Visualizing your Data

Joseph Insley
Lead, Visualization & Data Analysis
Argonne Leadership Computing Facility

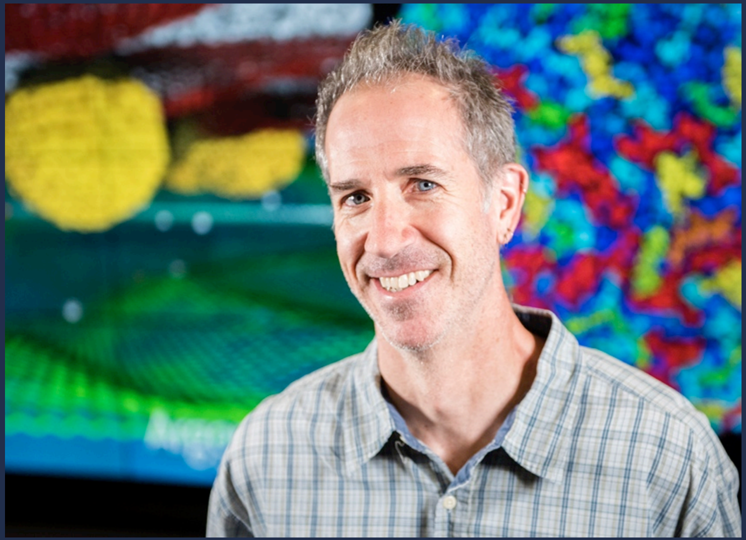
Silvio Rizzi
Computer Scientist
Argonne Leadership Computing Facility

Janet Knowles
Principal Software Engineering Specialist
Argonne Leadership Computing Facility

Victor Mateevitsi
Computer Scientist
Argonne Leadership Computing Facility

ALCF Visualization Group

Silvio Rizzi



Joe Insley

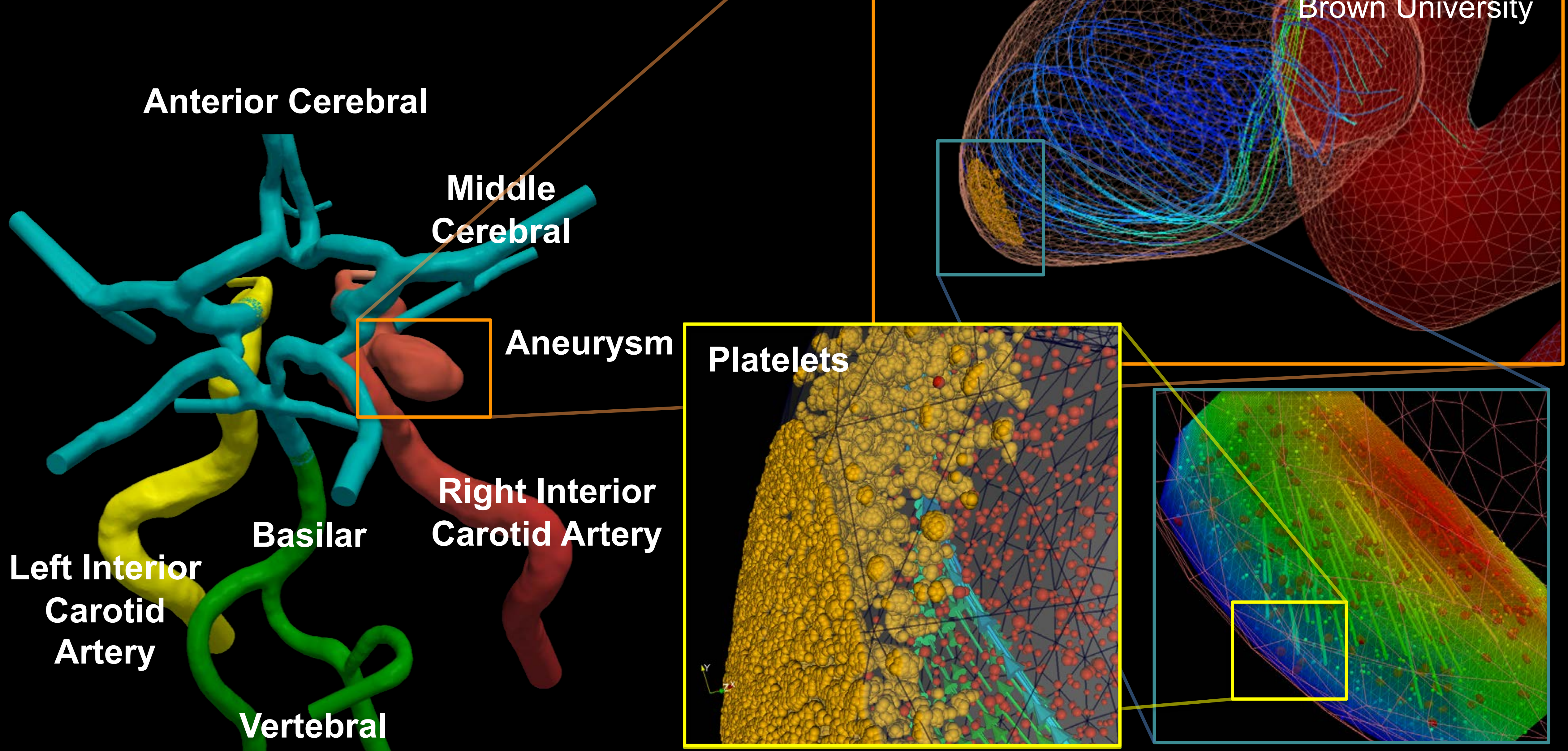
Janet Knowles

Victor Mateevitsi

Here's the plan…

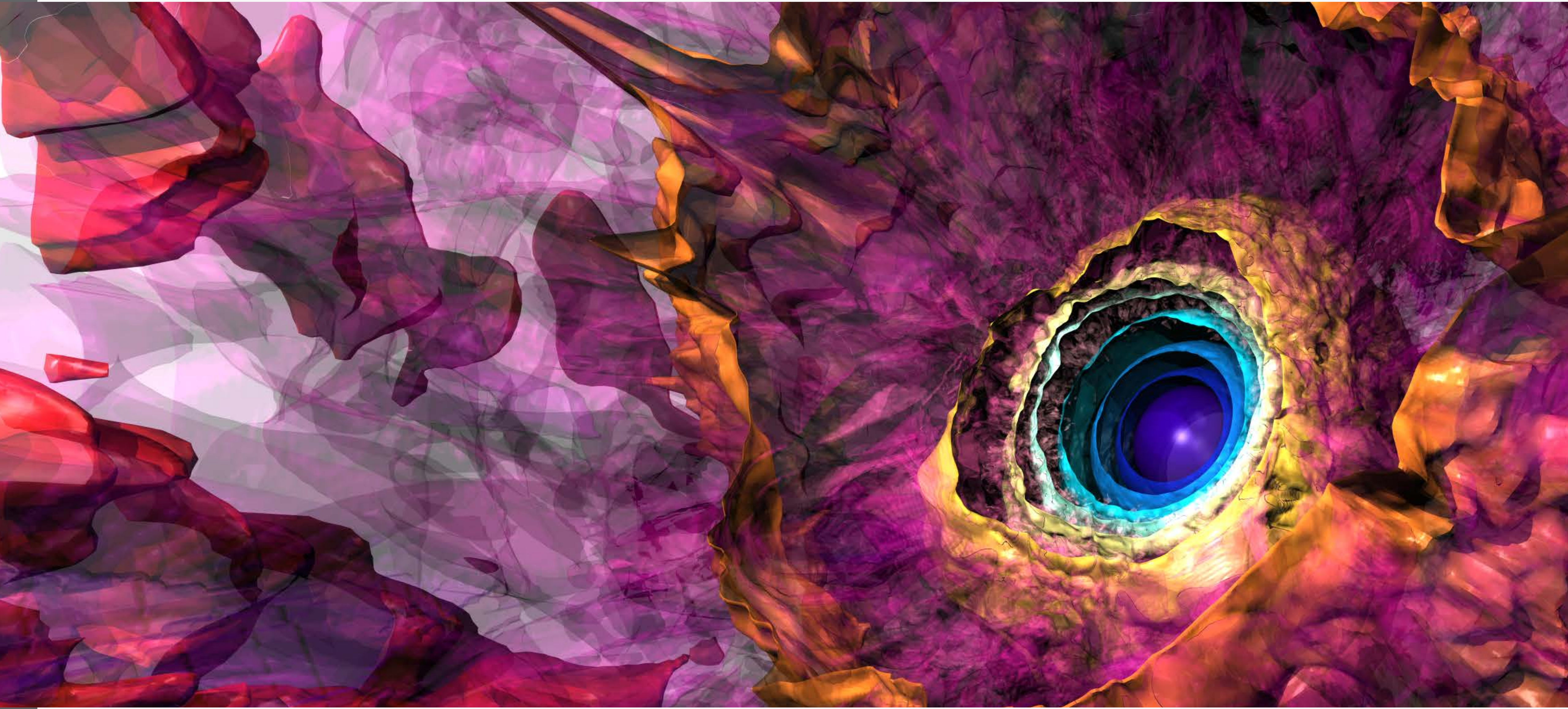
- Examples of visualizations
- Visualization resources
- Visualization tools and formats
- Data representations
- Visualization for debugging
- Simple ParaView scripting example
- In Situ Visualization and Analysis

Multi-Scale Simulation / Visualization
Arterial Blood Flow

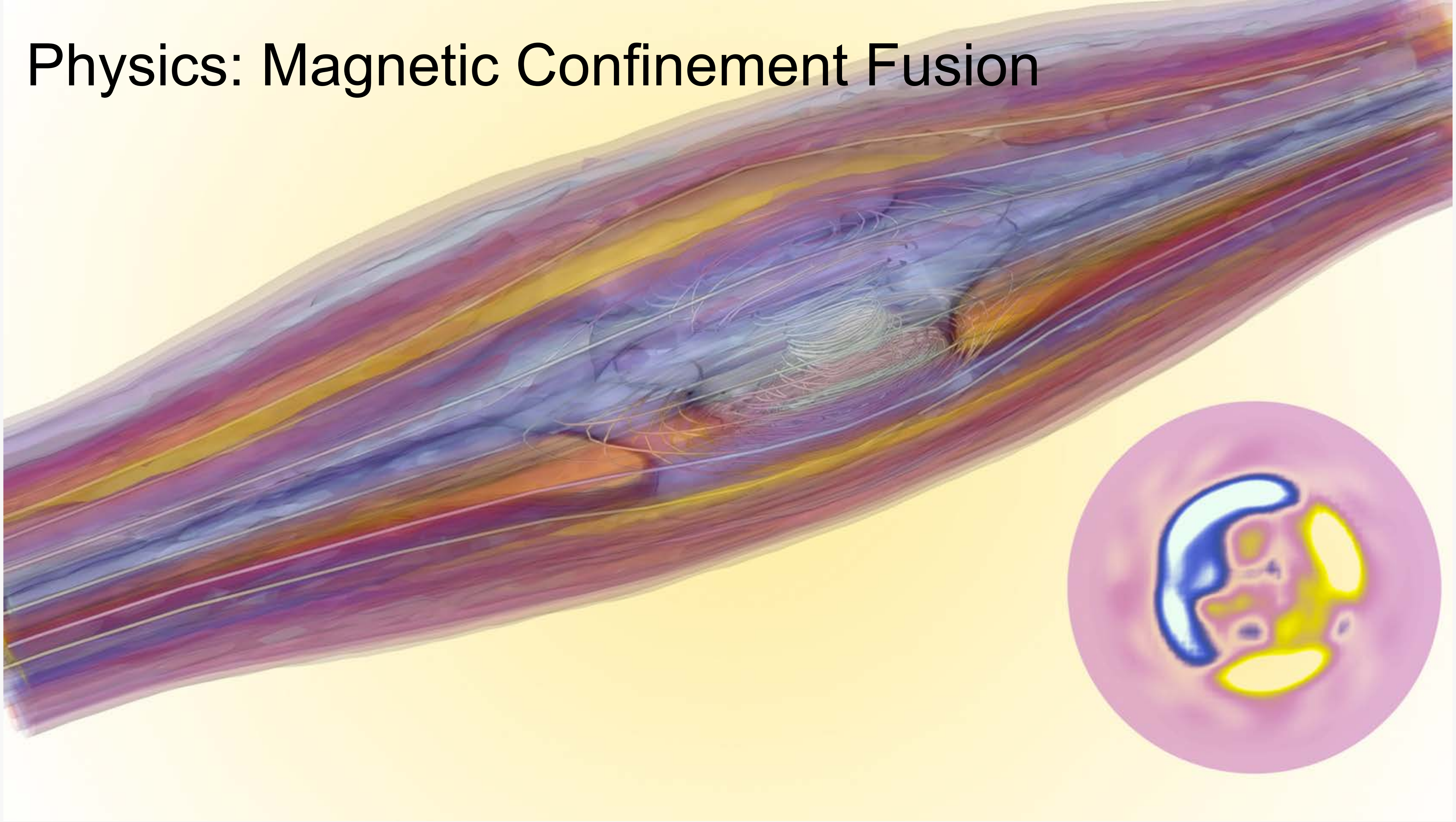


Physics: Stellar Radiation

Data courtesy of: Lars Bildsten and Yan-Fei Jiang,
University of California at Santa Barbara

