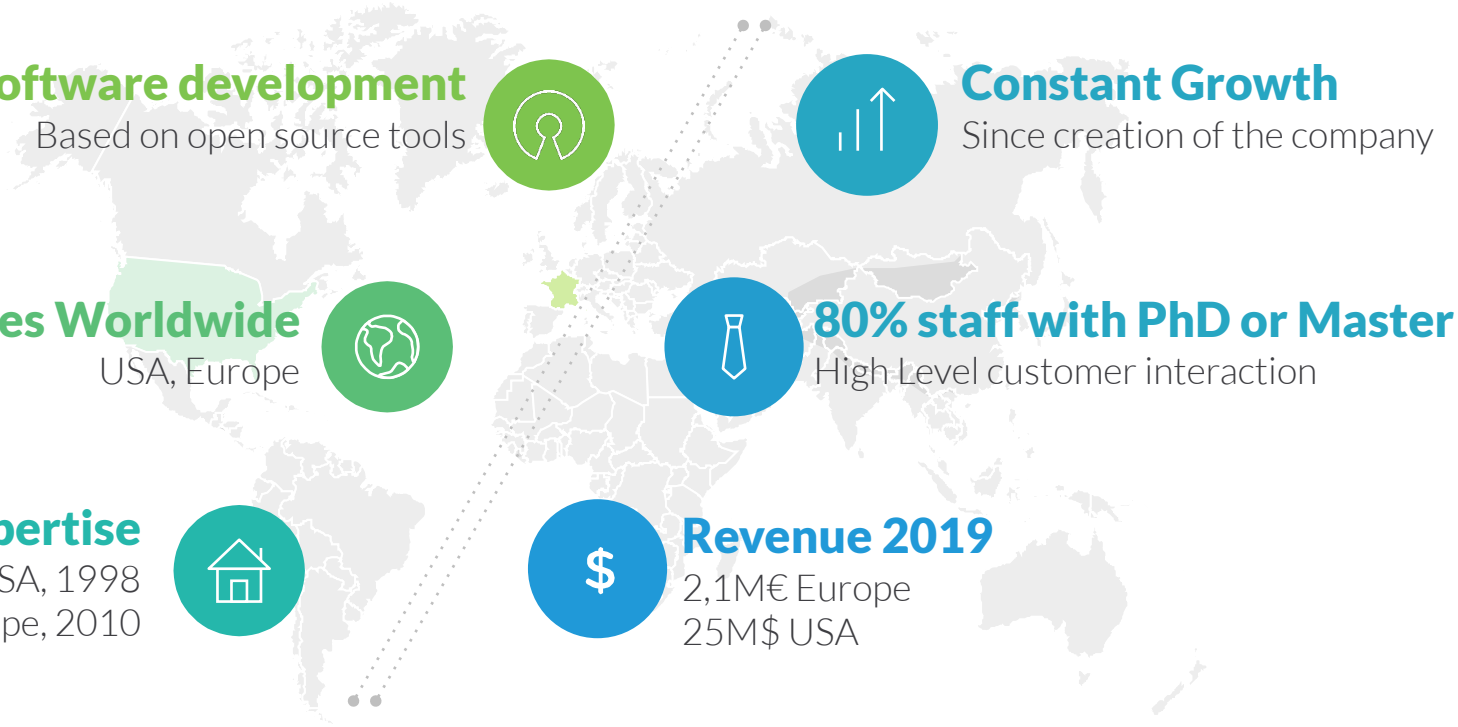
Optimized Rendering Techniques

Harness the power of ParaView

Anthony Geay, Charles Gueunet, Joachim Pouderoux, Thomas Caissard, Nicolas Vuaille, Mathieu Westphal

