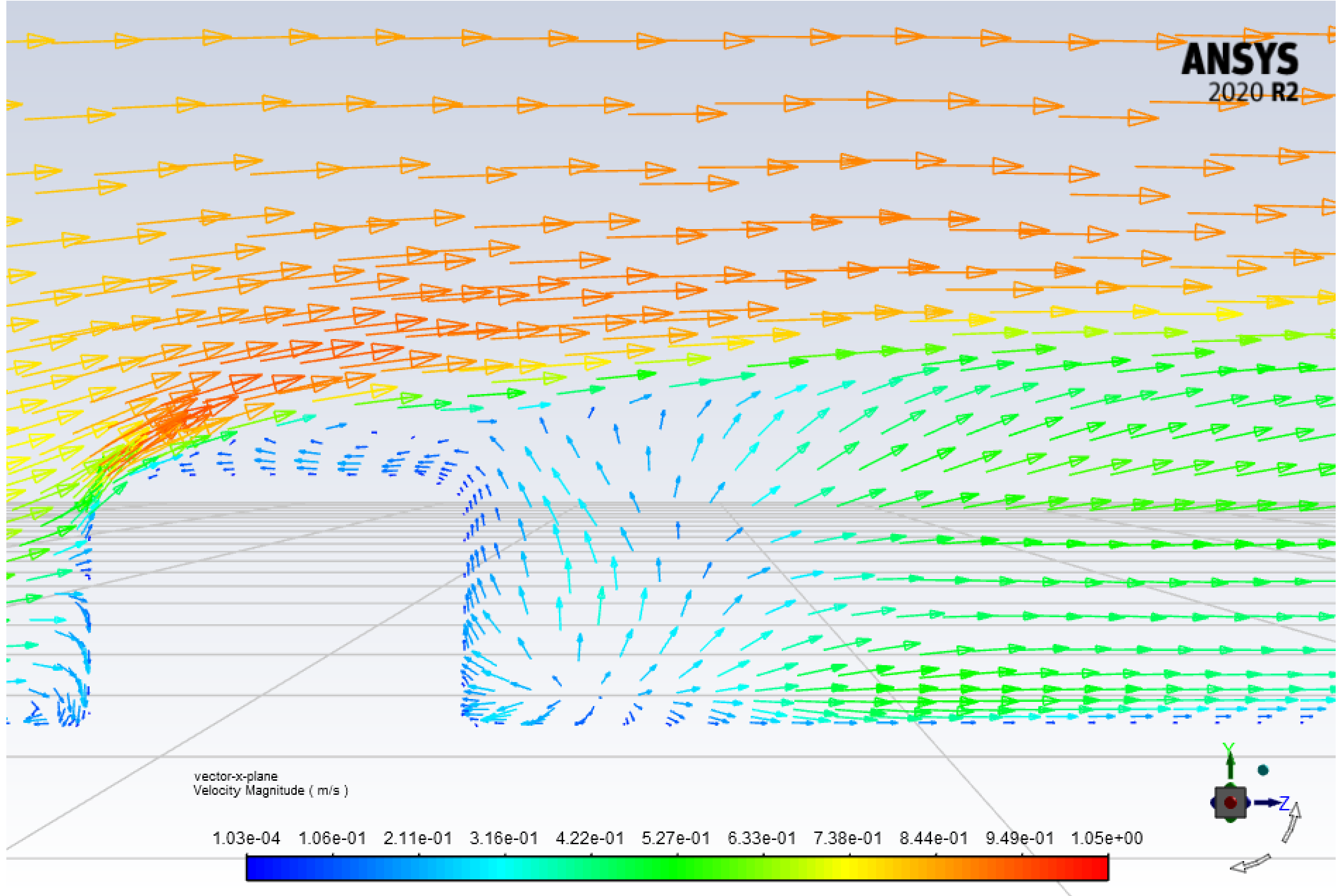
Mimic experimental oil flow.

Useful to visualize flow separation/reattachment.