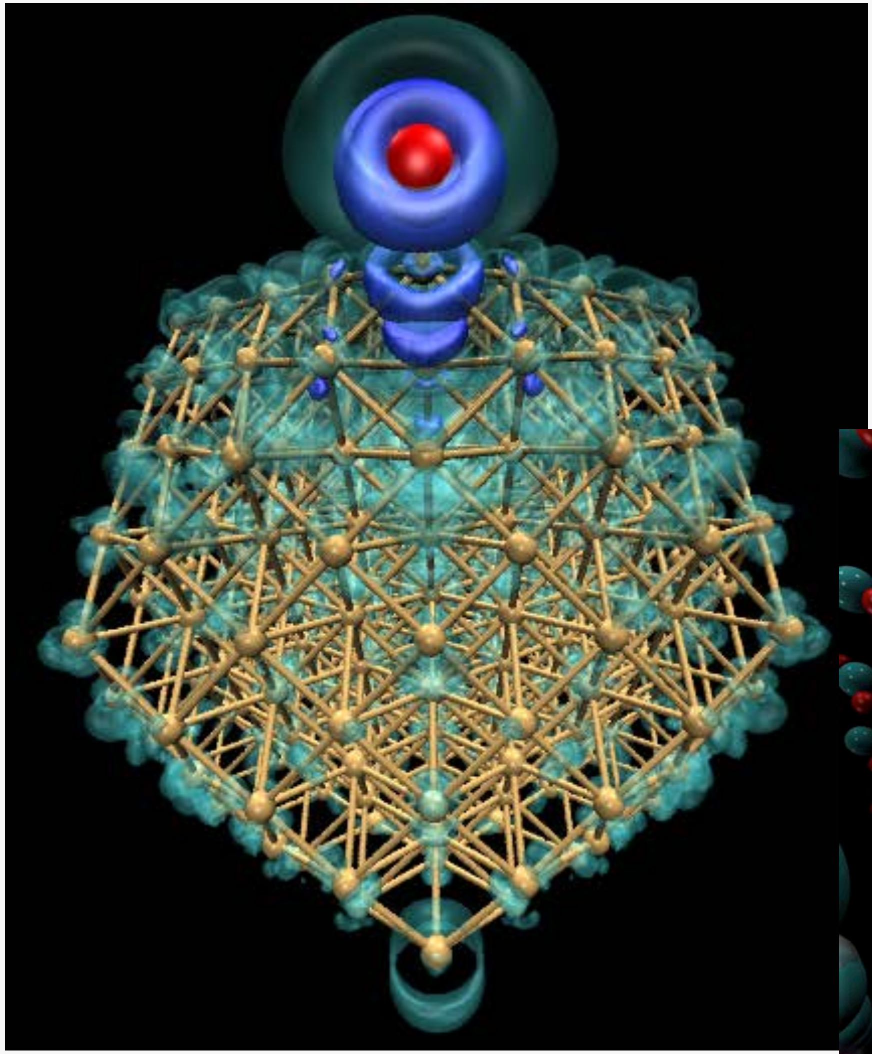


Physics: Magnetic Confinement Fusion

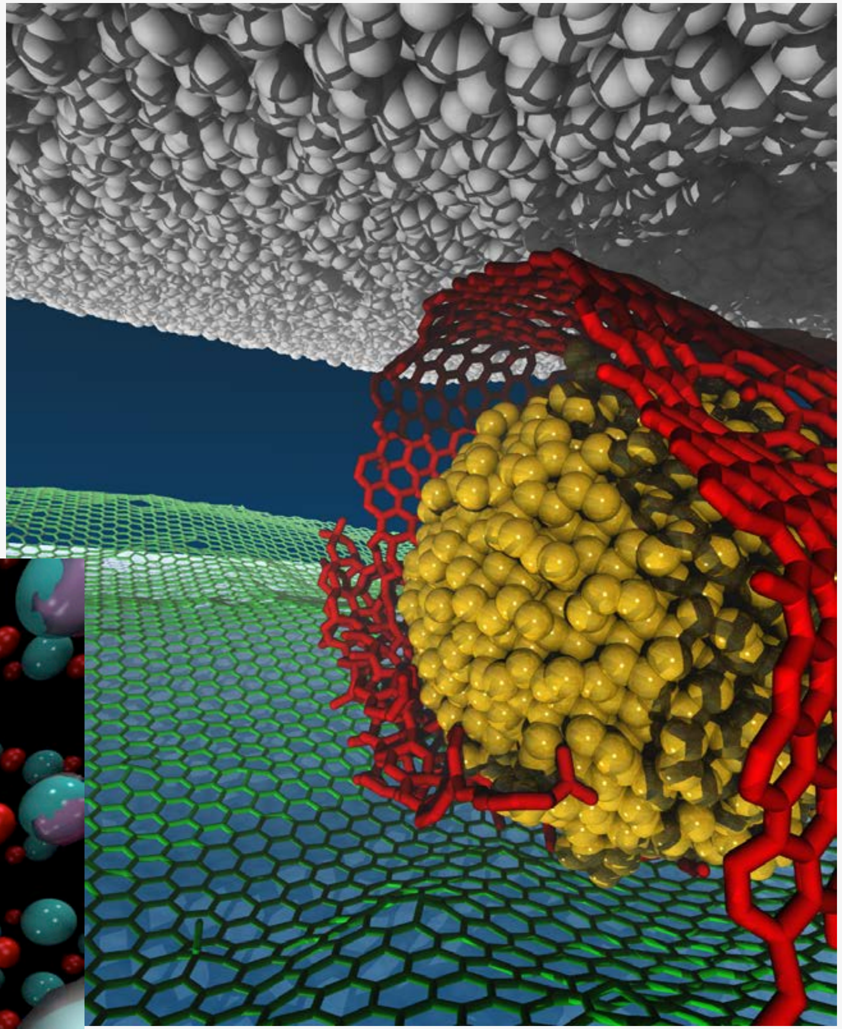


Data courtesy of Sean Dettrick, TAE Technologies, Inc.

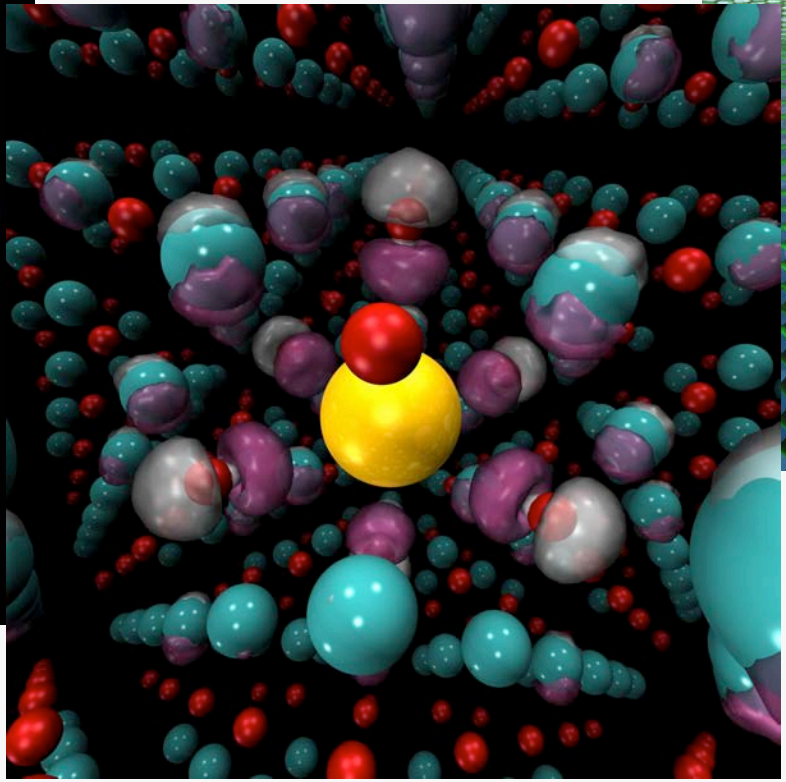
Materials Science / Molecular



Data courtesy of: Jeff Greeley, Nichols Romero, Argonne National Laboratory

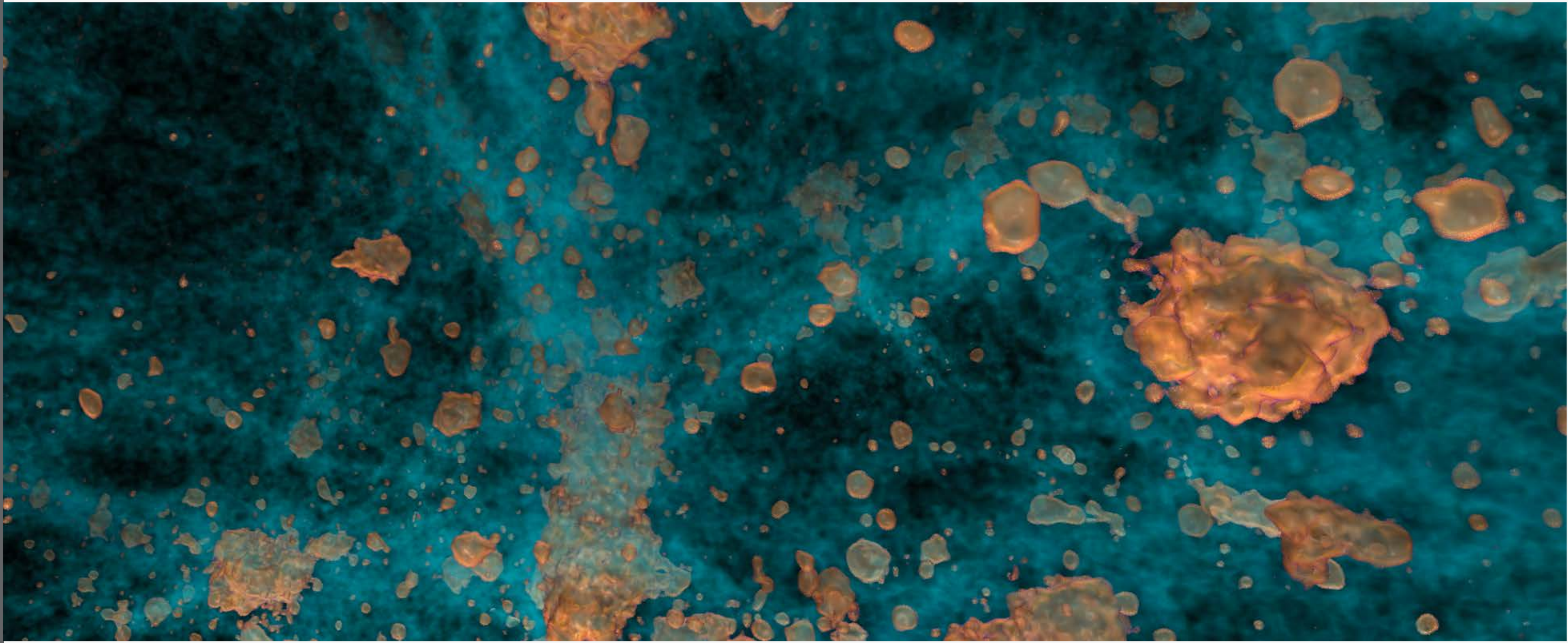


Data courtesy of: Subramanian Sankaranarayanan, Argonne National Laboratory



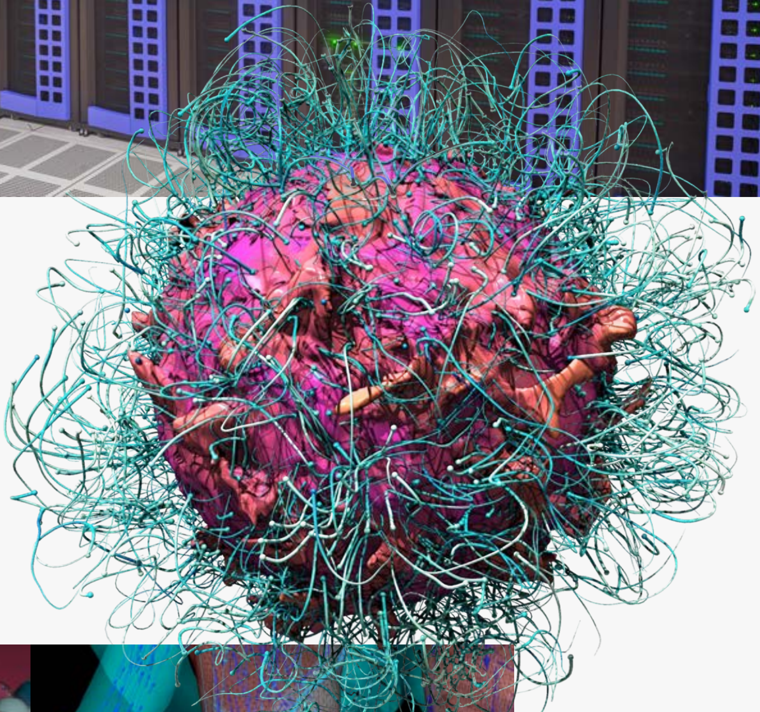
Data courtesy of: Paul Kent, Oak Ridge National Laboratory, Anouar Benali, Argonne National Laboratory

Cosmology



Data courtesy of: Salman Habib, Katrin Heitmann, and the HACC team, Argonne National Laboratory

Cooley: Analytics/Visualization cluster



Peak 223 TF

126 nodes; each node has

– Two Intel Xeon E5-2620 Haswell 2.4 GHz 6-core processors

– NVIDIA Telsa K80 graphics processing unit (24GB)

– 384 GB of RAM

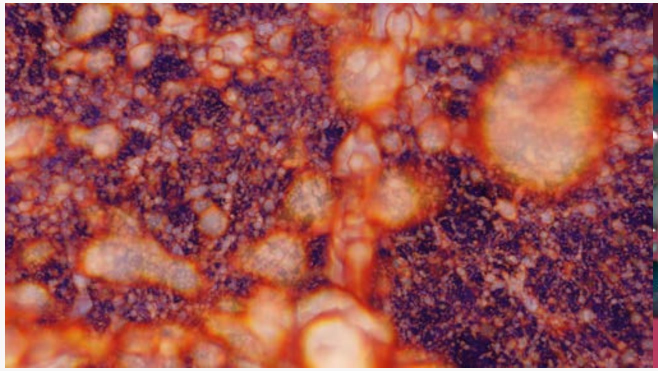
Aggregate RAM of 47 TB

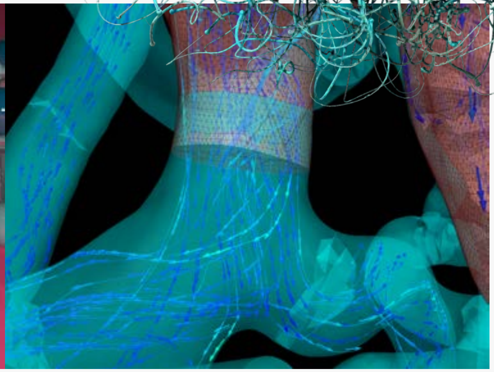
Aggregate GPU memory of ~3TB

Cray CS System

216 port FDR IB switch with uplinks to our QDR infrastructure

Mounts the Theta, Eagle, and Grand file systems





Visualization Tools and Data Formats

All Sorts of Tools

Visualization Applications
- VisIt *
- ParaView*
- EnSight

Domain Specific
- VMD, PyMol, Ovito

APIs
- VTK*: visualization
- ITK: segmentation & registration

GPU performance
- vl3: shader-based volume and particle rendering

Analysis Environments
- Matlab
- Parallel R

Utilities
- GnuPlot
- ImageMagick*

■ Available on Cooley

* Available on Theta

ParaView & VisIt vs. vtk

ParaView & VisIt

- General purpose visualization applications
- GUI-based
- Client / Server model to support remote visualization
- Scriptable / Extendable
- Built on top of vtk (largely)
- *In situ* capabilities

vtk

- Programming environment / API
- Additional capabilities, finer control
- Smaller memory footprint
- Requires more expertise (build custom applications)