Kitware / Leader in scientific open source solutions



Software development
Based on open source tools

Constant Growth
Since creation of the company

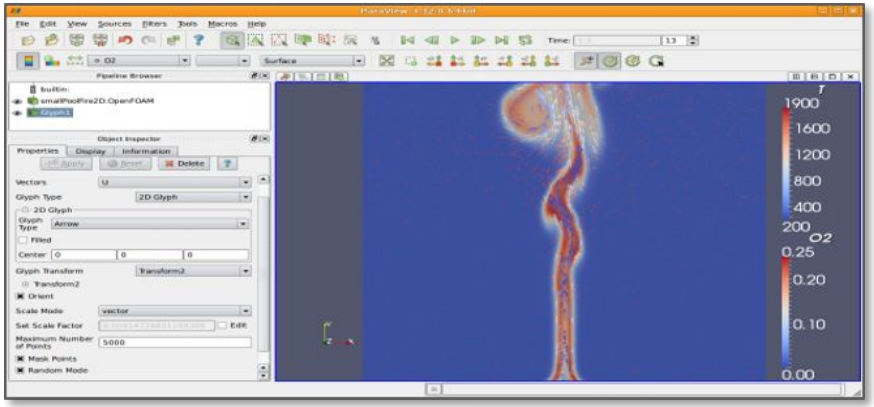
200 employees Worldwide
USA, Europe

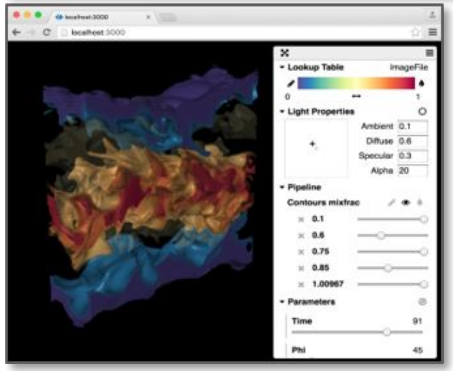
80% staff with PhD or Master
High Level customer interaction

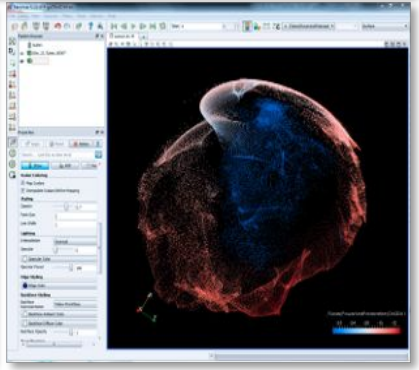
20 years of expertise
Kitware Inc USA, 1998
Kitware SAS Europe, 2010

Revenue 2019
2,1M€ Europe
25M$ USA

ParaView Scientific visualization software







FEATURES

Tackle and answer customer's technical challenges with

- Massive data visualization
- Highly customizable framework
- Community of +10 thousands users
- Cross-platform
- +1.35 million lines of code
- BSD License

WHAT FOR

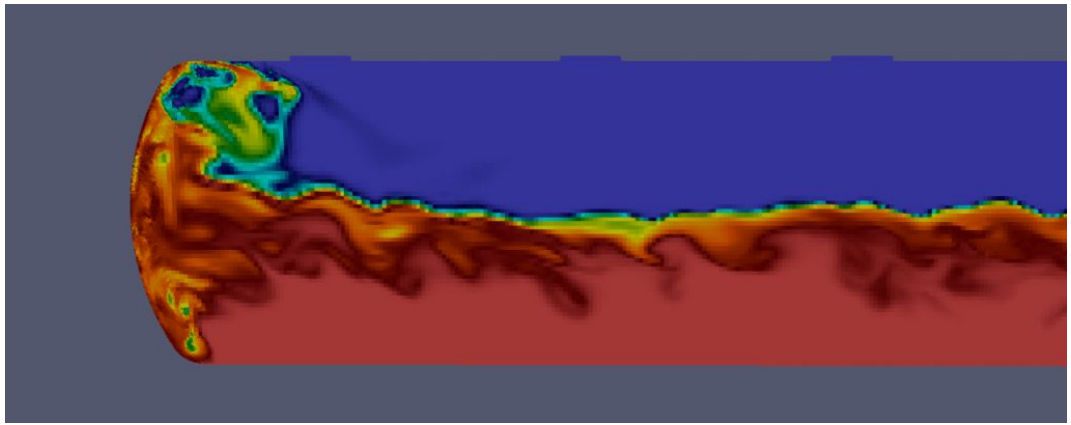
- HPC
- Visualization
- VR

Time to render

- ParaView and VTK are very generic tools for scientific visualization.
- There is no shortcut for the general case.
  - surfaces must be extracted.
  - computation must be done.
- What about specific cases ?
- ParaView flexible architecture lets user and developers create their own optimized rendering techniques in dedicated plugins for specific cases.
  - NvidiaIndex plugin for volume rendering
  - StreamLinesRepresentation for streamlines

StaticMeshPlugin : The problem

- In many domains, simulation data is temporal
- In some cases, it has variable data with static geometries
- ParaView still needs to read and process the geometry for each time step !
- What if ParaView could know if the geometry changed or not ?



a temporal dataset with a static geometry from Code_Saturne

StaticMeshPlugin : VTK Data Model

How is temporal unstructured data stored in VTK / ParaView ?