Basic visualization in Fluent

- Velocity vector field

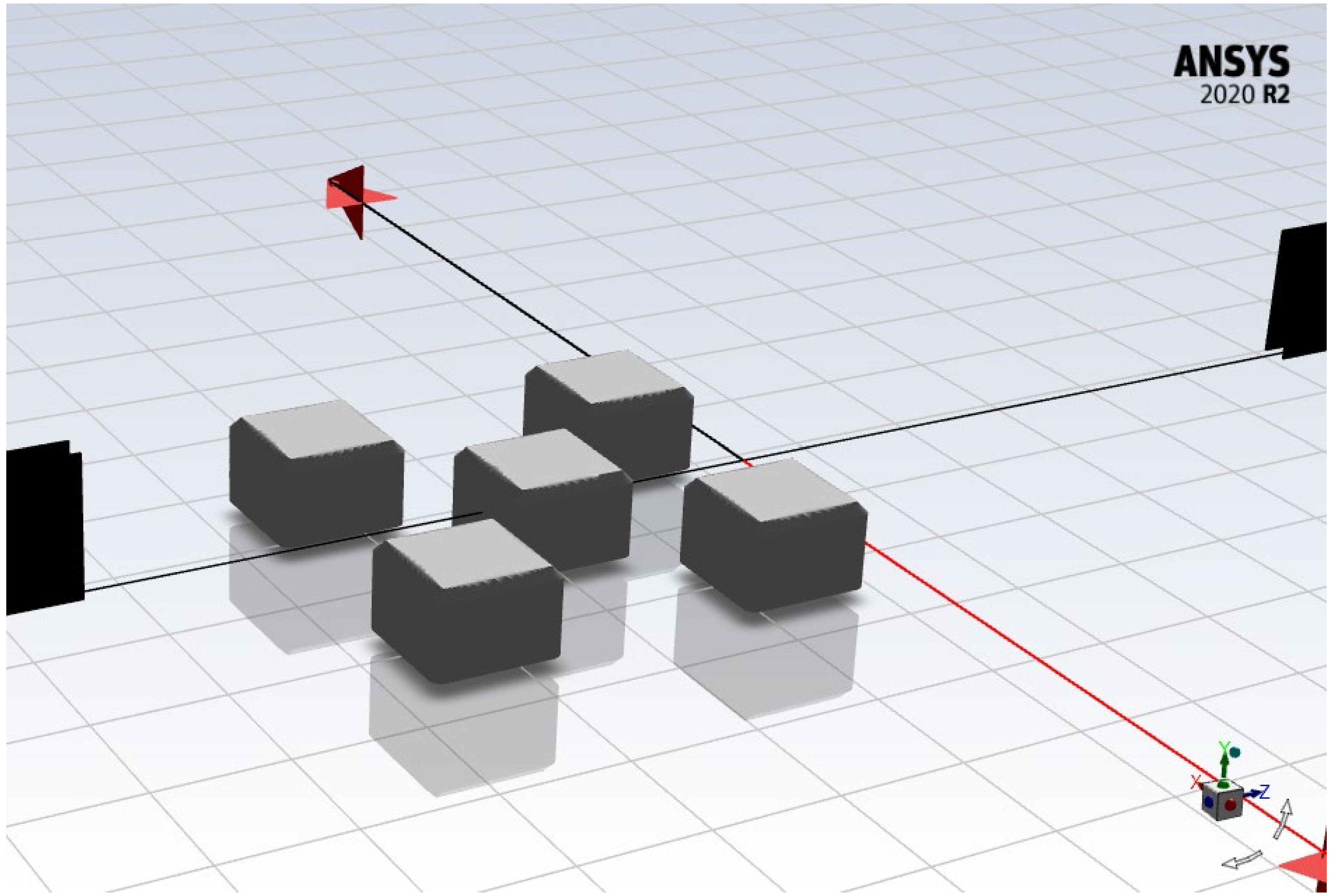


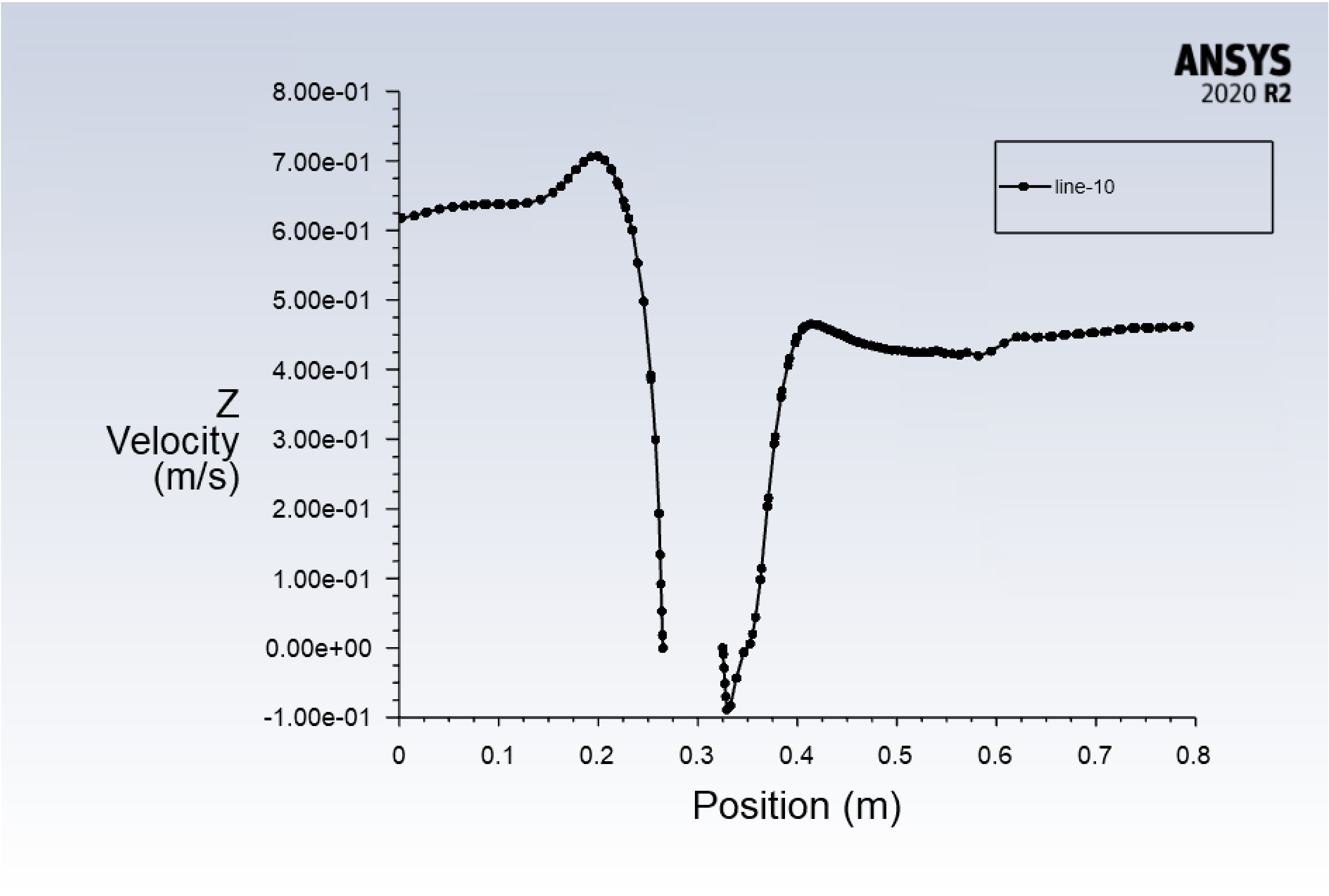


Useful to show velocity magnitude and direction simultaneously.
Must adapt the parameters (density and length of the arrows) to make an image that is easy to interpret.