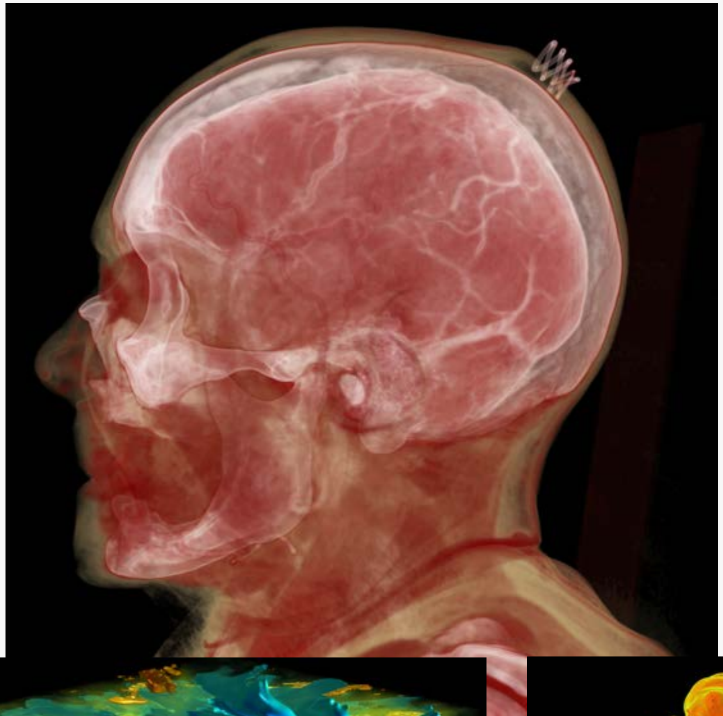
Data File Formats (ParaView & VisIt)

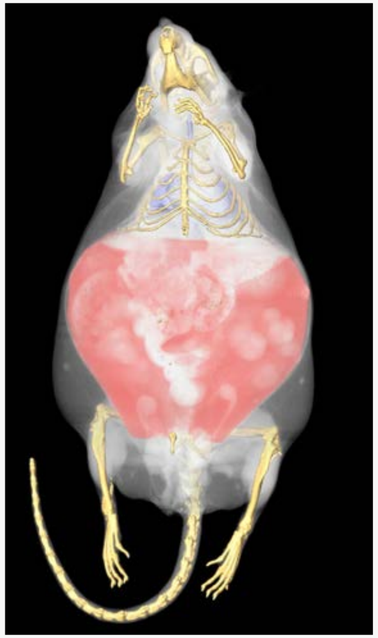
- VTK
- Parallel (partitioned) VTK
- VTK MultiBlock (MultiGroup, Hierarchical, Hierarchical Box)
- Legacy VTK
- Parallel (partitioned) legacy VTK
- EnSight files
- EnSight Master Server
- Exodus
- BYU
- XDMF
- PLOT2D
- PLOT3D
- SpyPlot CTH
- HDF5 raw image data
- DEM
- VRML
- PLY
- Polygonal Protein Data Bank
- XMol Molecule
- Stereo Lithography
- Gaussian Cube
- Raw (binary)
- AVS
- Meta Image
- Facet
- PNG
- SAF
- LS-Dyna
- Nek5000
- OVERFLOW
- paraDIS
- PATRAN
- PFLOTRAN
- Pixie
- PuReMD
- S3D
- SAS
- Tetrad
- UNIC
- VASP
- ZeusMP
- ANALYZE
- BOV
- GMV
- Tecplot
- Vis5D
- Xmdv
- XSF