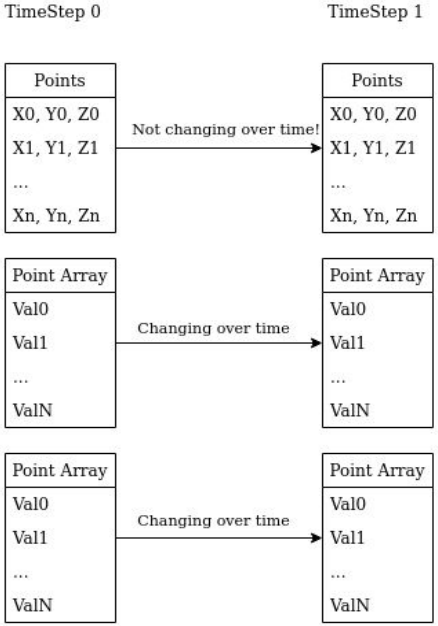
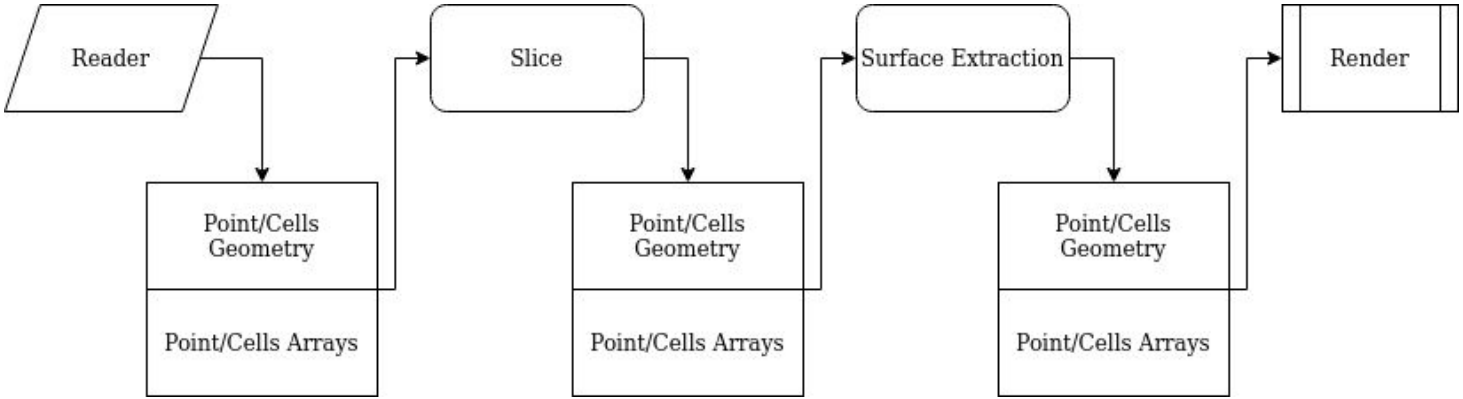


Illustration of point data changing over time

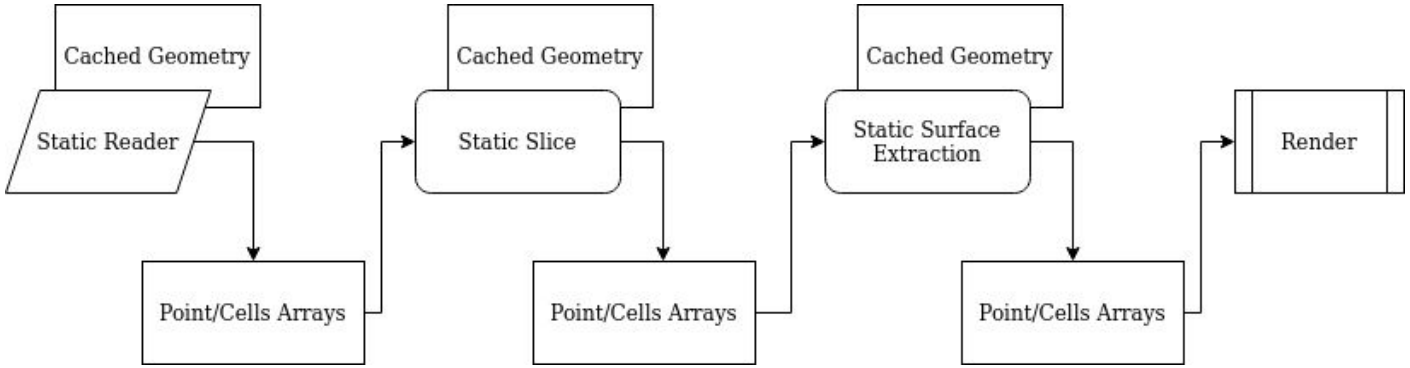
- Points/Cells info and arrays are stored in similar ways
- The user knows that the geometry is not changing
- Can the software know ?