Basic visualization in Fluent

- Plot data along lines (can then be exported to a file).

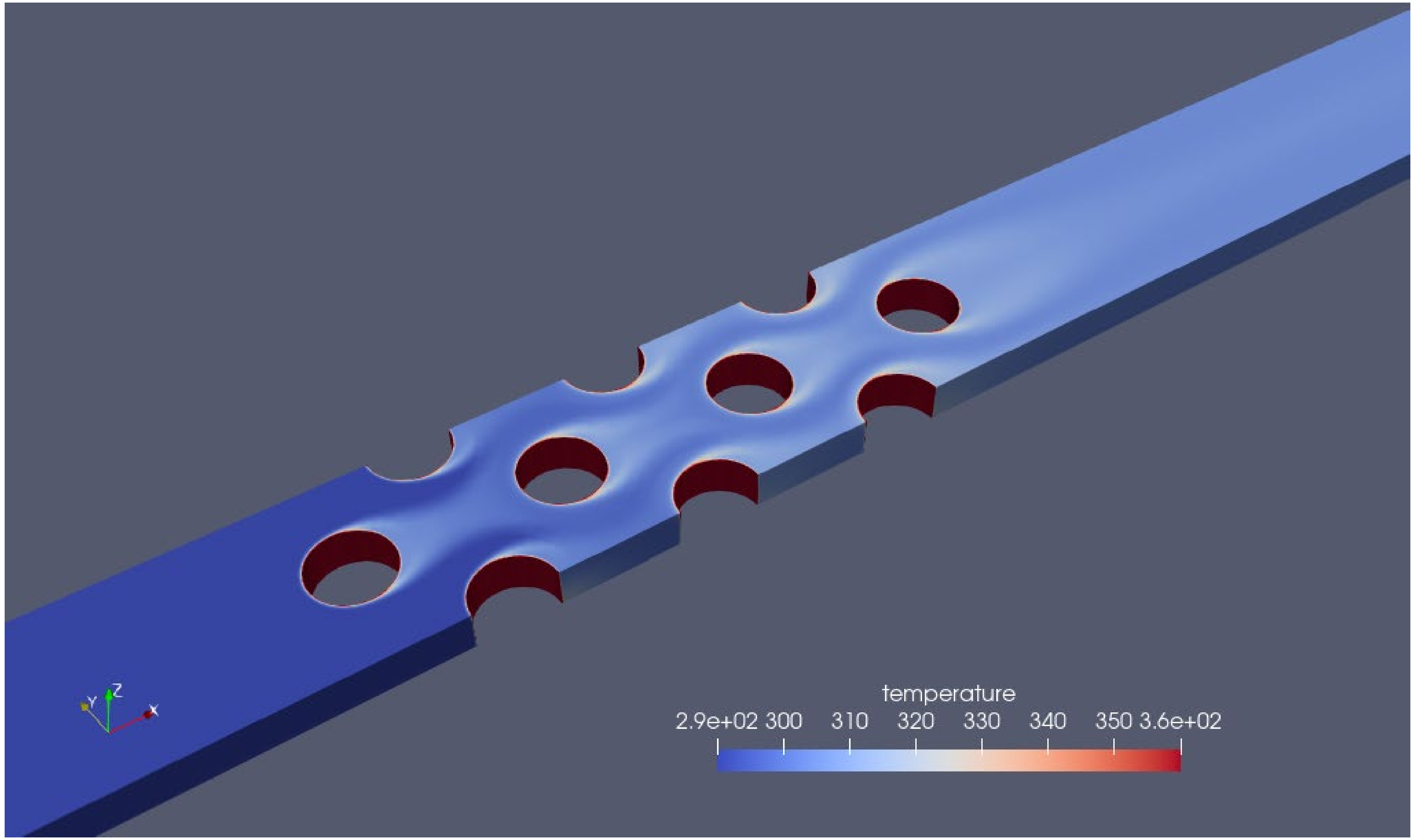


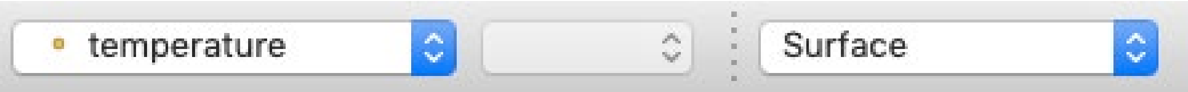


Streamwise velocity along a line oriented in the streamwise direction.