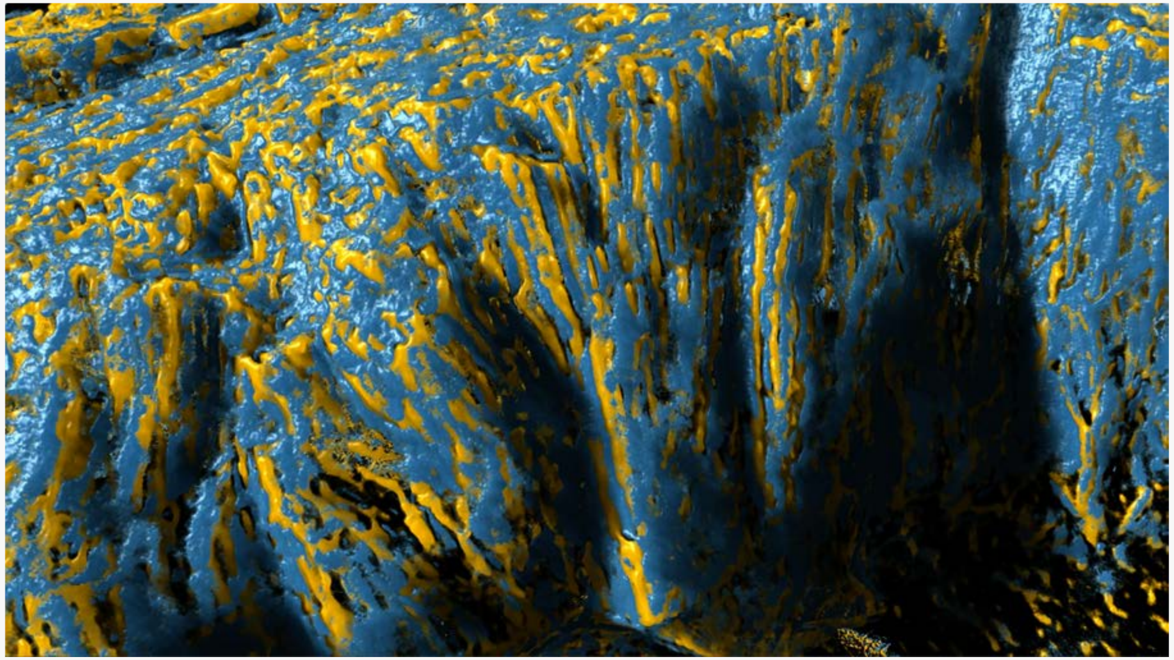
Data Representations

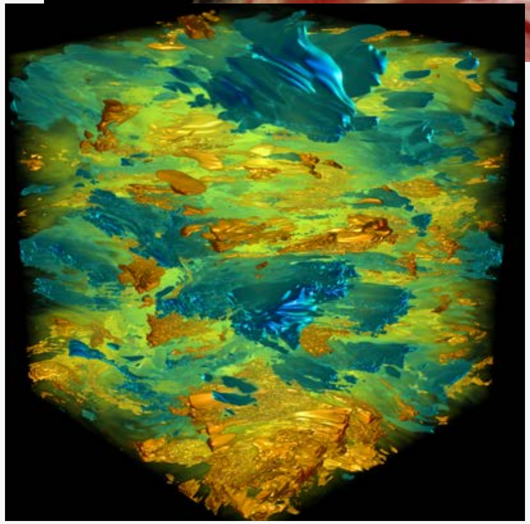
Argonne NATIONAL LABORATORY

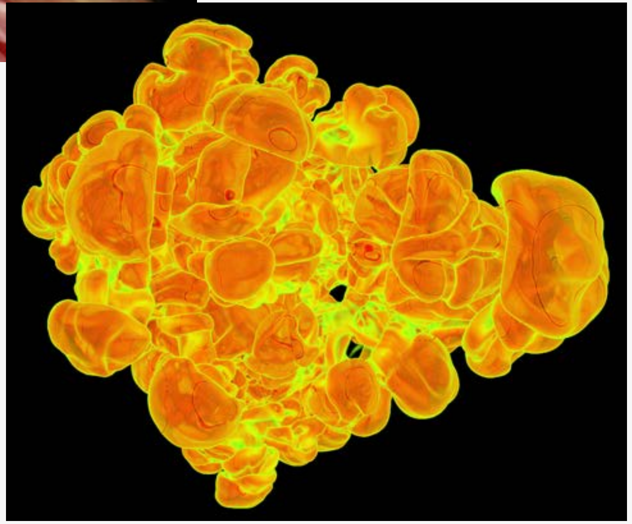
Data Representations: Volume Rendering

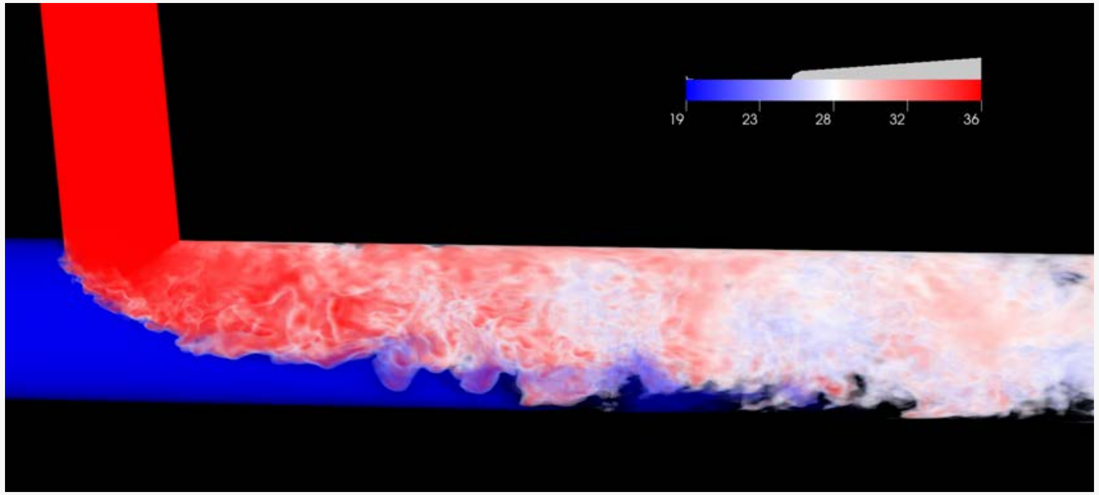






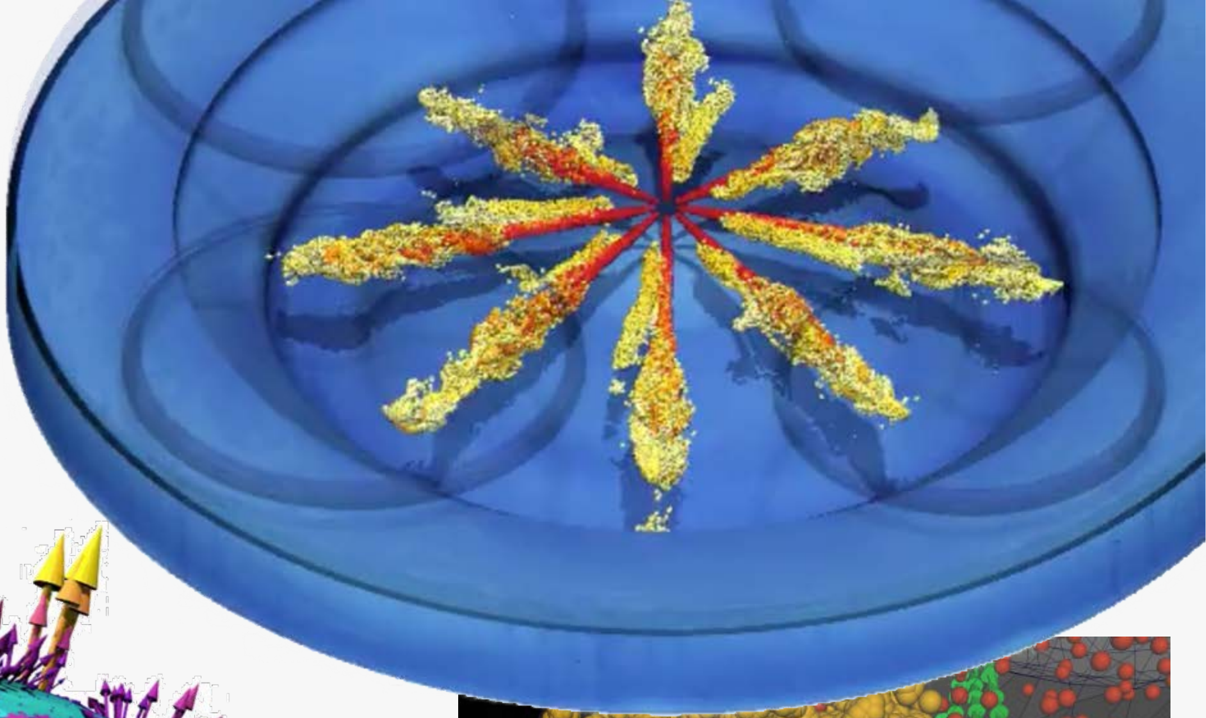






Argonne NATIONAL LABORATORY

Data Representations: Glyphs



2D or 3D geometric object to represent point data

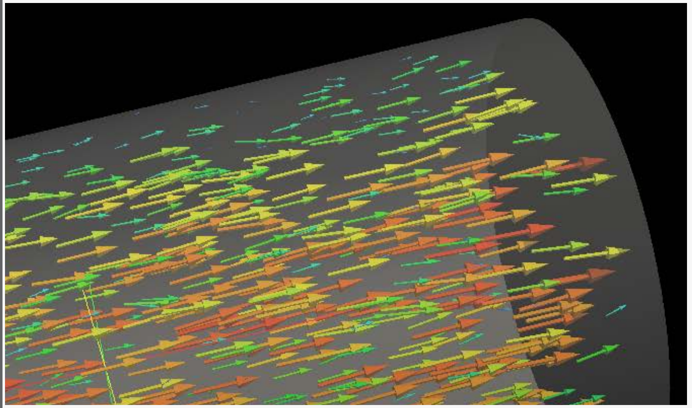
Location dictated by coordinate

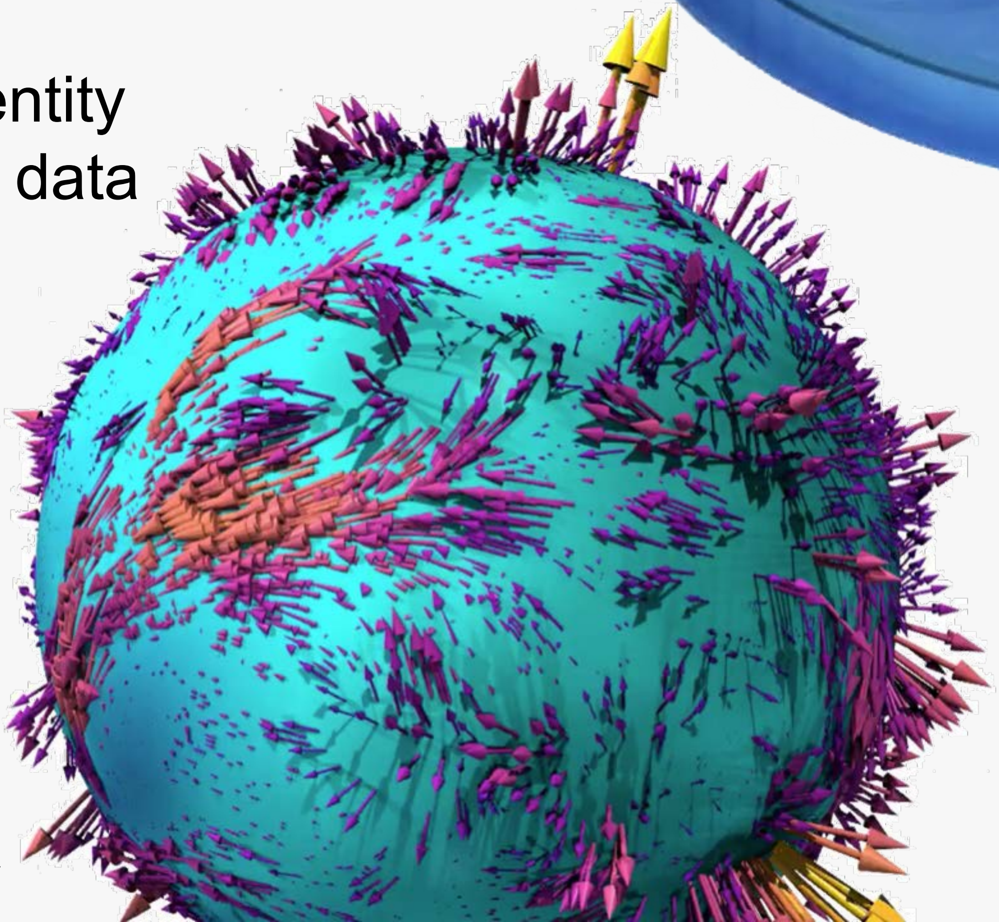
– 3D location on mesh

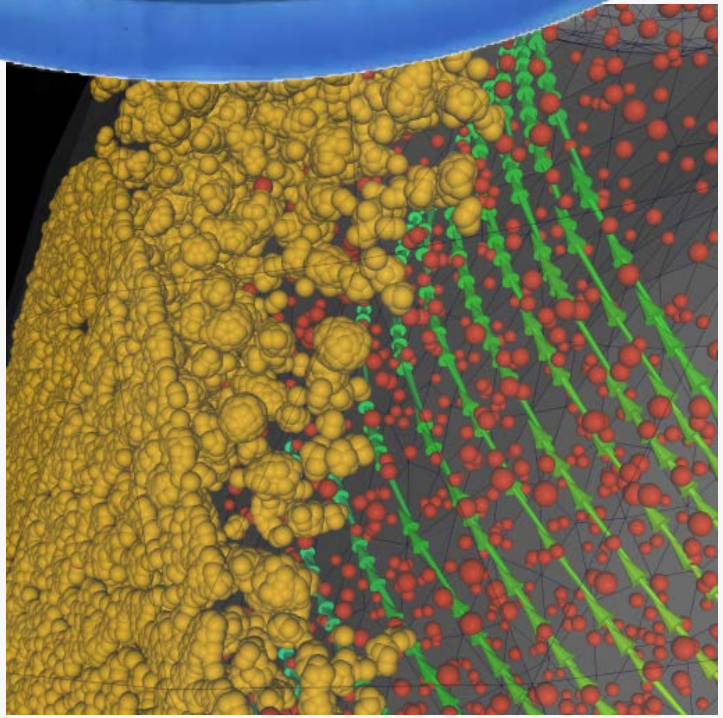
– 2D position in table/graph

Attributes of graphical entity dictated by attributes of data

– color, size, orientation







Data Representations: Contours (Isosurfaces)

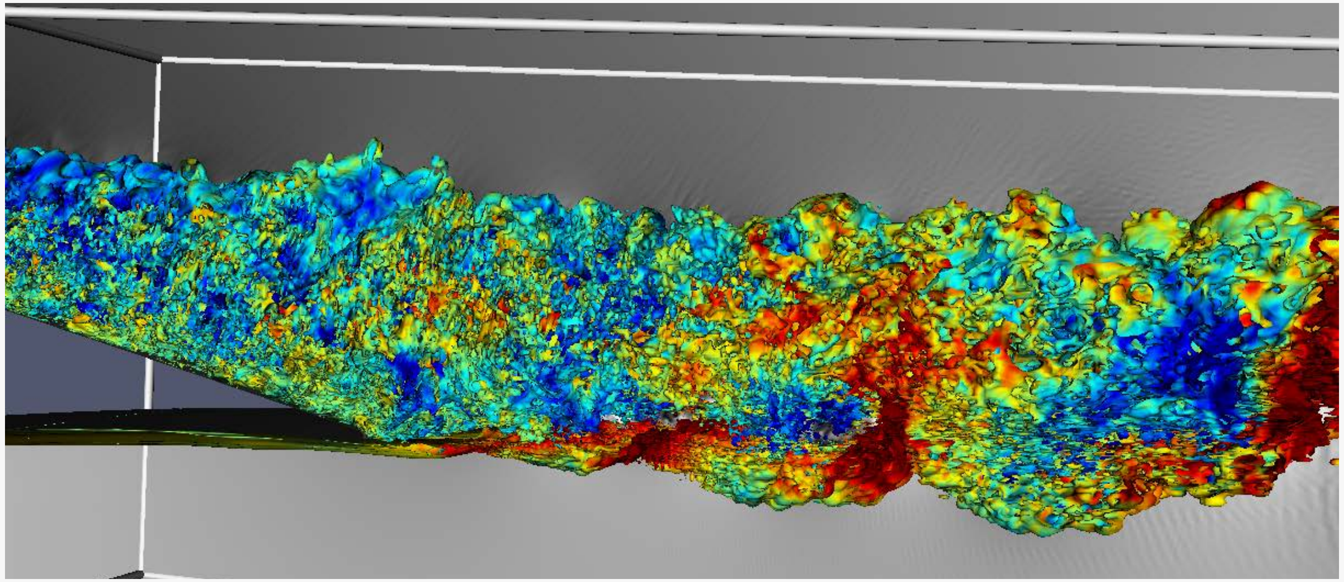
A Line (2D) or Surface (3D), representing a constant value

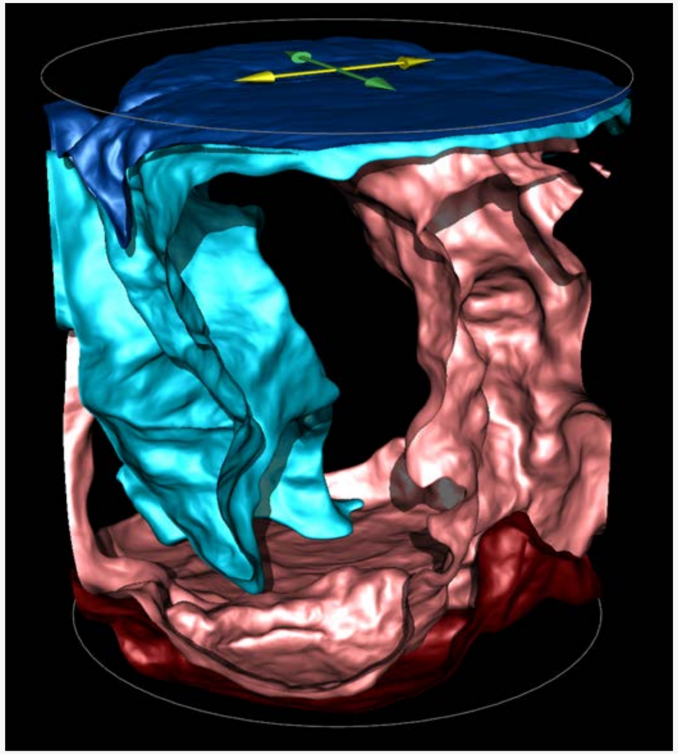
VisIt & ParaView:

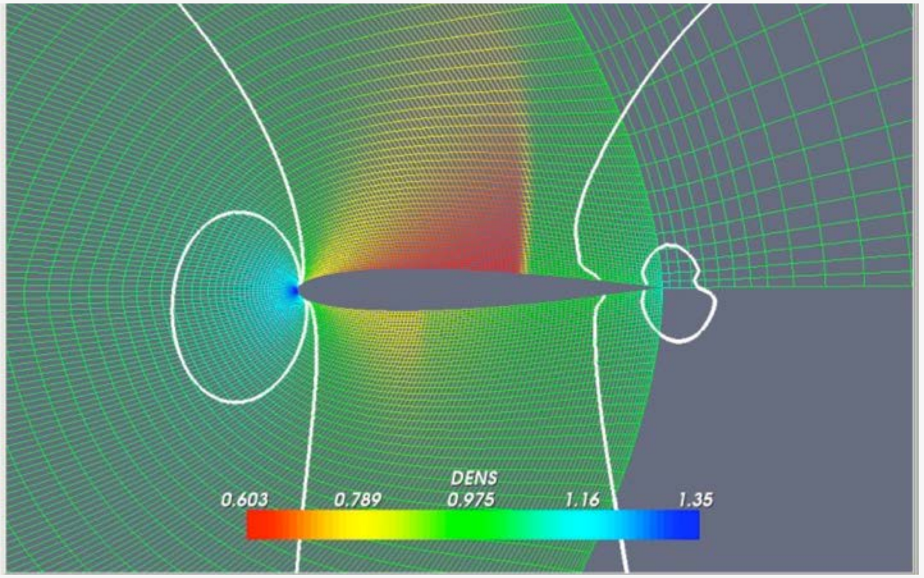
– good at this

vtk:

– same, but again requires more effort







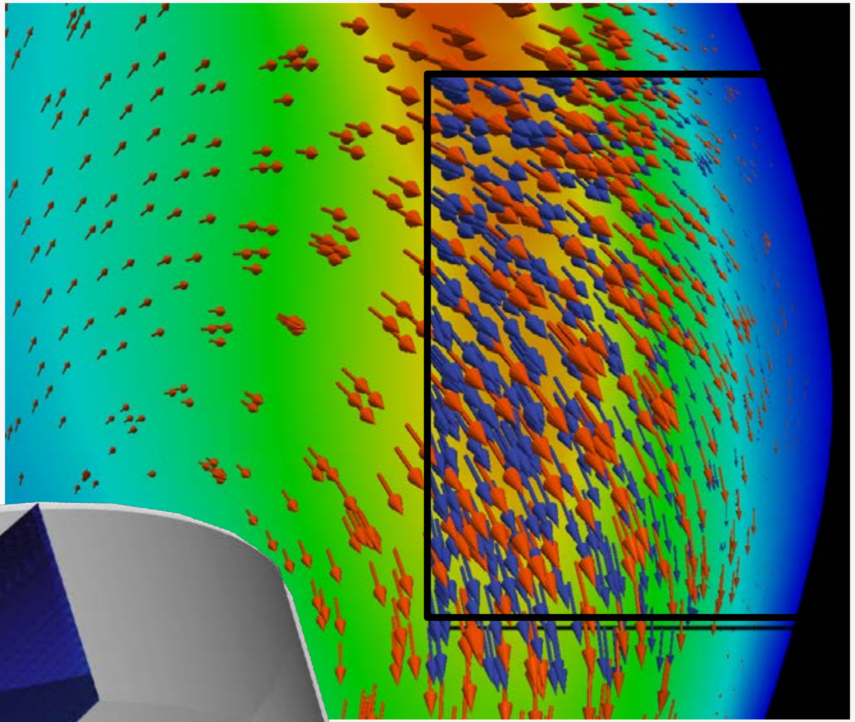
Data Representations: Cutting Planes

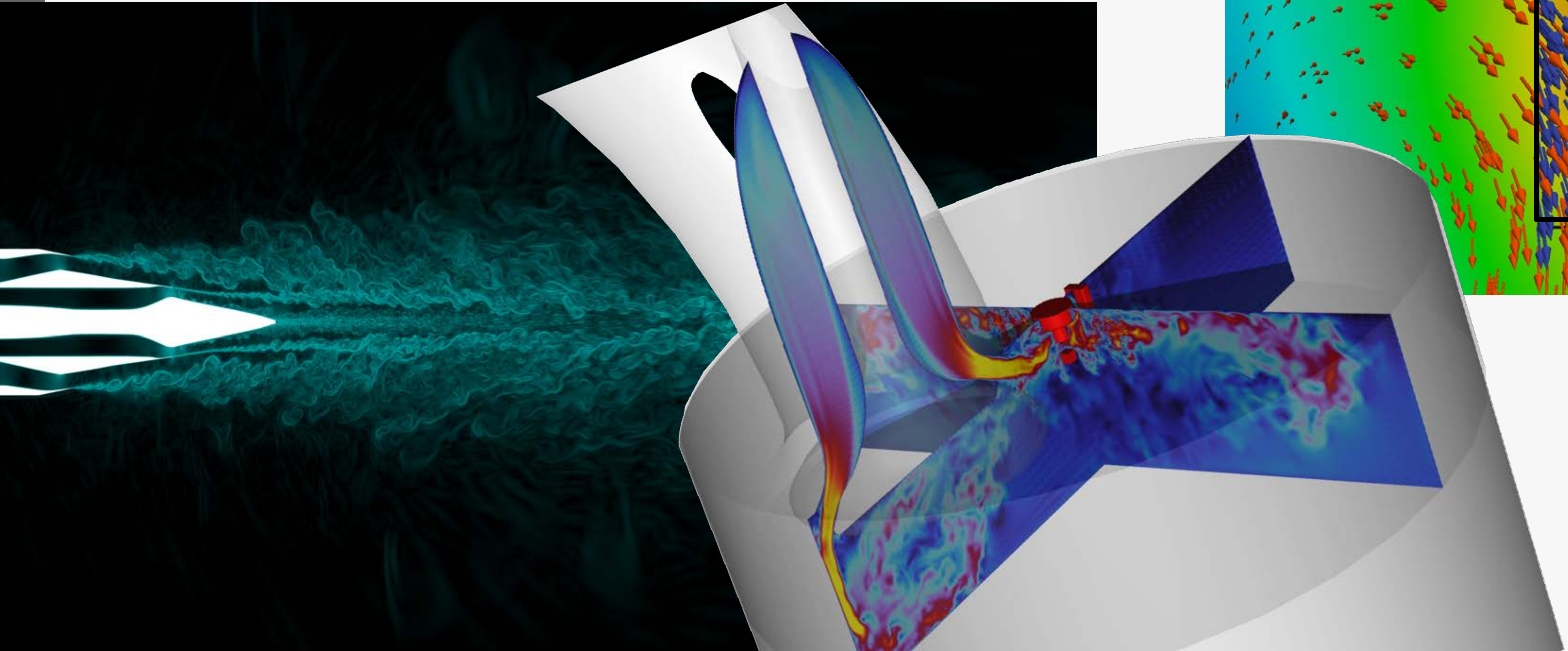
Slice a plane through the data

– Can apply additional visualization methods to resulting plane

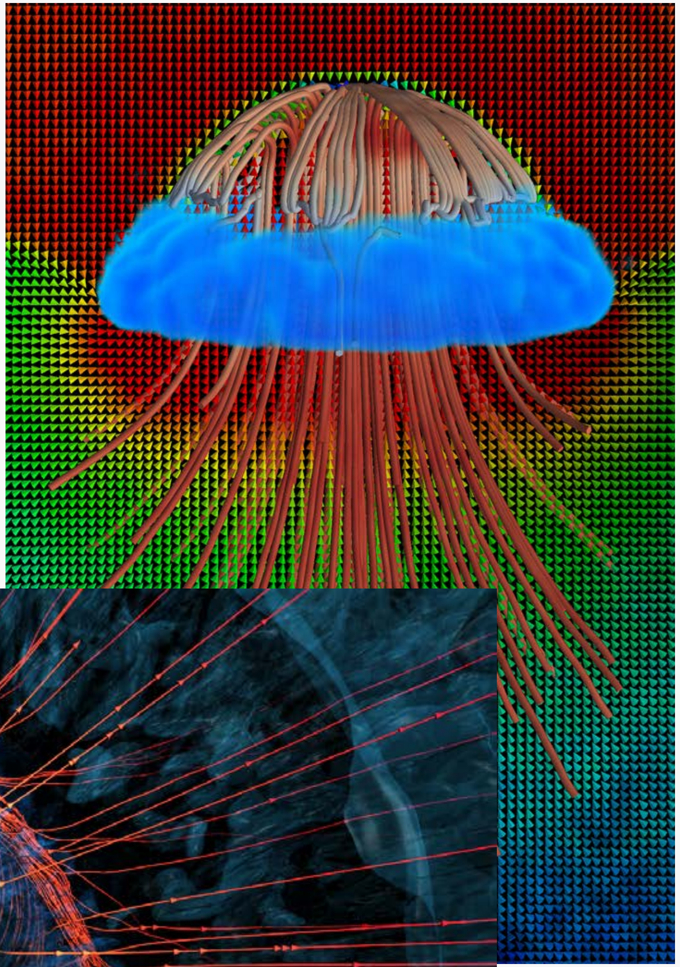
VisIt & ParaView & vtk good at this

VMD has similar capabilities for some data formats





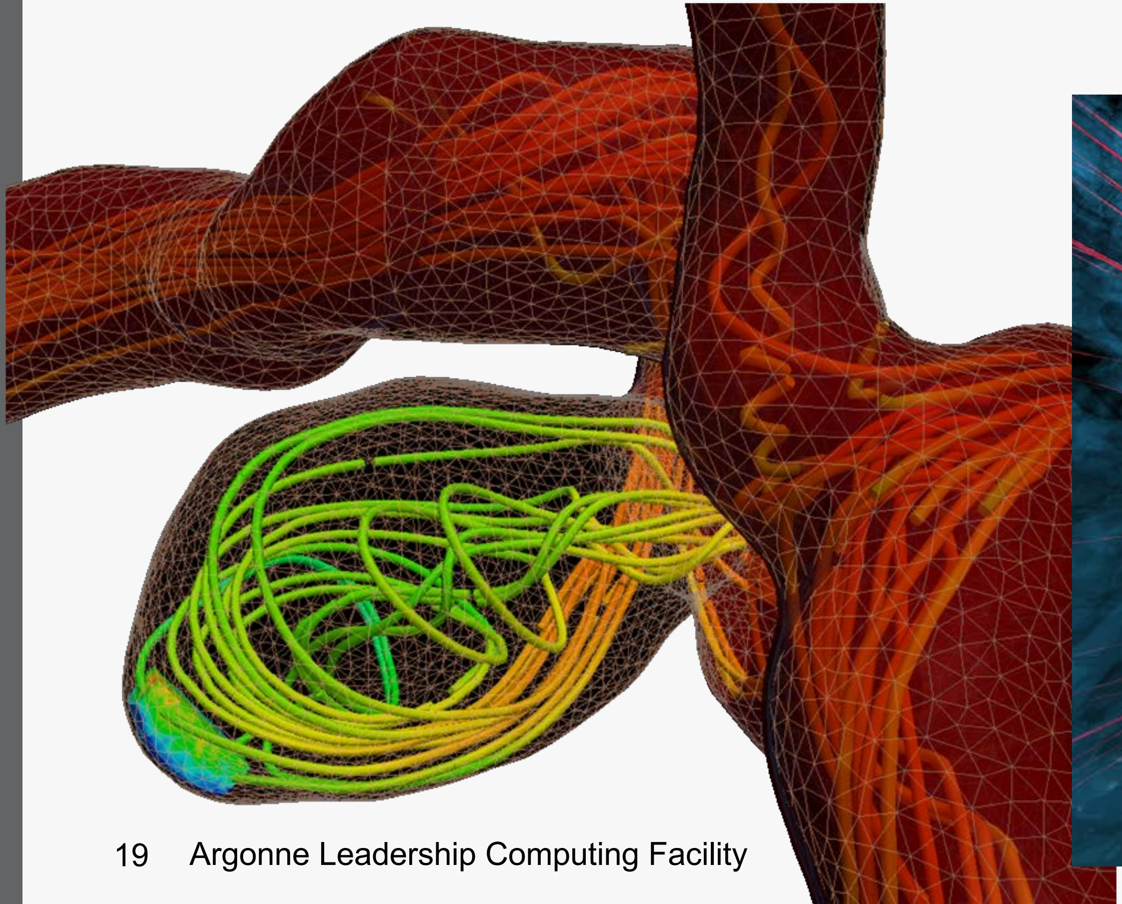
Data Representations: Streamlines

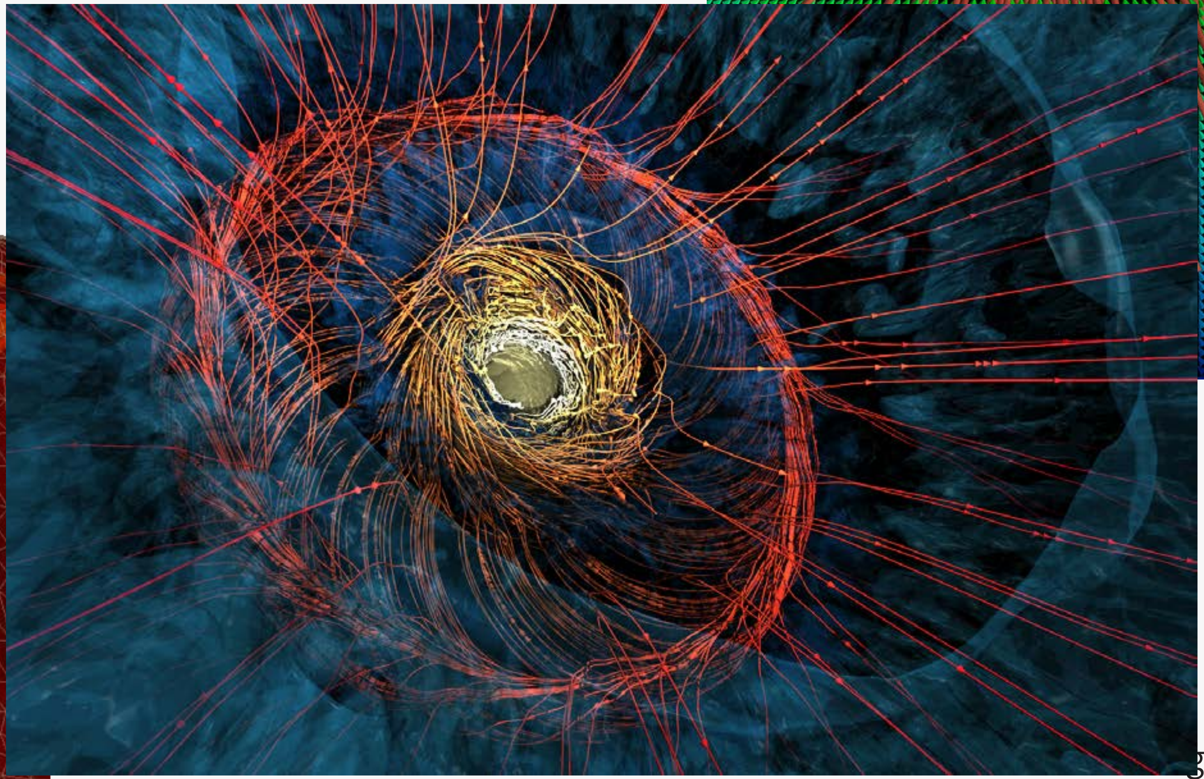


From vector field on a mesh (needs connectivity)

– Show the direction an element will travel in at any point in time.

VisIt & ParaView & vtk good at this



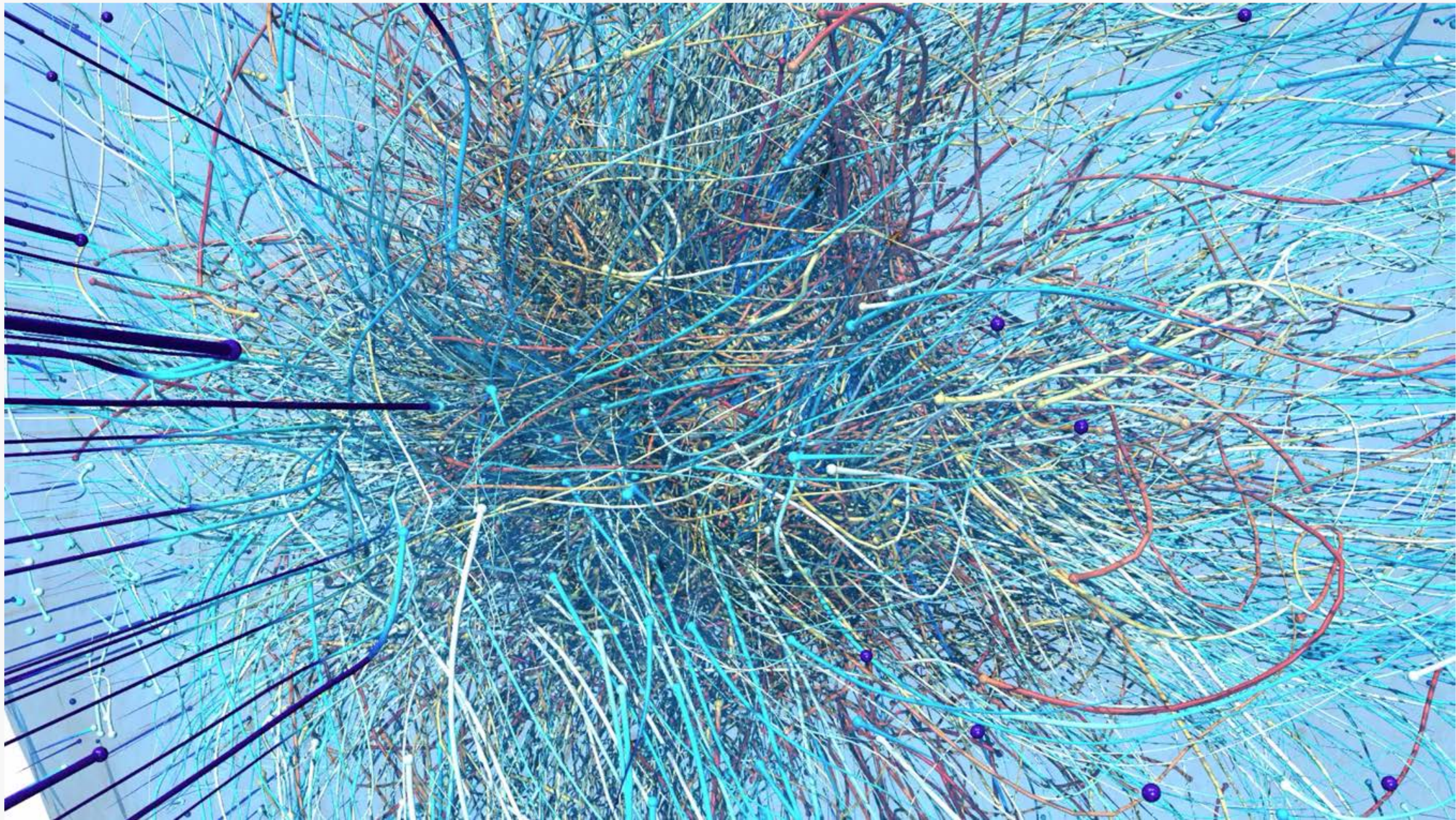


Data Representations: Pathlines

From vector field on a mesh (needs connectivity)

– Trace the path an element will travel over time.

VisIt & ParaView & vtk good at this



Molecular Dynamics Visualization

VMD:

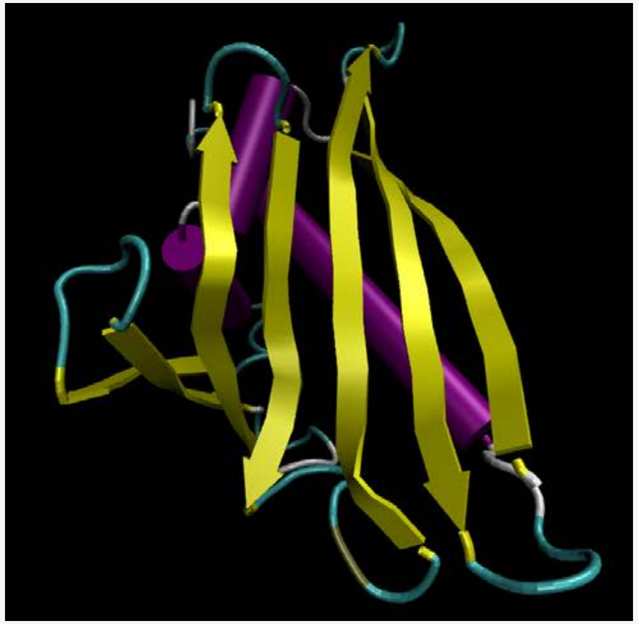
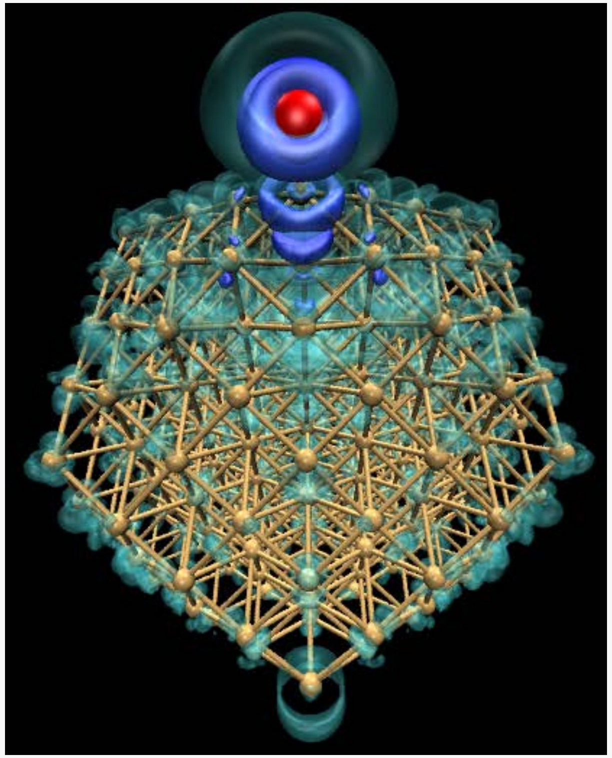
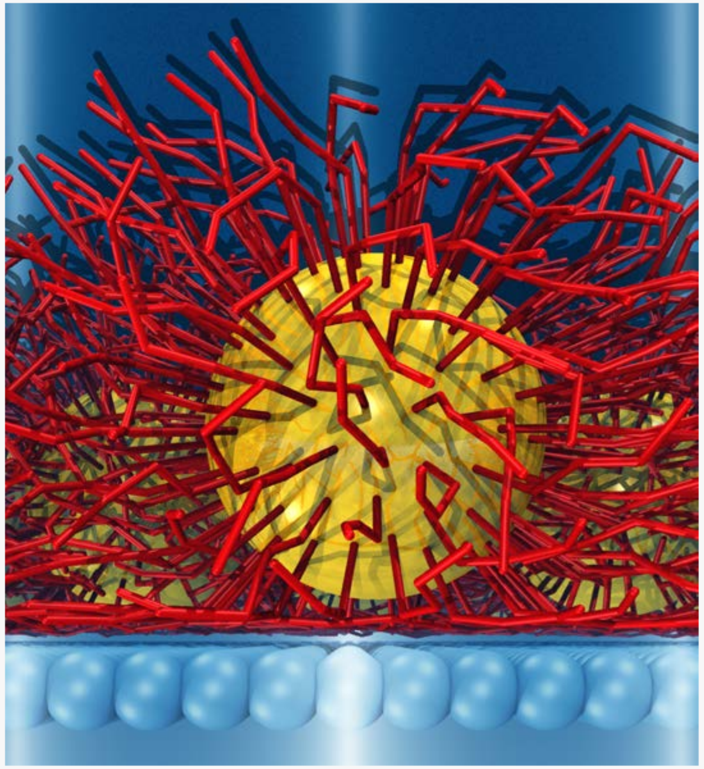
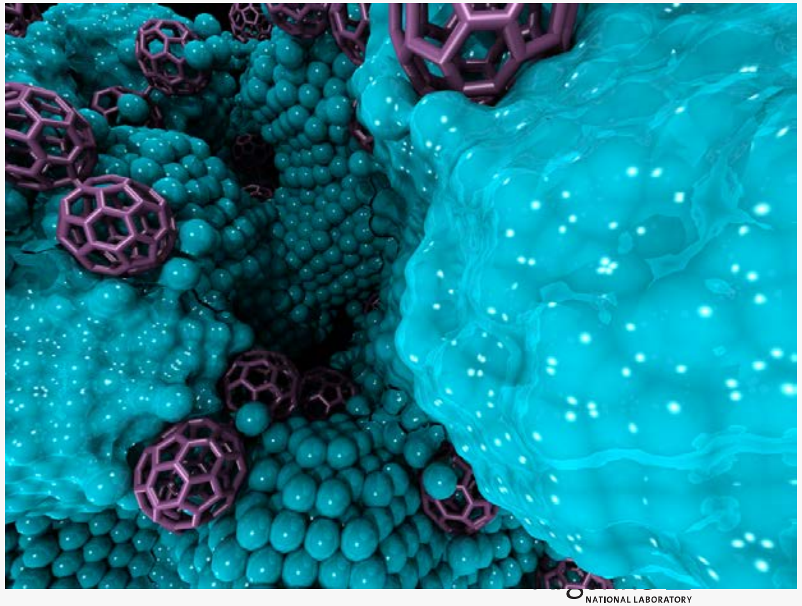
– Lots of domain-specific representations
– Many different file formats
– Animation
– Scriptable

VisIt & ParaView:

– Limited support for these types of representations, but improving

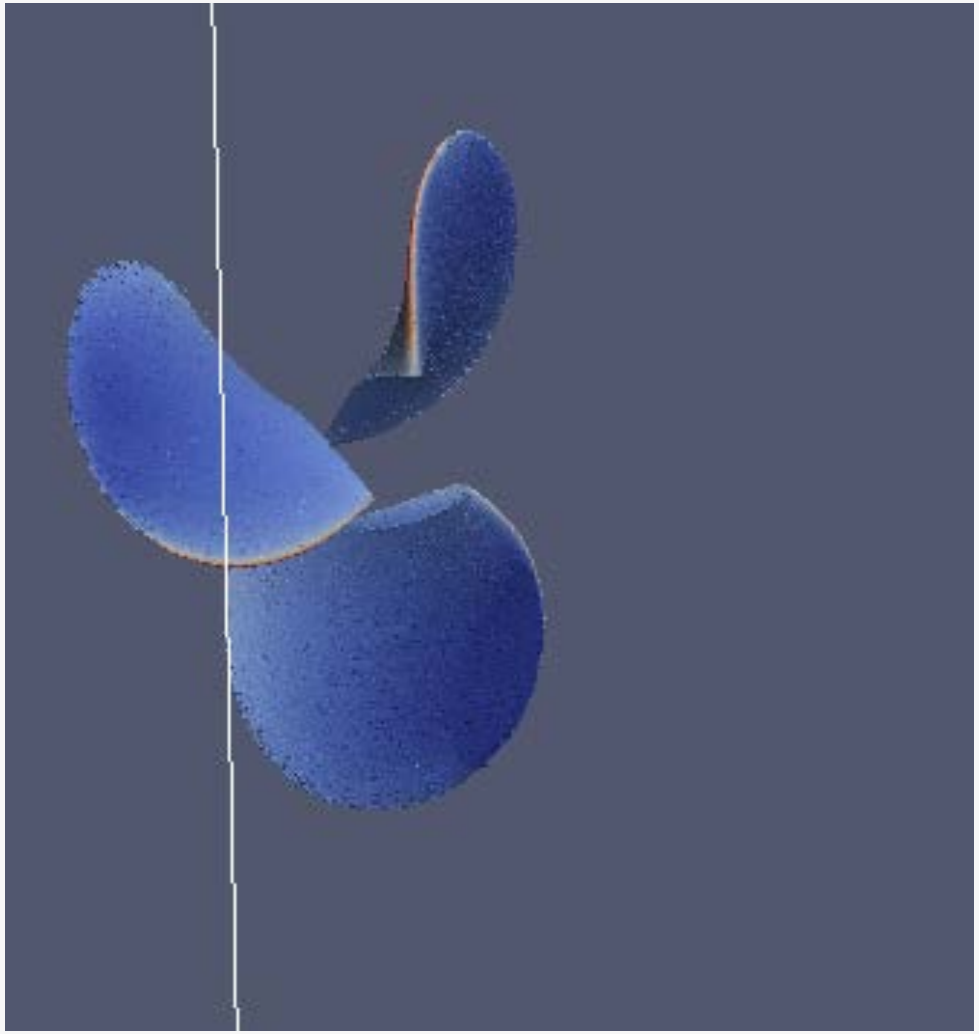
VTK:

– Anything's possible if you try hard enough

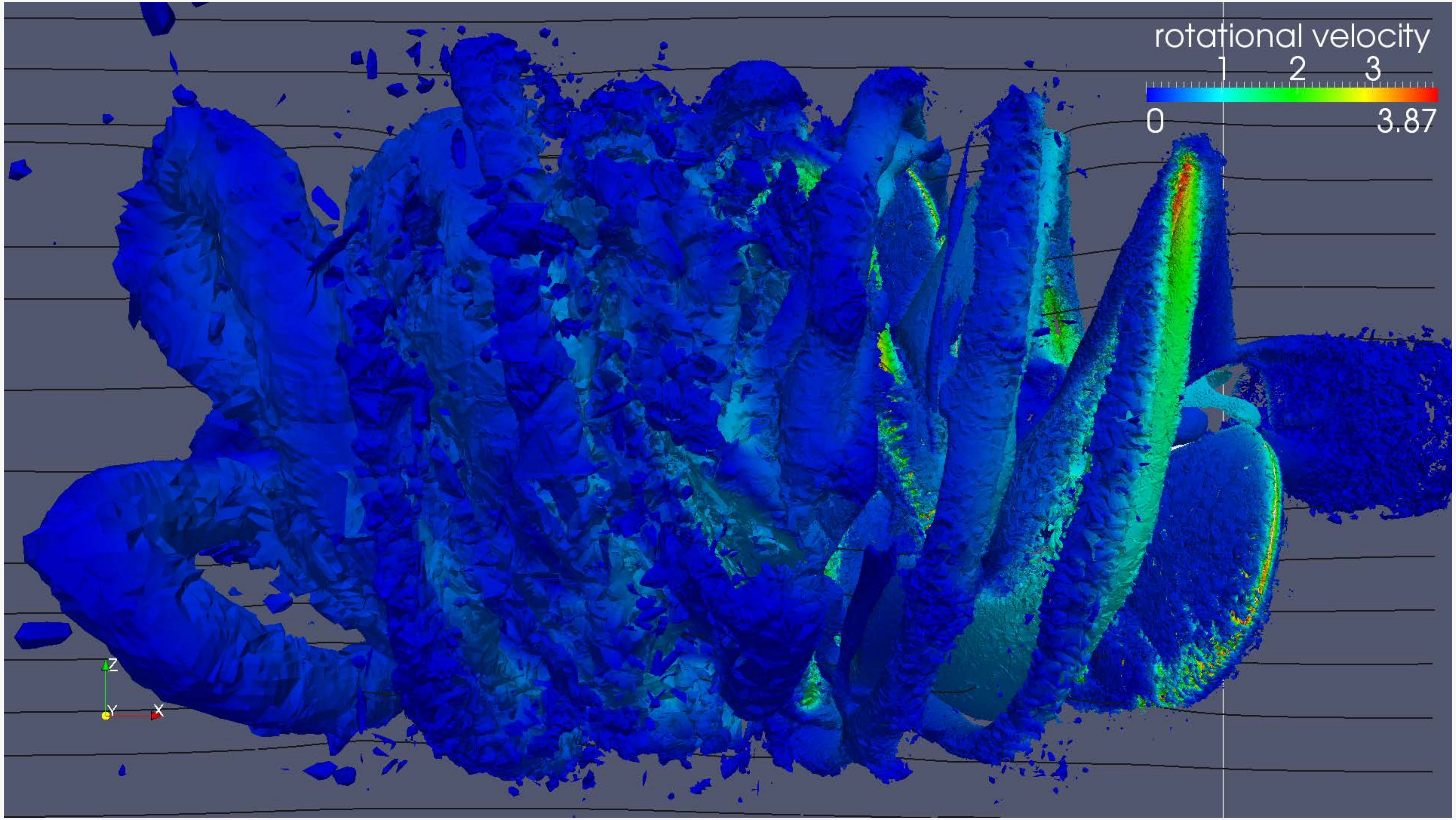





Visualization for Debugging

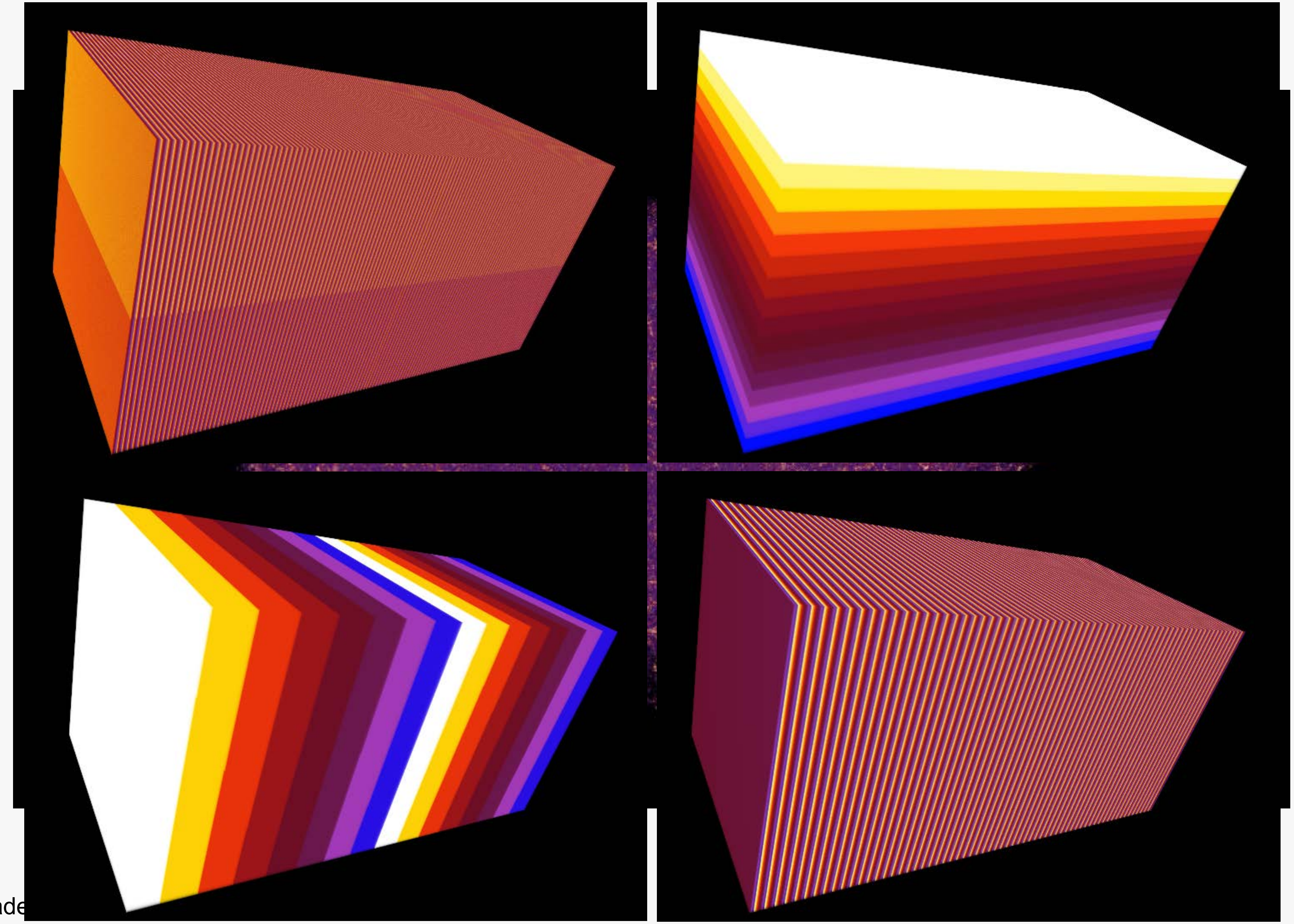
Visualization for Debugging



Visualization for Debugging



Visualization as Diagnostics: Color by Thread ID



Simple ParaView Scripting Example

ParaView States and Scripting

Choose File → Save State…

– .pvsm (for restoring state in interactive mode)
– saved on the client side

Choose File → Save State…

– .py (for use with pvbatch)
– saved on the client side

Edit .py script

– short example, loop over time steps, saving images

ParaView States and Scripting

my_script.py:

```
import sys

IMAGE_DIR = "/projects/my_project/FRAMES/”
IN_DATA_DIR = ”/projects/my_project/DATA/"
step_start= 0
step_count = 300
step_inc = 10

start_frame=int(sys.argv[1])
num_frames=int(sys.argv[2])

DATA_FILES = []

for i in range(step_start, step_start+(step_count*step_inc), step_inc):
    temp_file = ”%s/data_step_%04d%s" % (IN_DATA_DIR, i, “.vtu”)
    DATA_FILES.append(TEMP_FILE)
```

ParaView States and Scripting

```
…
from paraview.simple import *
paraview.simple._DisableFirstRenderCameraReset()
RenderView1 = CreateRenderView()
RenderView1.ViewSize = [1920, 1080]
…
mydata = XMLUnstructuredGridReader( guiName=”my_datafiles*",
CellArrayStatus=['temp', 'equiv_ratio', 'rank'], FileName=DATA_FILES)
```

ParaView States and Scripting

```
…
#Render()

time_vals = mydata.TimestepValues

for i in range(start_frame, start_frame+num_frames):

    RenderView1.ViewTime = time_vals[i]

    IMAGE_FILE="%s/frame_%04d.png" % (IMAGE_DIR, i)
    print "saving: " + IMAGE_FILE
    WriteImage(IMAGE_FILE)
```

Running ParaView in Batch

```
> qsub –n 1 –t 1:00 –A <project_id> pvbatch my_script.py 0 100
> qsub –n 1 –t 1:00 –A <project_id> pvbatch my_script.py 100 100
> qsub –n 1 –t 1:00 –A <project_id> pvbatch my_script.py 200 100
```

```
> soft add +ffmpeg
> ffmpeg –r 25 –i FRAMES/frame_%04d.png –r 25 -pix_fmt yuv420p movie.mp4
```

In Situ Visualization and Analysis

Five orders of magnitude between compute and I/O capacity on Titan Cray system at ORNL

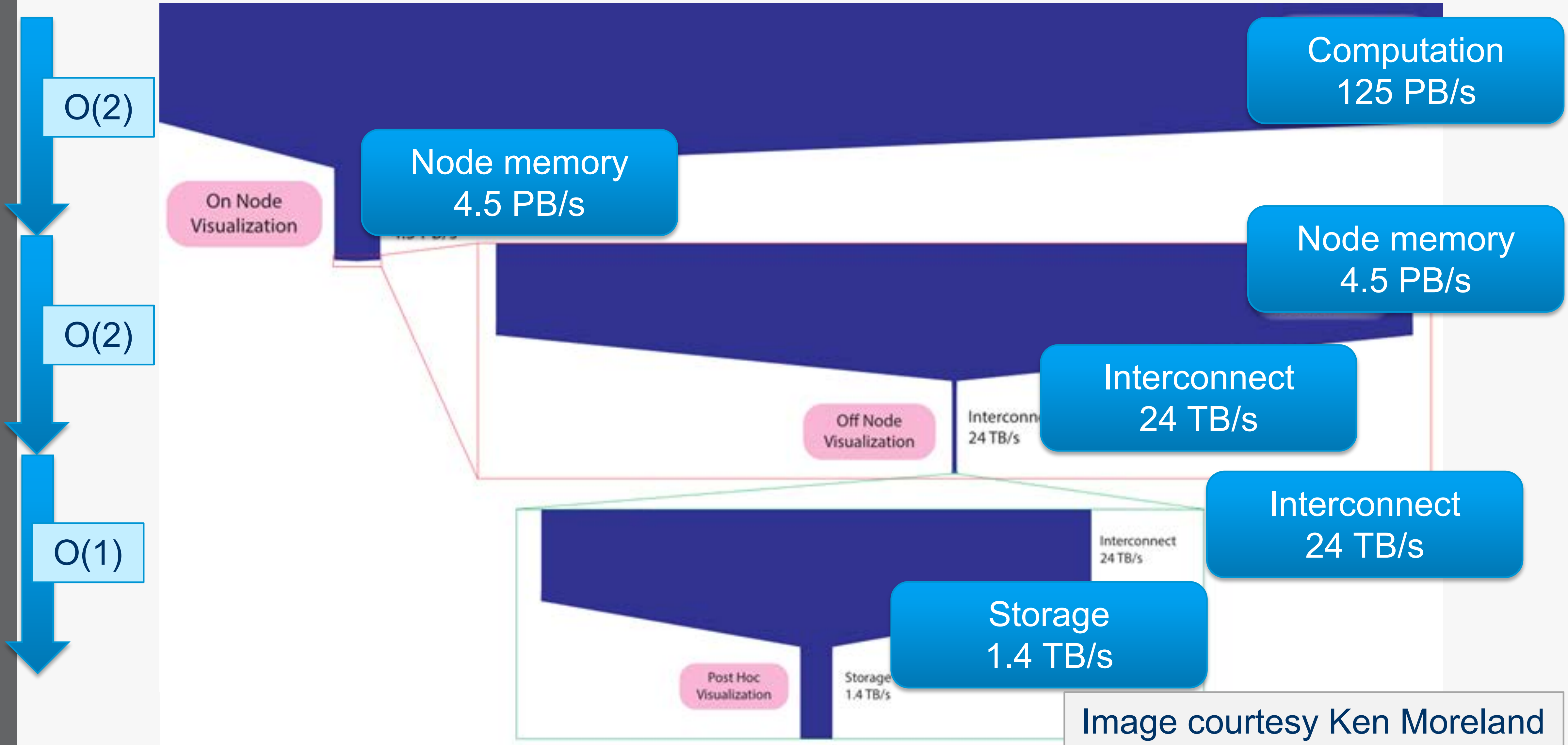


Image courtesy Ken Moreland

What are the problems?

- Not enough I/O capacity on current HPC systems, and the trend is getting worse.
- If there’s not enough I/O, you can’t write data to storage, so you can’t analyze it: lost science.
- Energy consumption: it costs a lot of power to write data to disk.
- Opportunity for doing better science (analysis) when have access to full spatiotemporal resolution data.

Slide courtesy the SENSEI team www.sensei-insitu.org

Two Frameworks for In Situ Vis and Analysis at ALCF

SENSEI insitu

- "Write once, run everywhere" design
- Data model based on VTK from Kitware
- Supports a variety of backends, including ParaView/Catalyst, VisIt/LibSim, ADIOS, Python

Ascent

- Flyweight design, minimizes dependencies
- Data model based on Conduit from LLNL
- Vis and analysis algorithms implemented in VTK-m

Instrumenting Simulation Codes

SENSEI
insitu

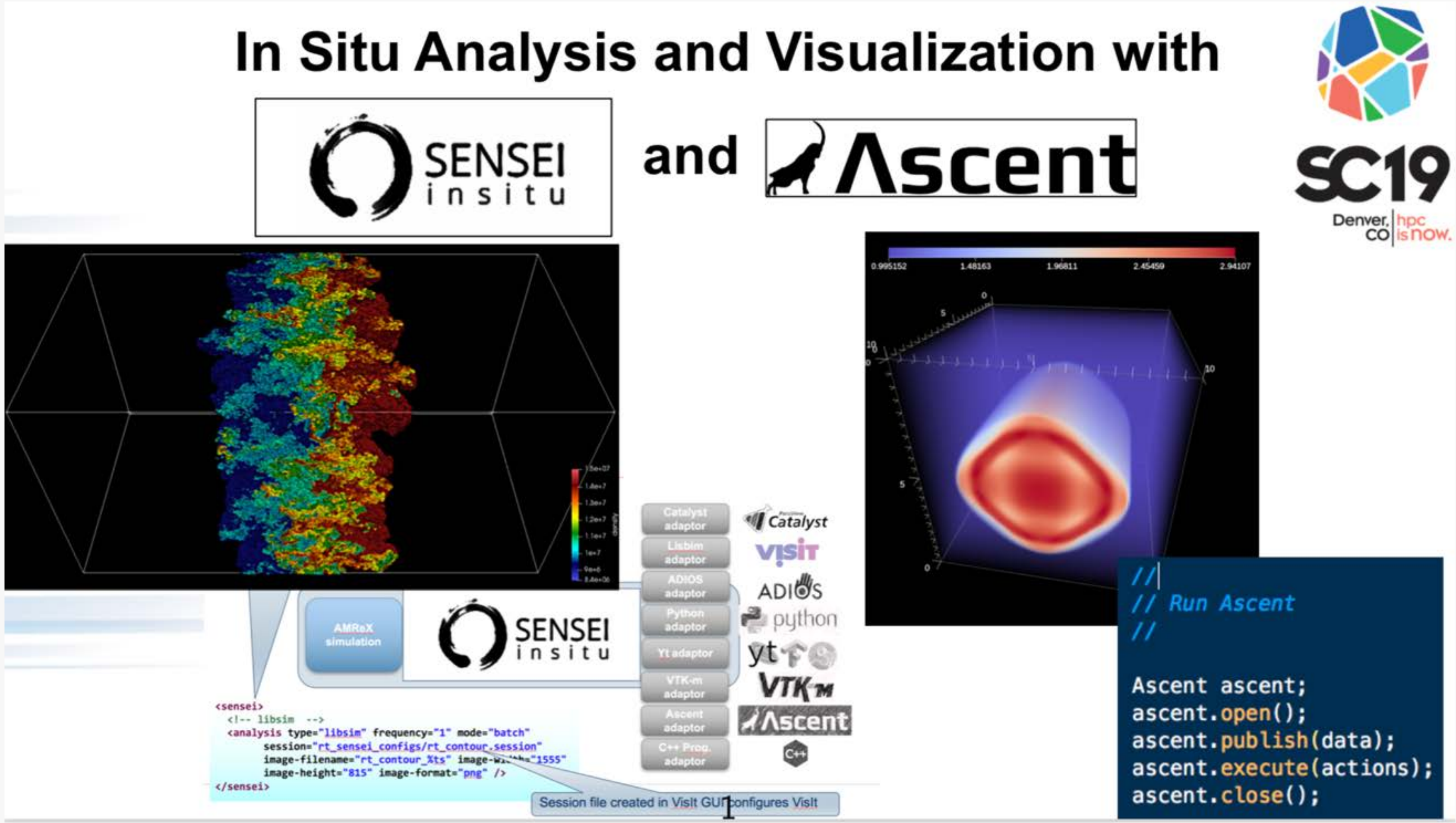
Ascent

```
initialize sim
if do_insitu bridge::initialize
do
  compute new state
  if do_io write plot file
  if do_insitu bridge::execute
while !done
if do_insitu bridge::finalize
finalize sim
```

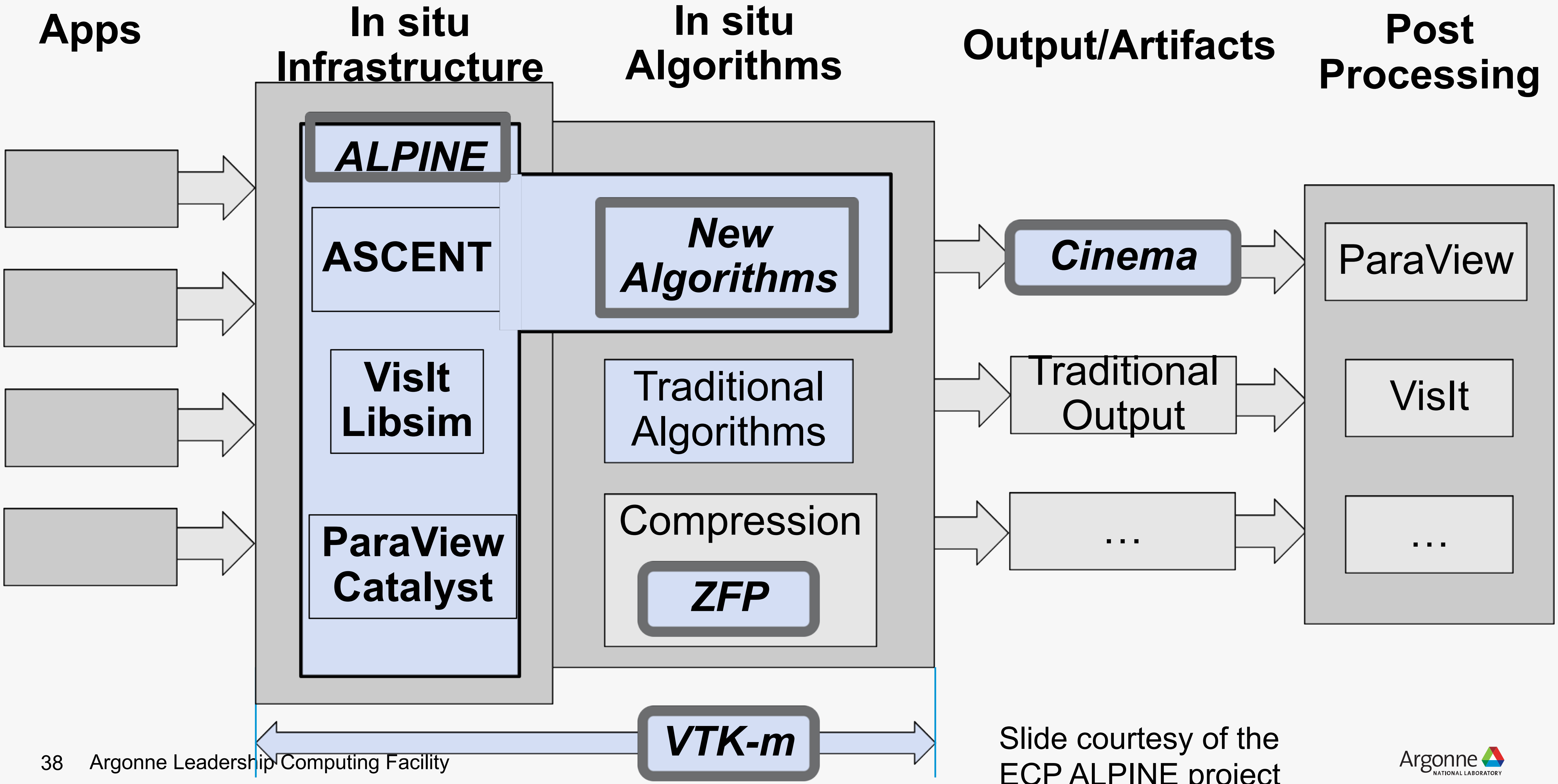
```
//
// Run Ascent
//

Ascent ascent;
ascent.open();
ascent.publish(data);
ascent.execute(actions);
ascent.close();
```

SENSEI + ASCENT tutorial at SC19 and SC20

Slides and Virtual Machine available here:
https://sensei-insitu.org/tutorials/sc19.html
https://ix.cs.uoregon.edu/~hank/sc20/

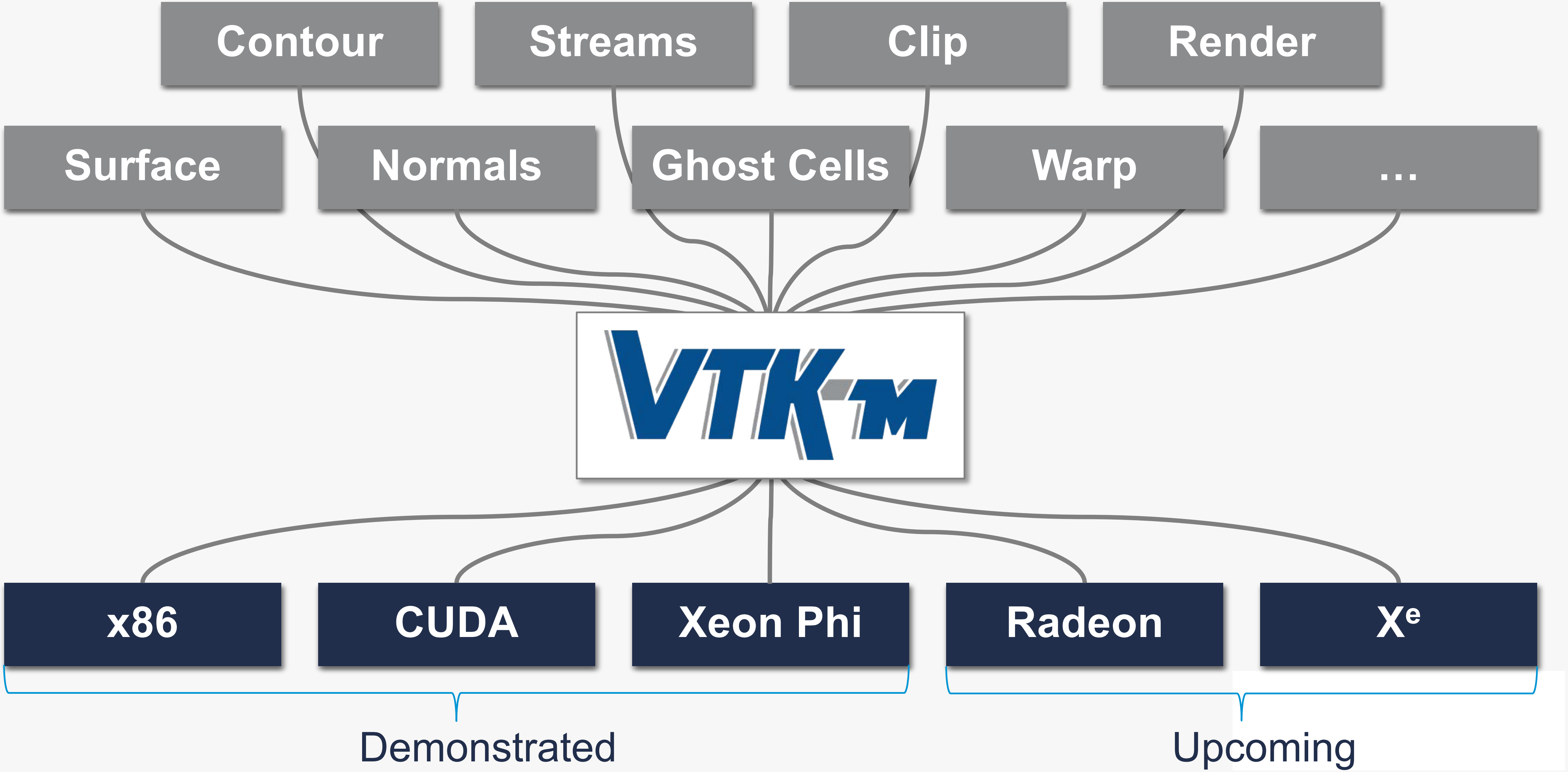


Exascale Computing Project
Software Technology Data and Visualization



Slide courtesy of the ECP ALPINE project

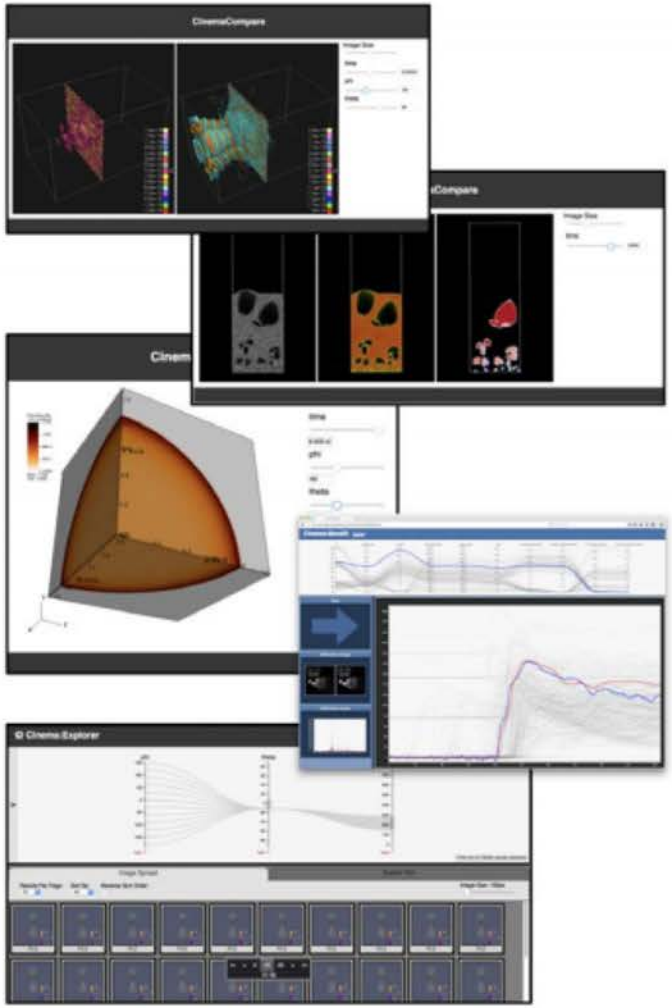
VTK-m’s main thrust: a write-once-run-everywhere framework



Slide courtesy of the ECP VTK-m project

What is Cinema?

- **Cinema** is part of an integrated workflow, providing a method of extracting, saving, analyzing or modifying and viewing complex data artifacts from large scale simulations.
  - If you're having difficulty exploring the complex results from your simulation, Cinema can help.
- **The Cinema 'Ecosystem'** is an integrated set of writers, viewers, and algorithms that allow scientists to export, analyze/modify and view Cinema databases.
  - This ecosystem is embodied in widely used tools (**ParaView, VisIt, Ascent**) and the database specification.



Slide courtesy of Los Alamos National Lab and the CINEMA project

Additional Resources

Visualization Help

support@alcf.anl.gov

Publication Images & Covers

Animations

- SC Visualization Showcase [Best Vis Finalist 2014-2020]
- APS Division of Fluid Dynamics Gallery of Fluid Motion
- SC Gordon Bell Submissions
- Press Releases

InSitu Vis and Analysis

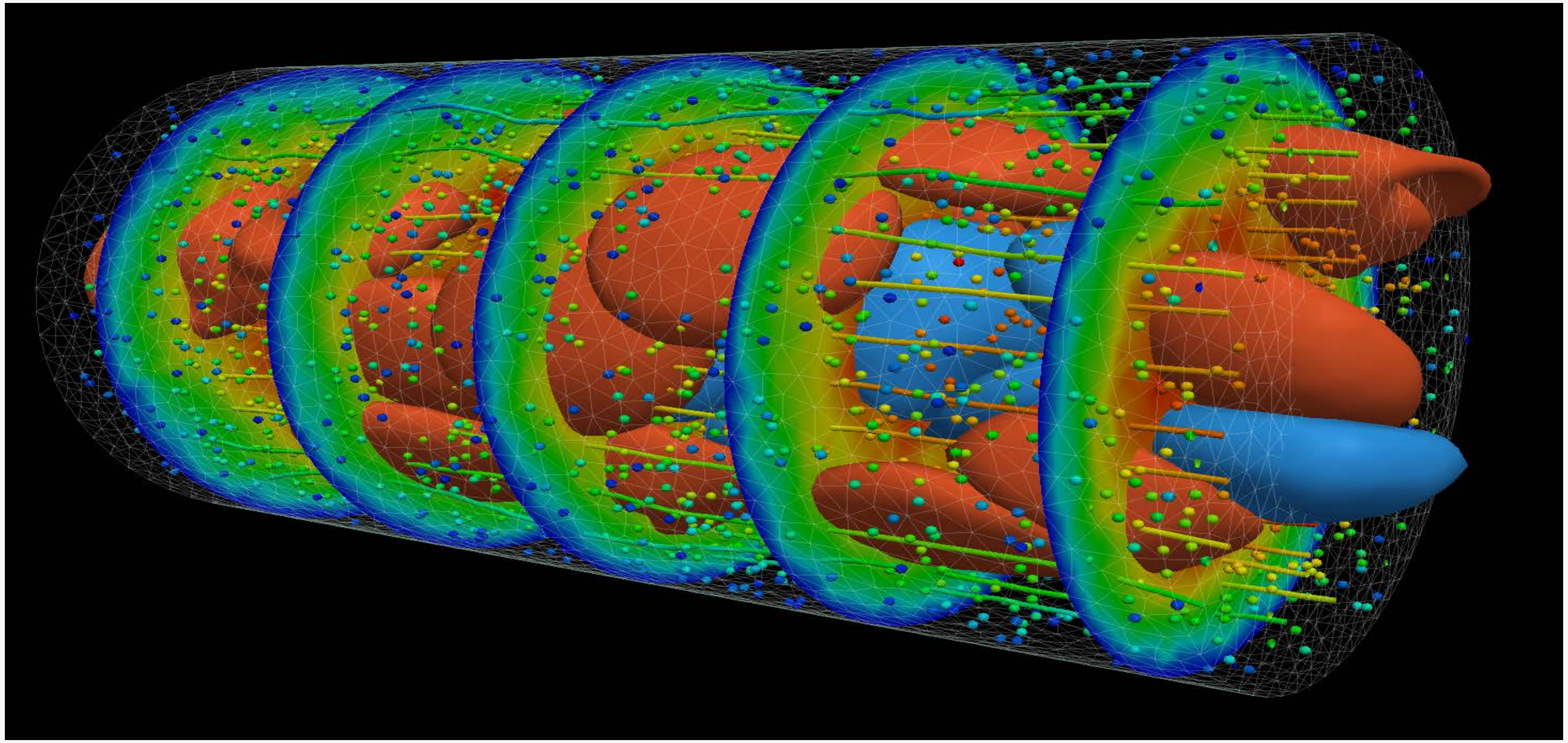
More info…

https://www.alcf.anl.gov/support-center/cooley/cooley-system-overview

http://www.visitusers.org/

https://www.paraview.org/Wiki/ParaView

https://imagemagick.org/index.php



In situ additional resources

- Project page https://sensei-insitu.org/
- Repository https://gitlab.kitware.com/sensei/sensei

- Website + Docs: http://ascent-dav.org
- Repository: https://github.com/Alpine-DAV/ascent

Exascale Computing Project additional resources

ALPINE	https://alpine.dsscale.org/
VTK-m	https://m.vtk.org/index.php/Main_Page
Cinema	https://cinemascience.github.io/

QUESTIONS?

Joe Insley
insley@anl.gov

Silvio Rizzi
srizzi@anl.gov

Janet Knowles
jknowles@anl.gov

Victor Mateevitsi
vmateevitsi@anl.gov