StaticMeshPlugin : Standard VTK Pipeline



The geometry will be read and transfer at each timesteps, alongside the point and cells arrays.

StaticMeshPlugin : A solution - “Static”

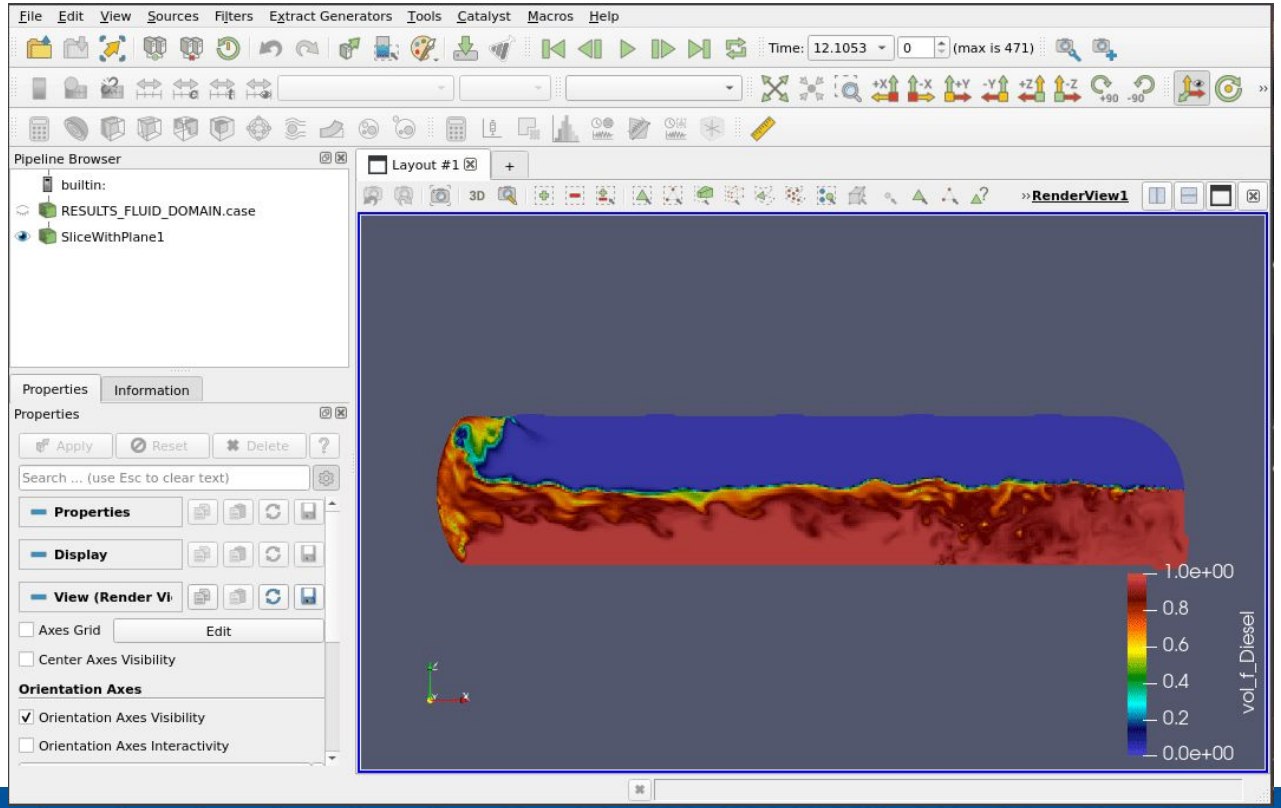


Once the geometry has been read and processed **once,** it will never be read or processed again, only the data arrays will be read and processed.