Basic visualization in Paraview

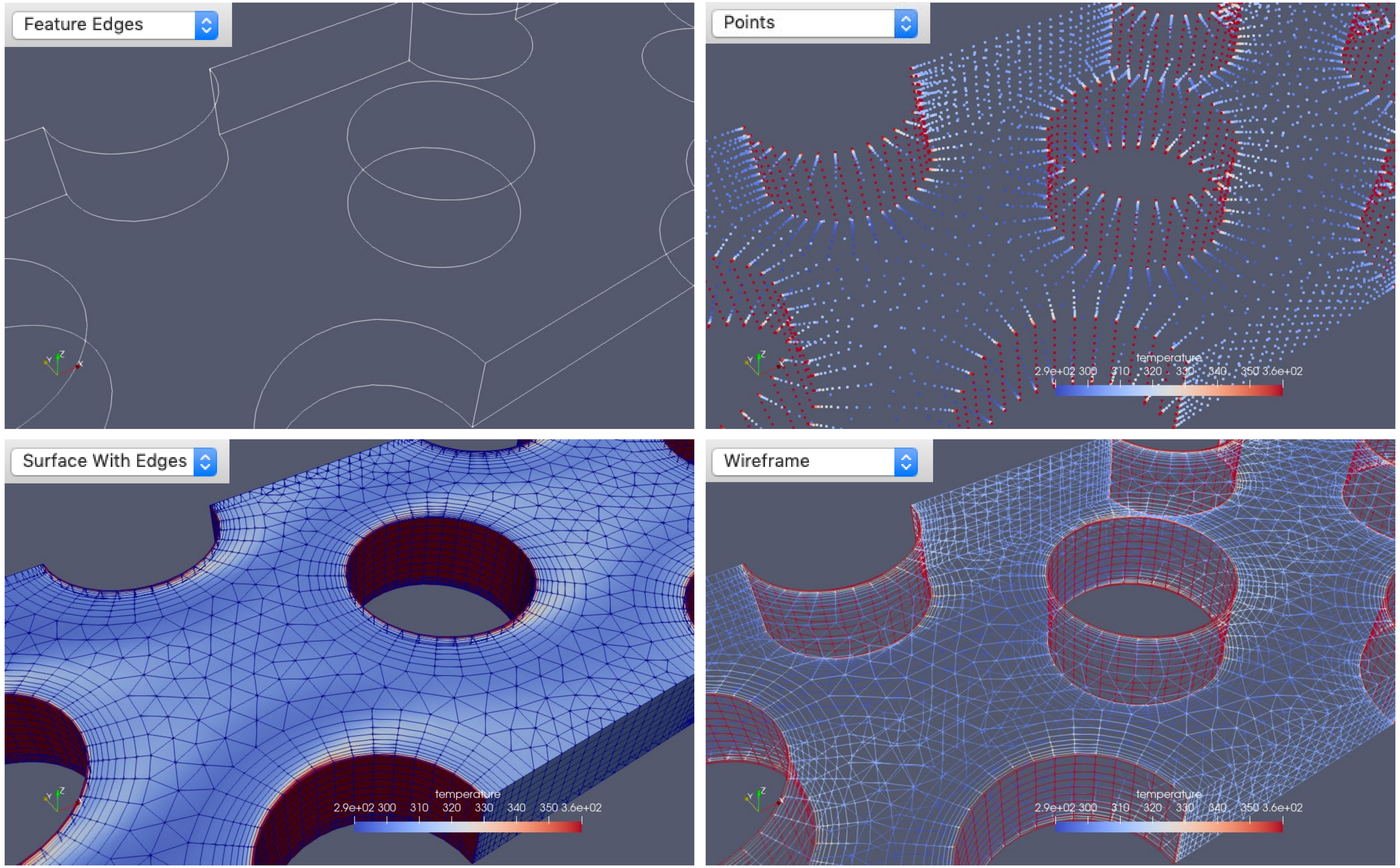
- Plot geometry, mesh, volume data





Basic visualization in Paraview

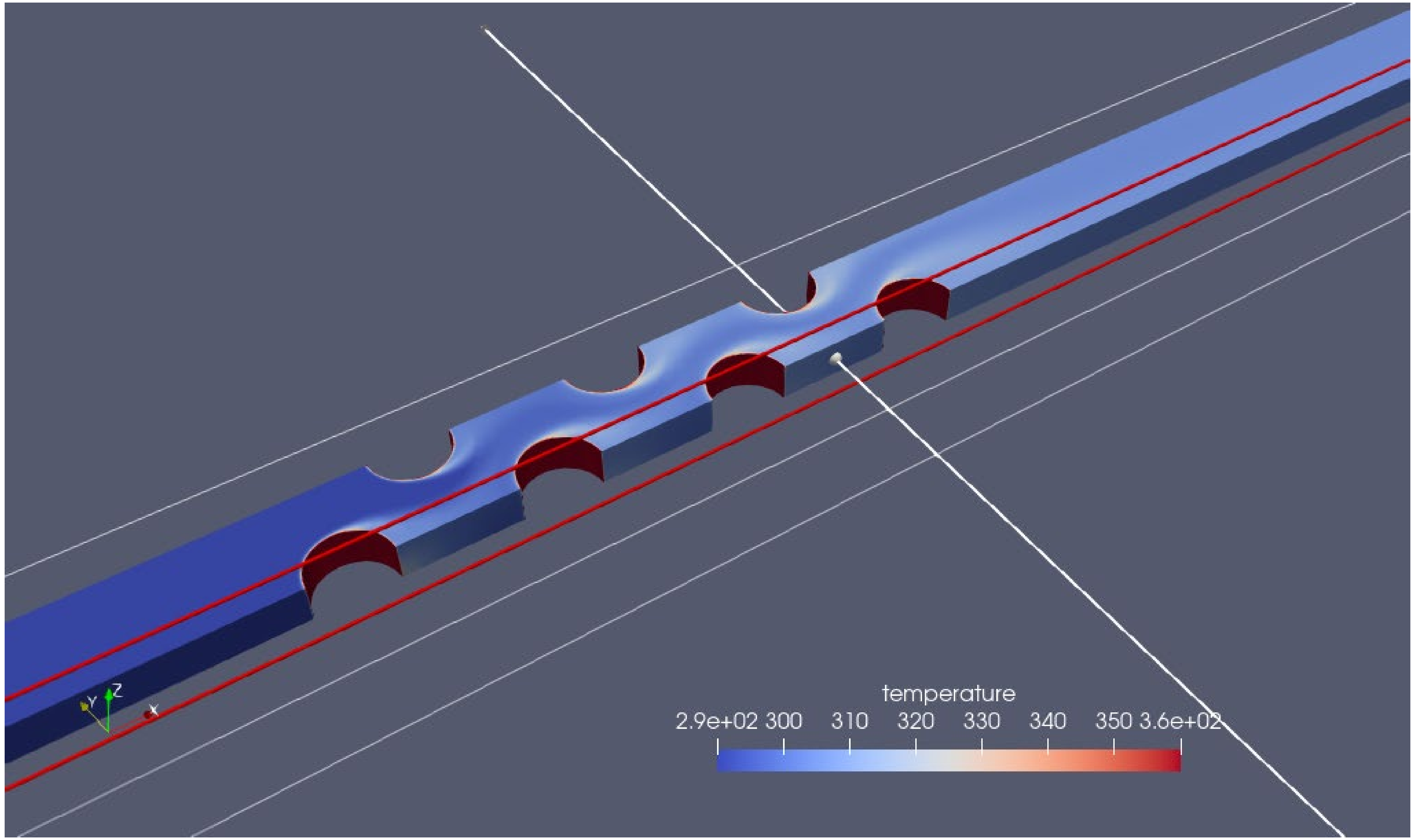
- Plot geometry, mesh, volume data



Numerical Flow Simulation

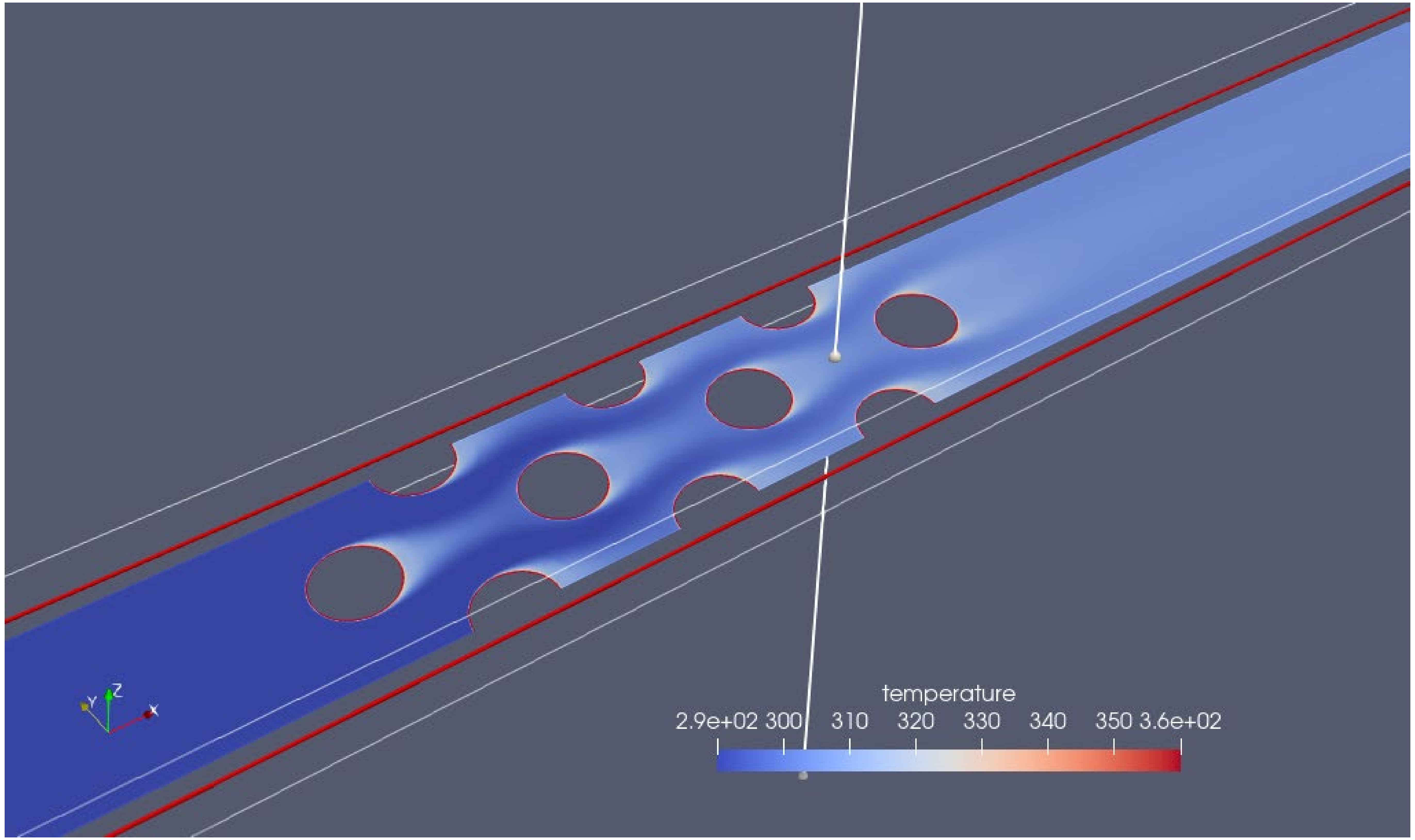
Basic visualization in Paraview

- Clip



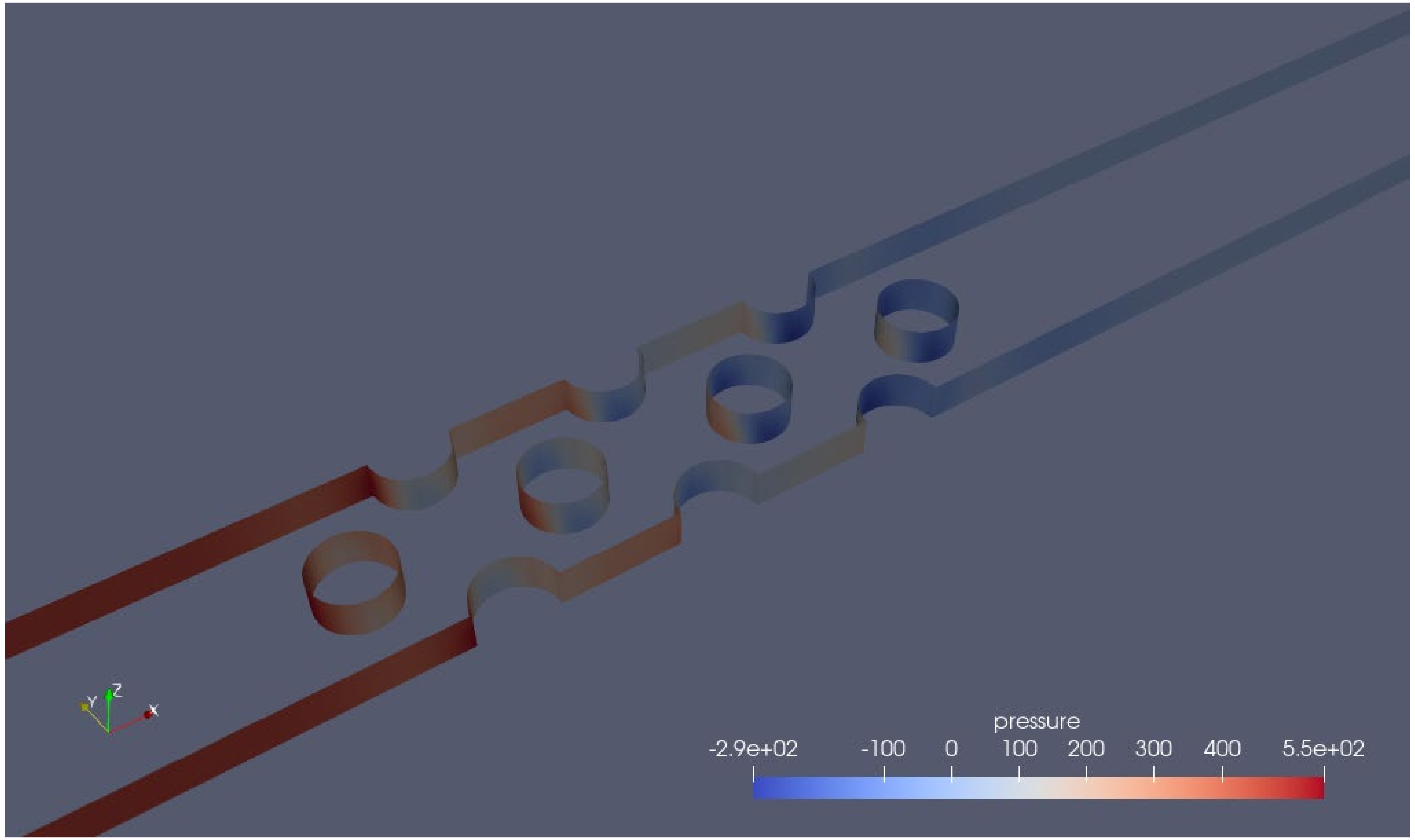
Basic visualization in Paraview

- Slice



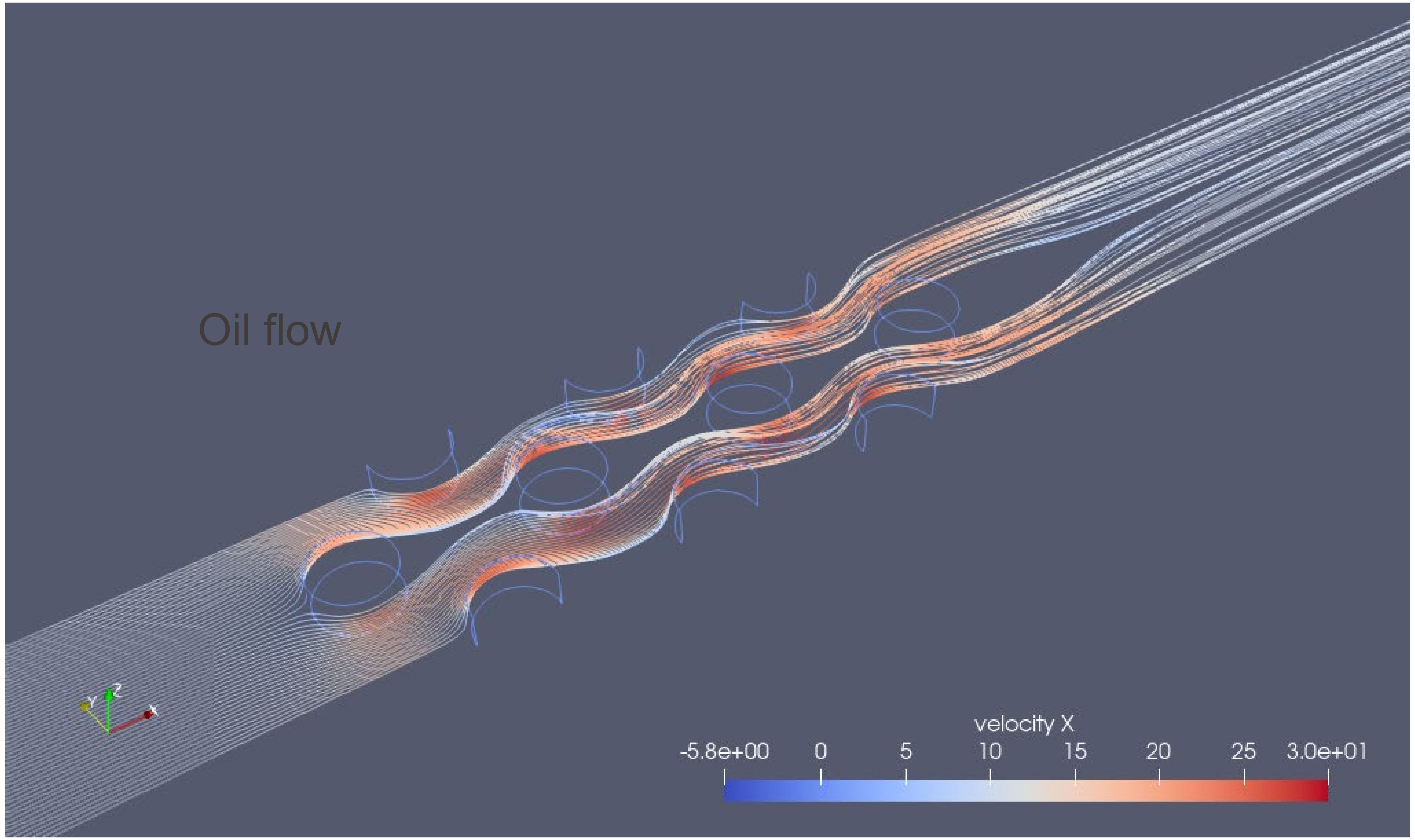
Basic visualization in Paraview

- Plot data on selected surfaces



Basic visualization in Paraview

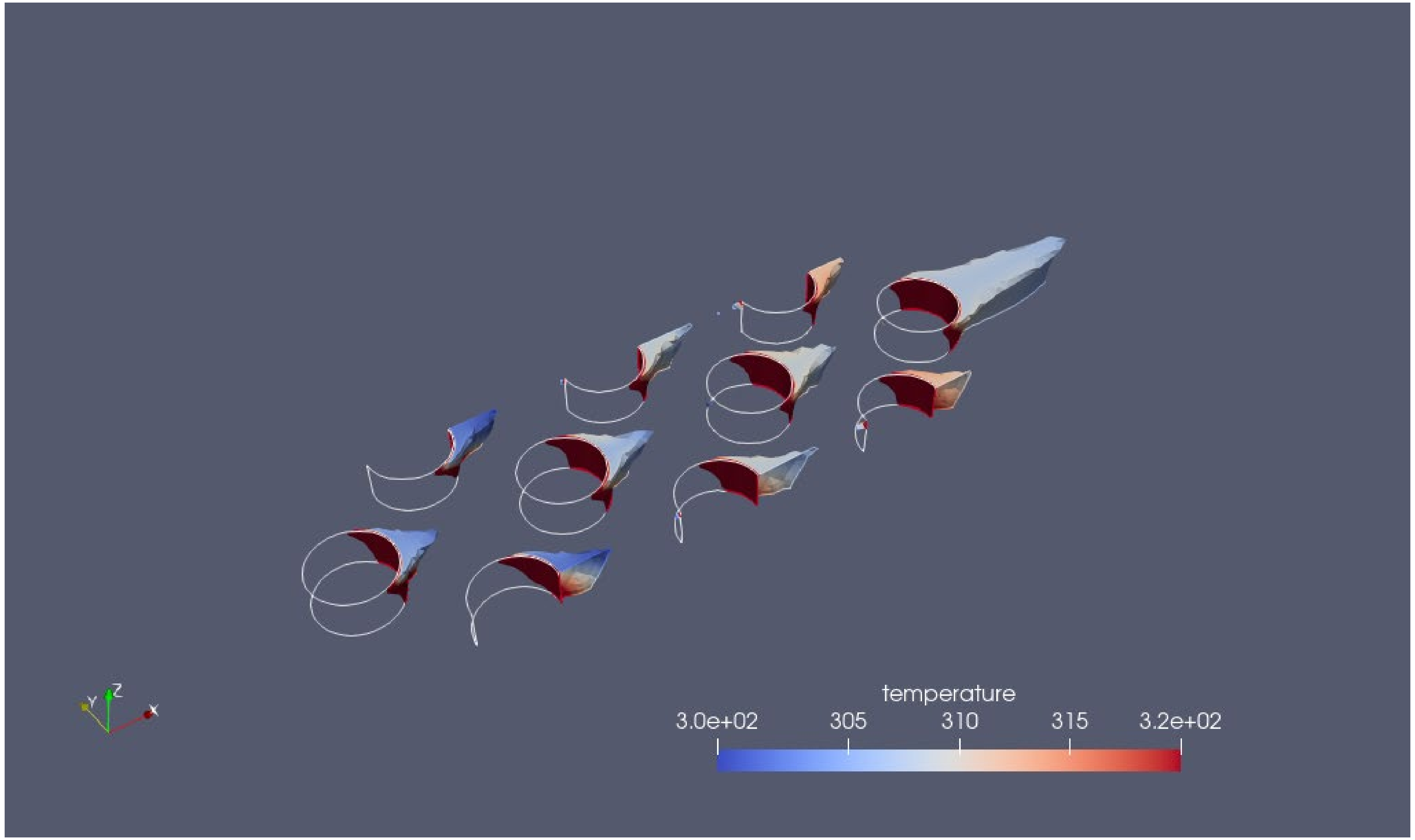
- Compute and plot streamlines



Numerical Flow Simulation

Basic visualization in Paraview

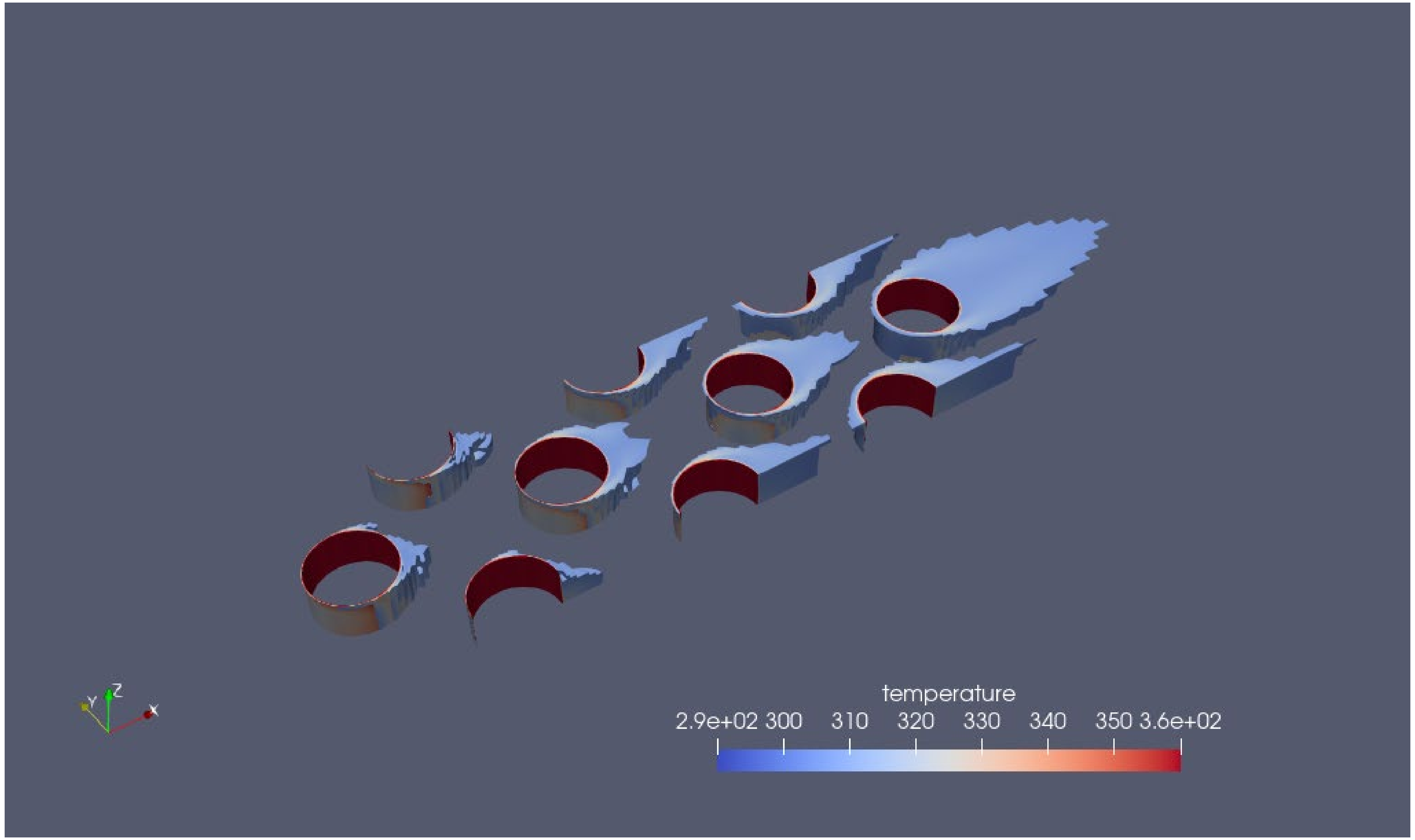
- Extract and plot iso-contours/iso-surfaces



Example:
iso-surfaces of zero streamwise velocity, colored by temperature

Basic visualization in Paraview

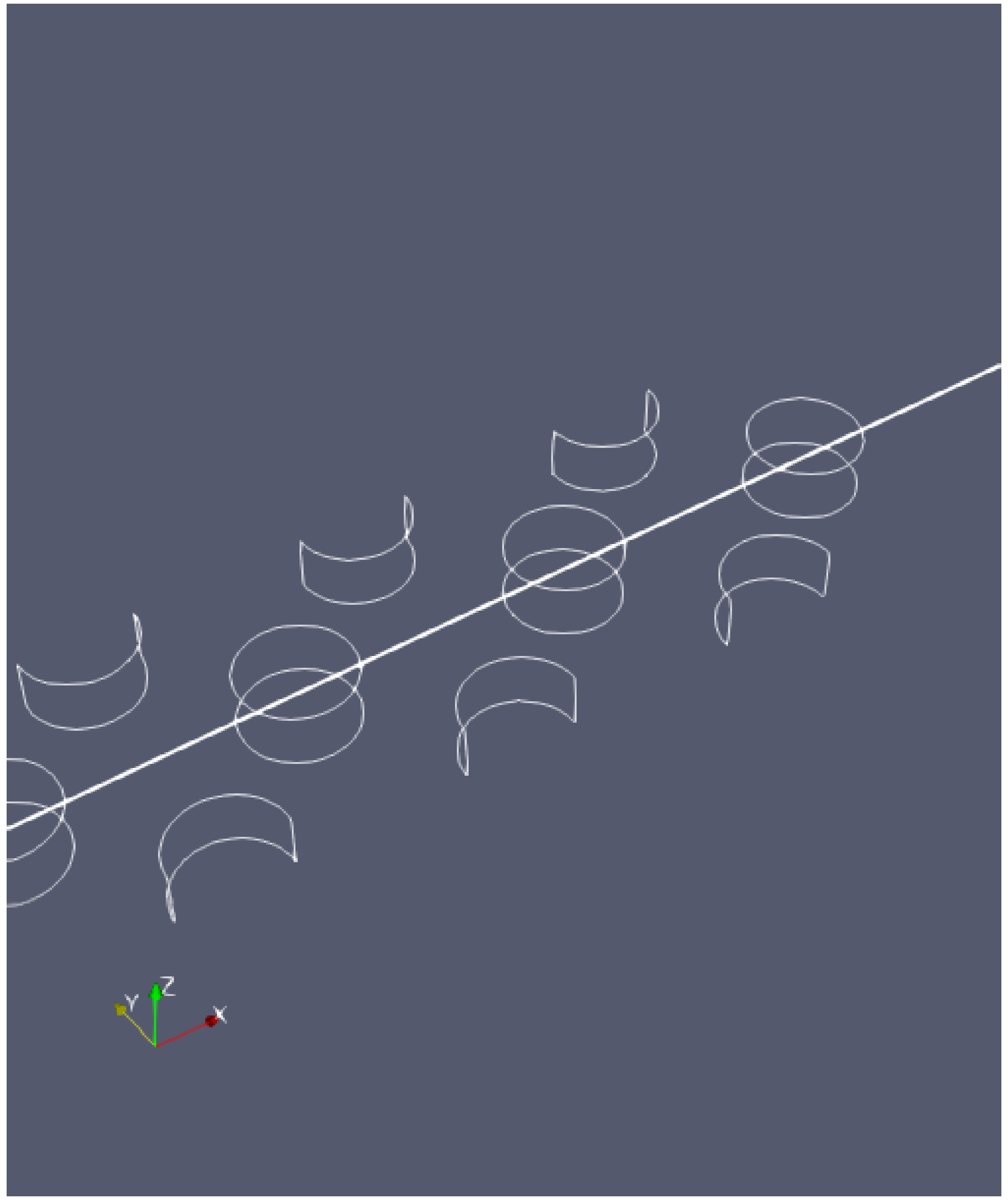
- Threshold

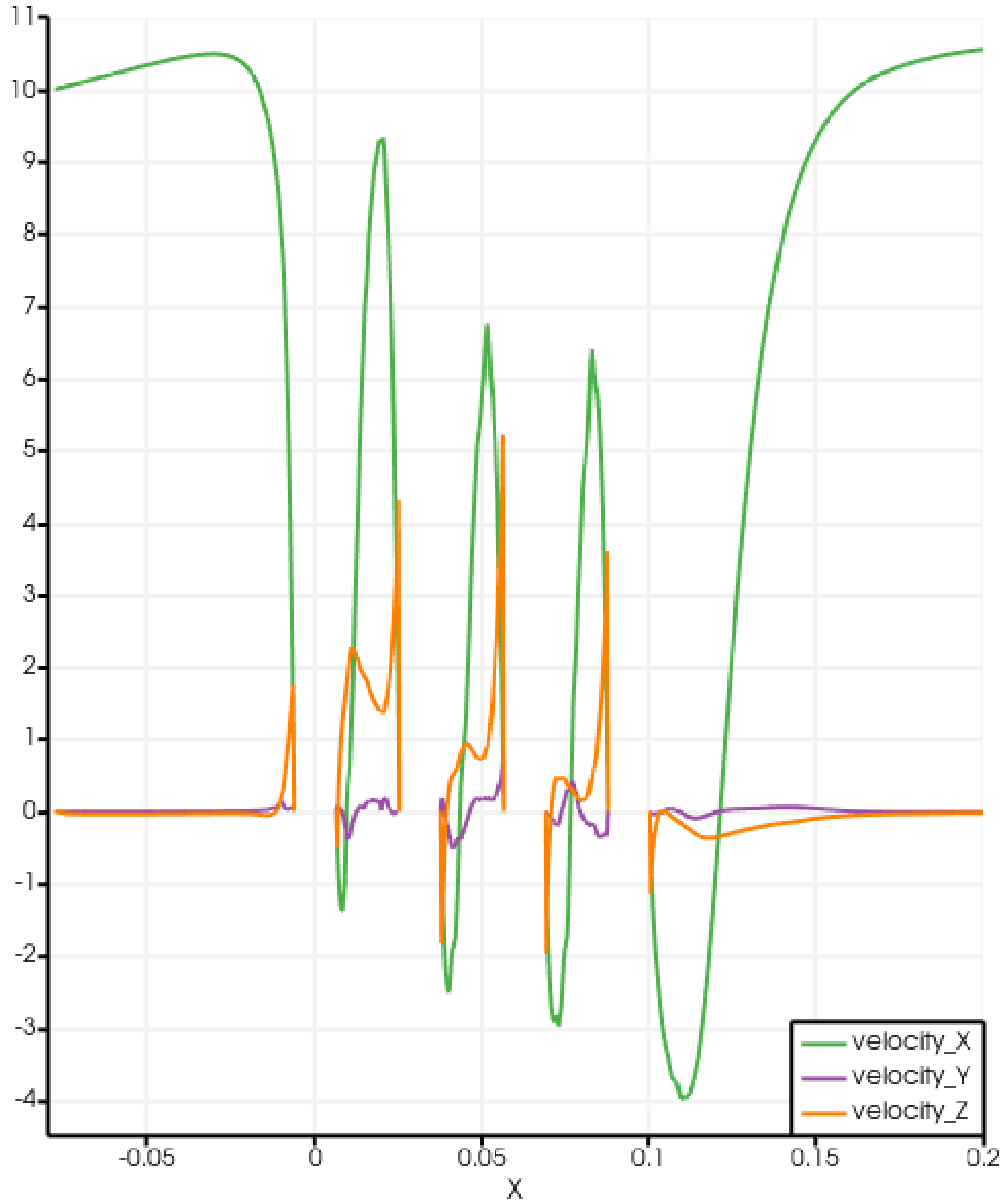


Example:
temperature > 305 K

Basic visualization in Paraview

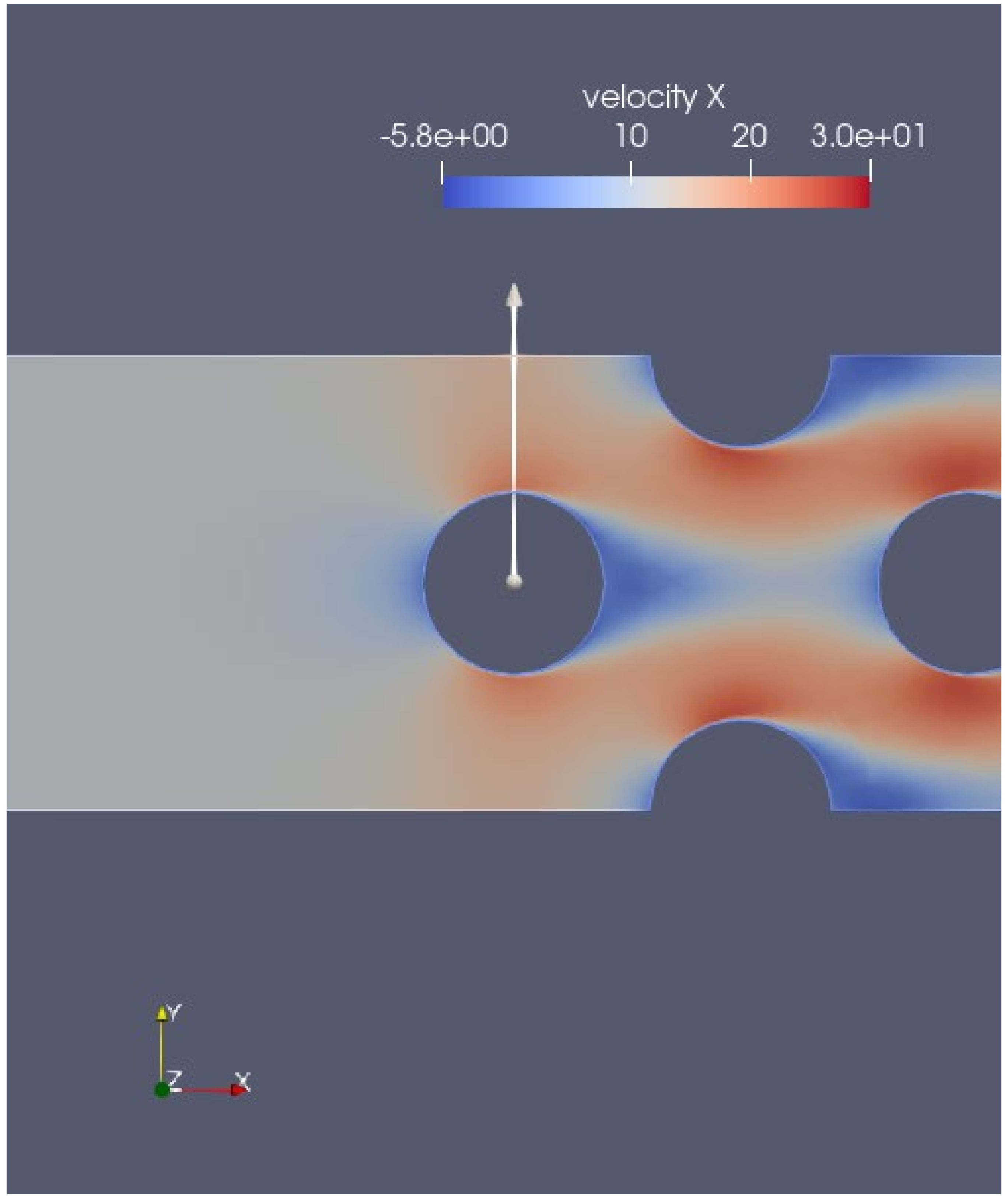
- Plot data along lines

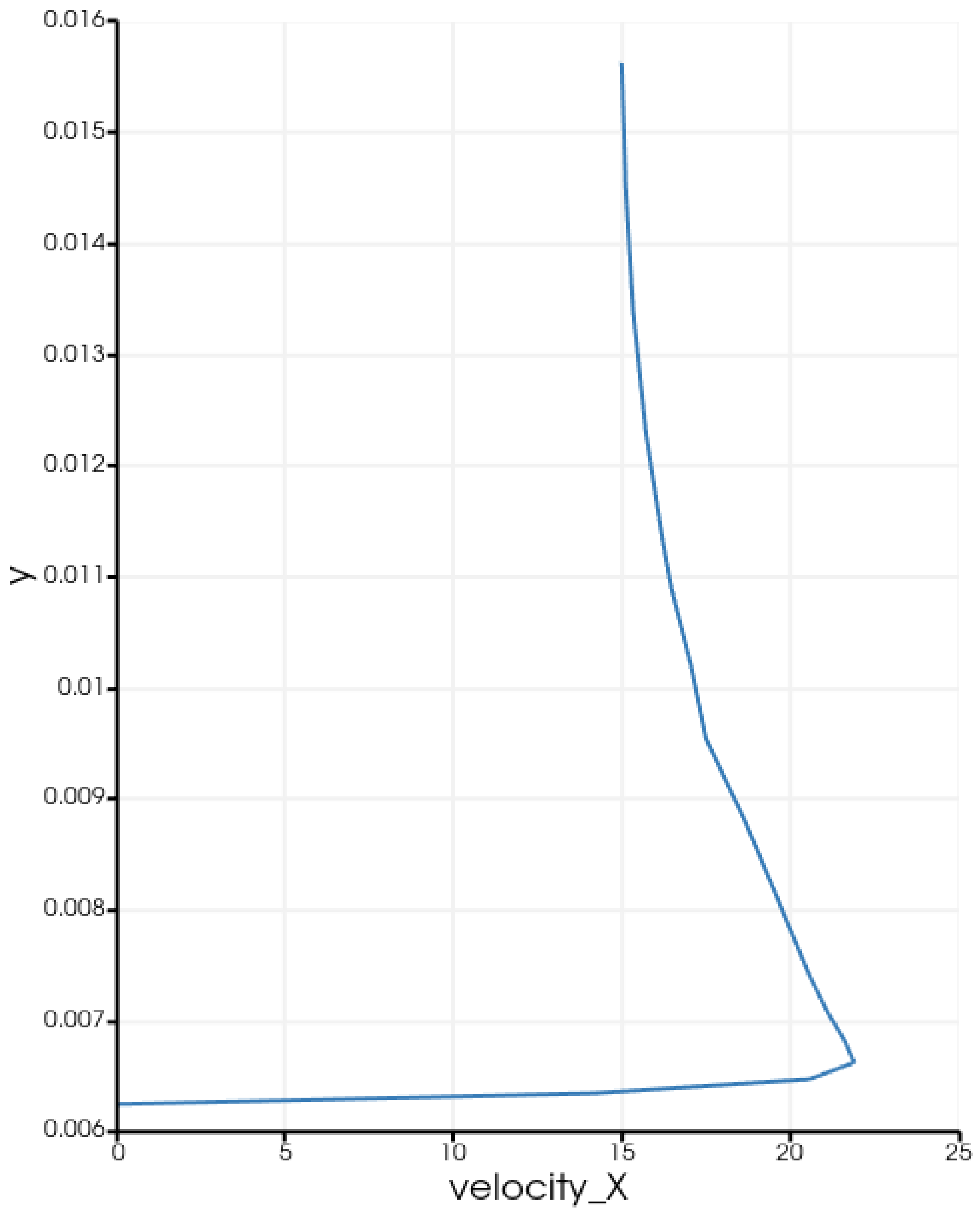




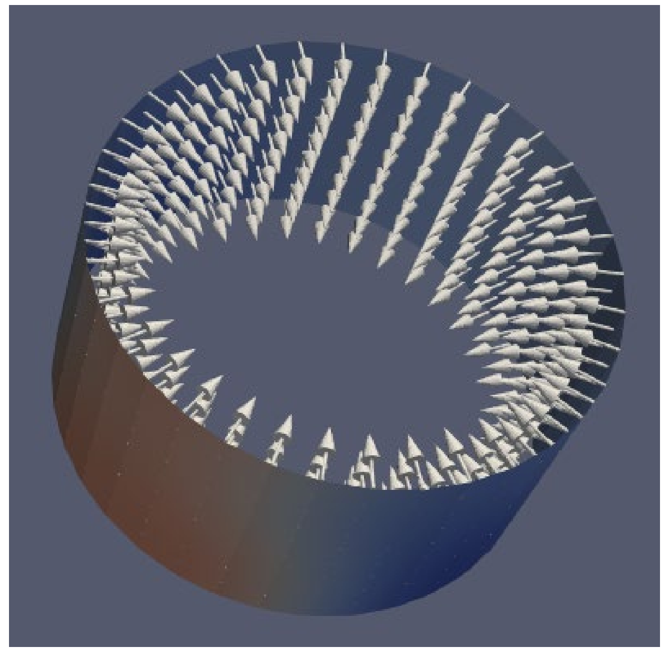
Basic visualization in Paraview

- Plot data along lines





More options in ParaView



- **Filters**: operations that transform data. For example:
  - Extract surfaces
  - Compute surface normal/tangent vectors
  - Compute derivatives/gradients (e.g. vorticity, stress)
  - Compute surface/volume integrals
  - Compute temporal average & standard deviation
  - …
  - Calculator: apply user-defined math function/operation to data. For example:
    - Compute magnitude of a vector
    - Build a vector from scalar components
    - Compute radial/azimuthal velocity components from Cartesian components
    - Compute the difference between 2 fields (e.g. to compare 2 solutions obtained with different methods, to compute fluctuations about a time average, etc.)

More options in ParaView

- **Batch scripts** (Python):
  - Automate/repeat visualization tasks
  - Perform complex operations on data (not available in Calculator)
  - Run parallel visualizations tasks
- **“Trace”**: record the actions performed manually in the GUI → obtain a batch script that can be later modified/executed.
- **“Plot over time”** and **Animations**

Summary

- Visualization can be used for different goals:
  - Scientific visualization (physical insight, scientific discovery)
  - Visual communication (pedagogy, outreach, marketing)

- Many complementary techniques can be employed and combined: photorealism, stereoscopy, 360° views, animations, immersion & interaction

- To optimize visualization effect:
  - create “uncommon” (striking and memorable) images/animations
  - use imagination & creativity
  - choose appropriate hardware and software