The geometry is cached within the filters.

The cache can be invalidated if a change is detected on the geometry.

StaticMeshPlugin: How does it look ?

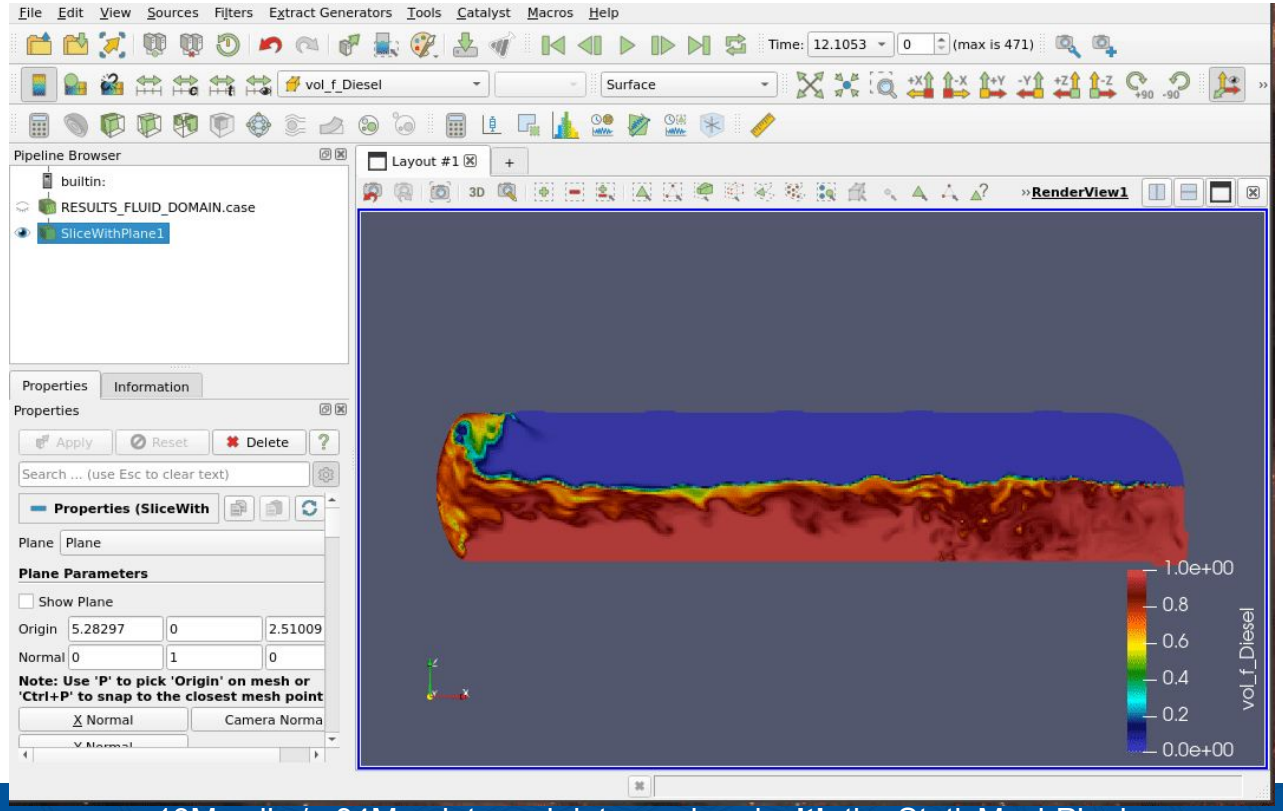


Without the static mesh plugin, one time step in the example takes roughly 5s to render.

This is not interactive.

~13M cells / ~34M points real data rendered without the StaticMeshPlugin

StaticMeshPlugin: How does it look ?



With the static mesh plugin, one time step in the example takes roughly <1s to render.

This is interactive.

~13M cells / ~34M points real data rendered **with** the StaticMeshPlugin

StaticMeshPlugin: Limitations / Information

- Requires to compile ParaView
- Reader implementation is complex
  - EnSight reader in the plugin
  - MedReader in Salome
- Each filter should be reimplemented to support this optimization
- Non-Static filter “breaks” the static pipeline

- Blog for benchmarks and implementation details: https://blog.kitware.com/staticmeshplugin/

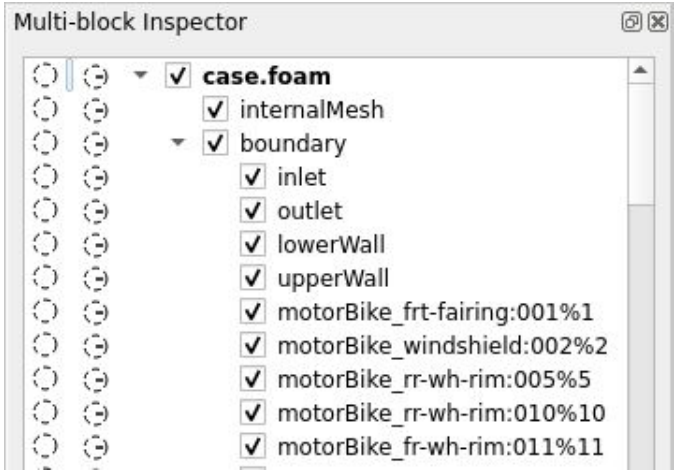
FastSurfaceMultiblock: The problem

- Certains volumic file format provides already computed surface information
  - OpenFOAM
  - Other proprietary format
- ParaView always want to compute the surface of a volume for rendering
- Could we directly use the surface information ?

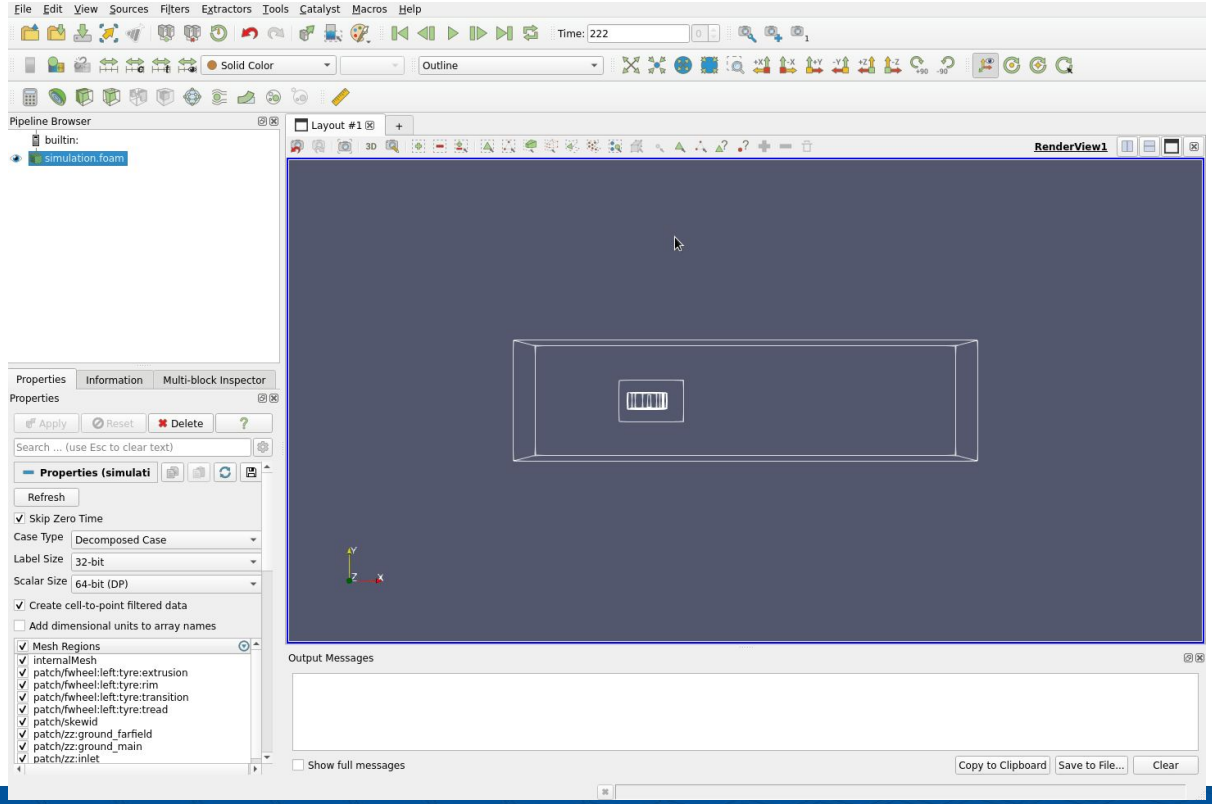
The OpenFoam Motorbike, rendered with the Intel OSPRay pathtracer in ParaView.

FastSurfaceMultiblock: A solution

- In OpenFOAM format, surfaces are clearly identified
- New representations could be added to use this “block” as the surface output instead of computing the surface of the internal mesh
- No computation needed !



FastSurfaceMultiblock: How does it look ?



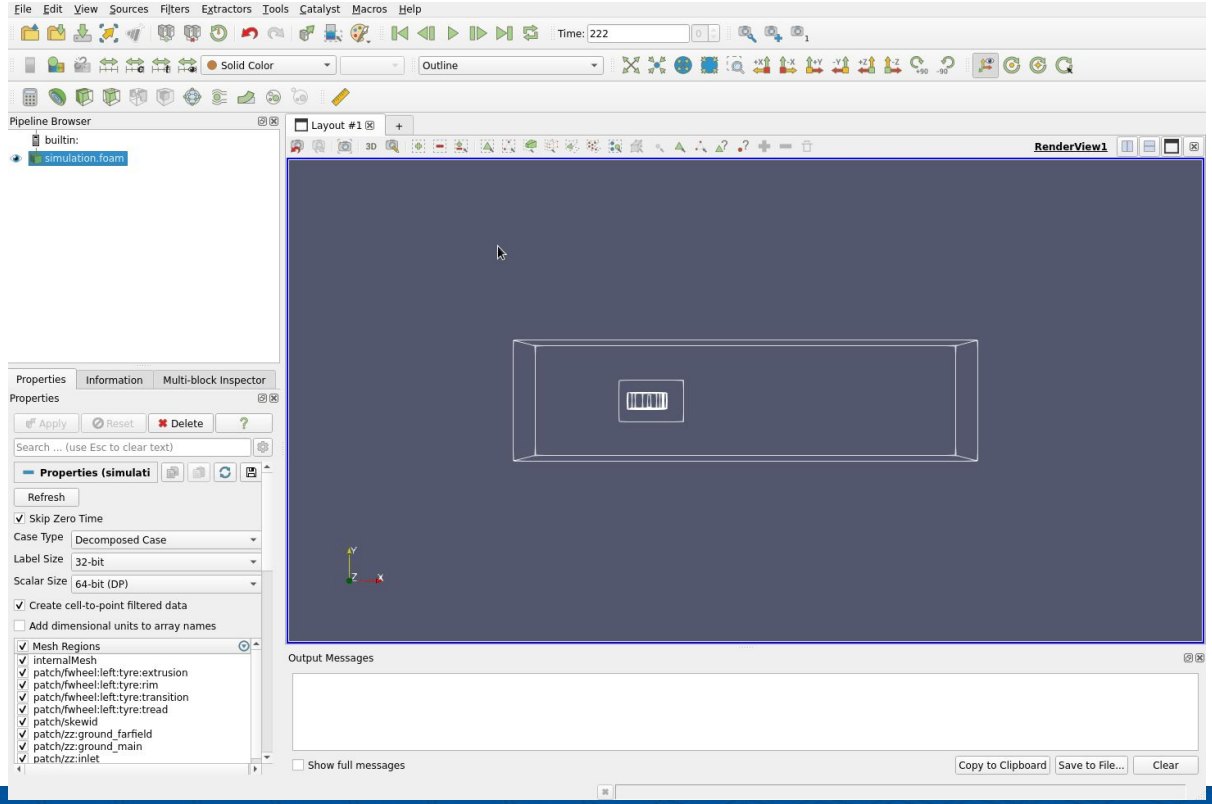
Without the plugin, switching to surface representation takes roughly 10s to render.

This is not interactive.

~20M cells real data rendered without the FastSurfaceMutliblockRepresentation plugin

Kitware

FastSurfaceMultiblock: How does it look ?



Using the plugin, switching to fast surface representation is instantaneous!

This is interactive.

~20M cells real data rendered **with** the FastSurfaceMutliblockRepresentation plugin

Kitware

FastSurfaceMultiblock: Limitations / Information

- Requires to compile ParaView
- Dependent of the structure of the data
- Require to set specific settings in ParaView

- Blog for benchmarks and implementation details: https://blog.kitware.com/fastsurfacemultiblockrepresentation-plugin-for-paraview/

Questions ?

Supported/Initiated by: