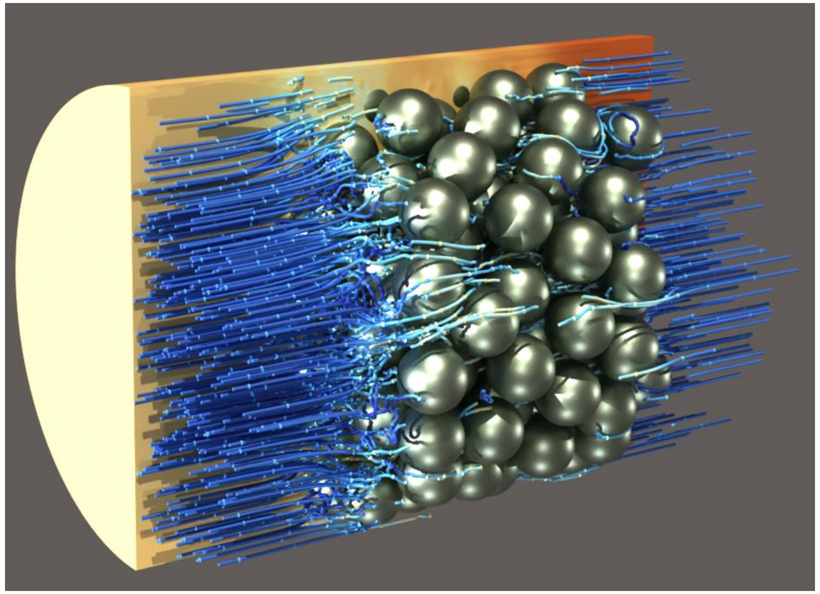
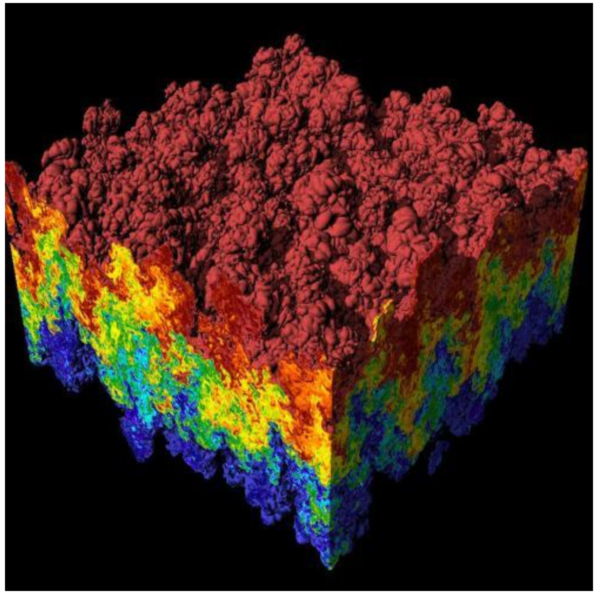
ARGONNE

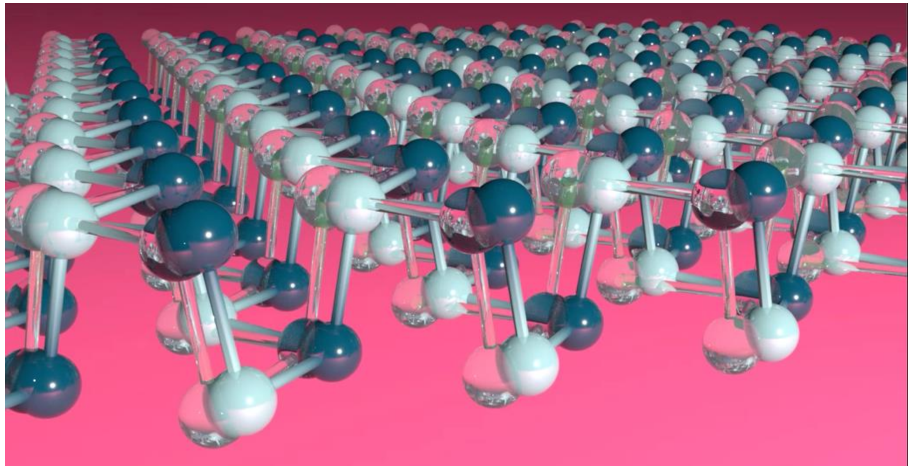
ATPESC2026

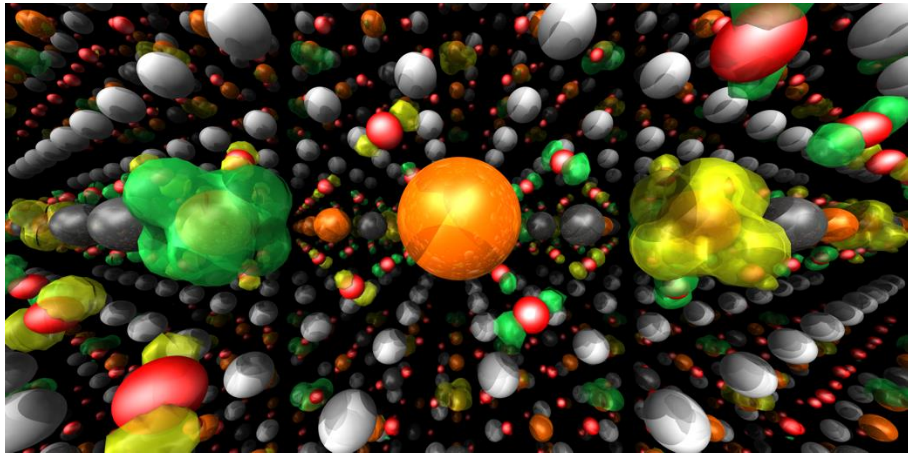
EXTREME-SCALE COMPUTING

Data Analysis and Visualization









Visualization & Data Analysis

Time	Title of presentation	Lecturer
8:30 am	**Data Analysis and Visualization Introduction**	Joe Insley, *ANL* Silvio Rizzi, *ANL*
9:15 am	**Visualization and Analysis of HPC Simulation Data with VisIt**	Cyrus Harrison, *LLNL*
10:15 am	*Break*	
10:45 am	**Large Scale Visualization with ParaView**	Cory Quammen, *Kitware* Patrick Avery, *Kitware*
11:45 am	**Leveraging AI for Scientific Visualization**	Victor Mateevitsi, *ANL*
12:30 pm	*Lunch*	

Here's the plan…

- Examples of visualizations
- Visualization tools and formats
- Data representations
- Visualization for debugging
- In Situ Visualization and Analysis

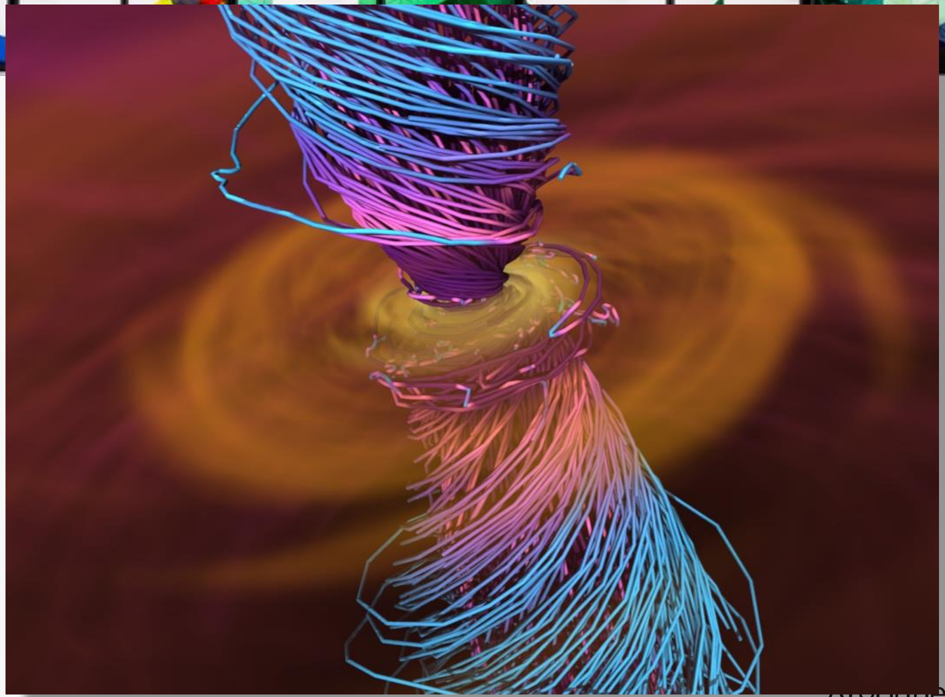
SCIENTIFIC DATA VISUALIZATION

SCIENTIFIC DATA VISUALIZATION

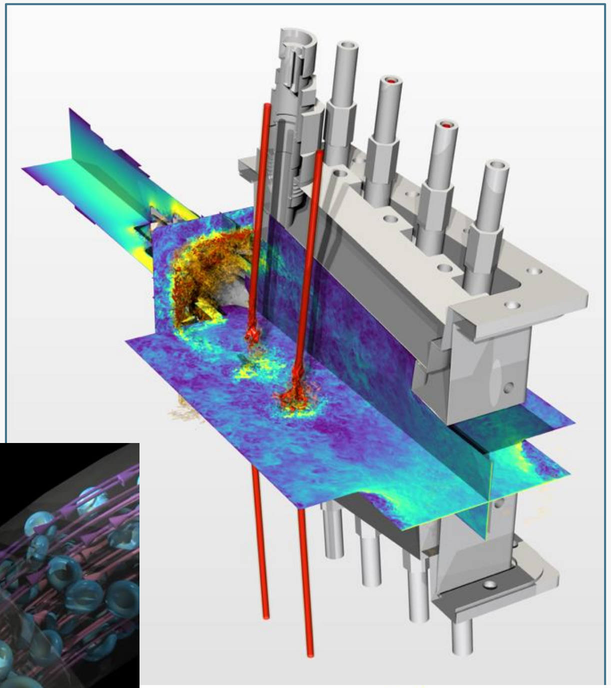
10293	43952	59302	40354
59390	03995	50492	69594
68303	69904	30405	10597
58303	30294	39853	92962
79402	95943	09390	49303
09394	84303	59032	15322

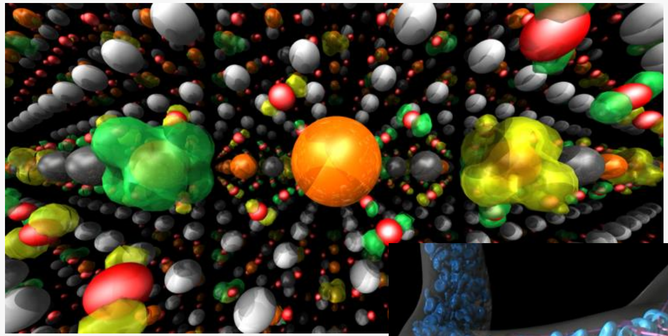
SCIENTIFIC DATA VISUALIZATION

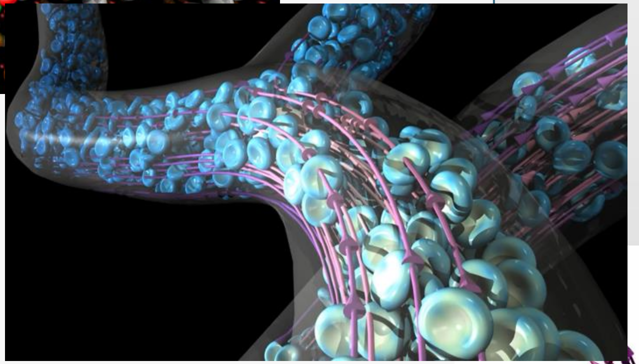
10293	43952	59302	40354
59390	03995	50492	69594
68303	69904	30405	10597
58303	30294	39853	92962
79402	95943	09390	49303
09394	84303	59032	15322



SCIENTIFIC DATA VISUALIZATION

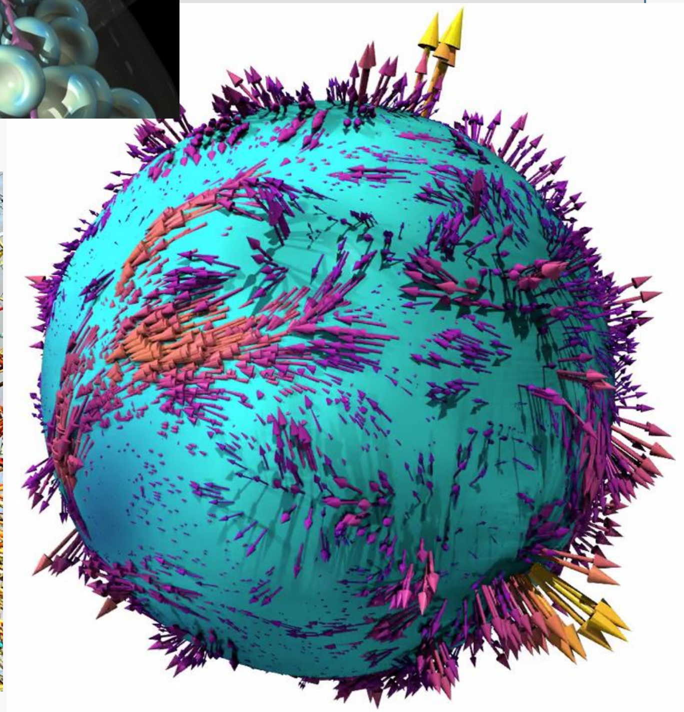






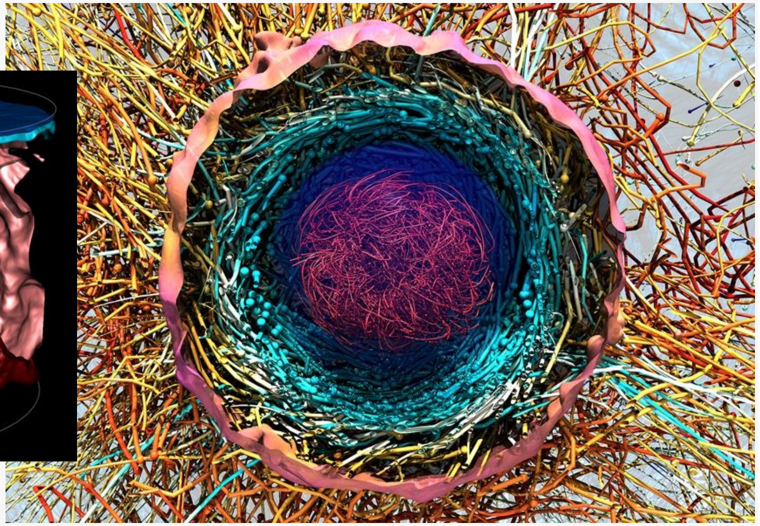
Use graphical attributes

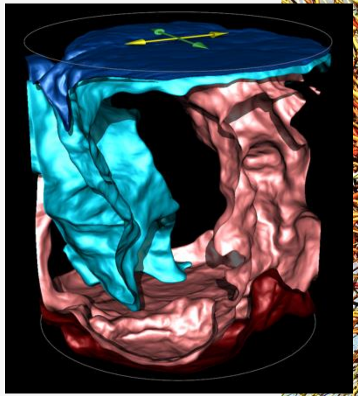
- Color
- Size
- Position
- Orientation
- Etc



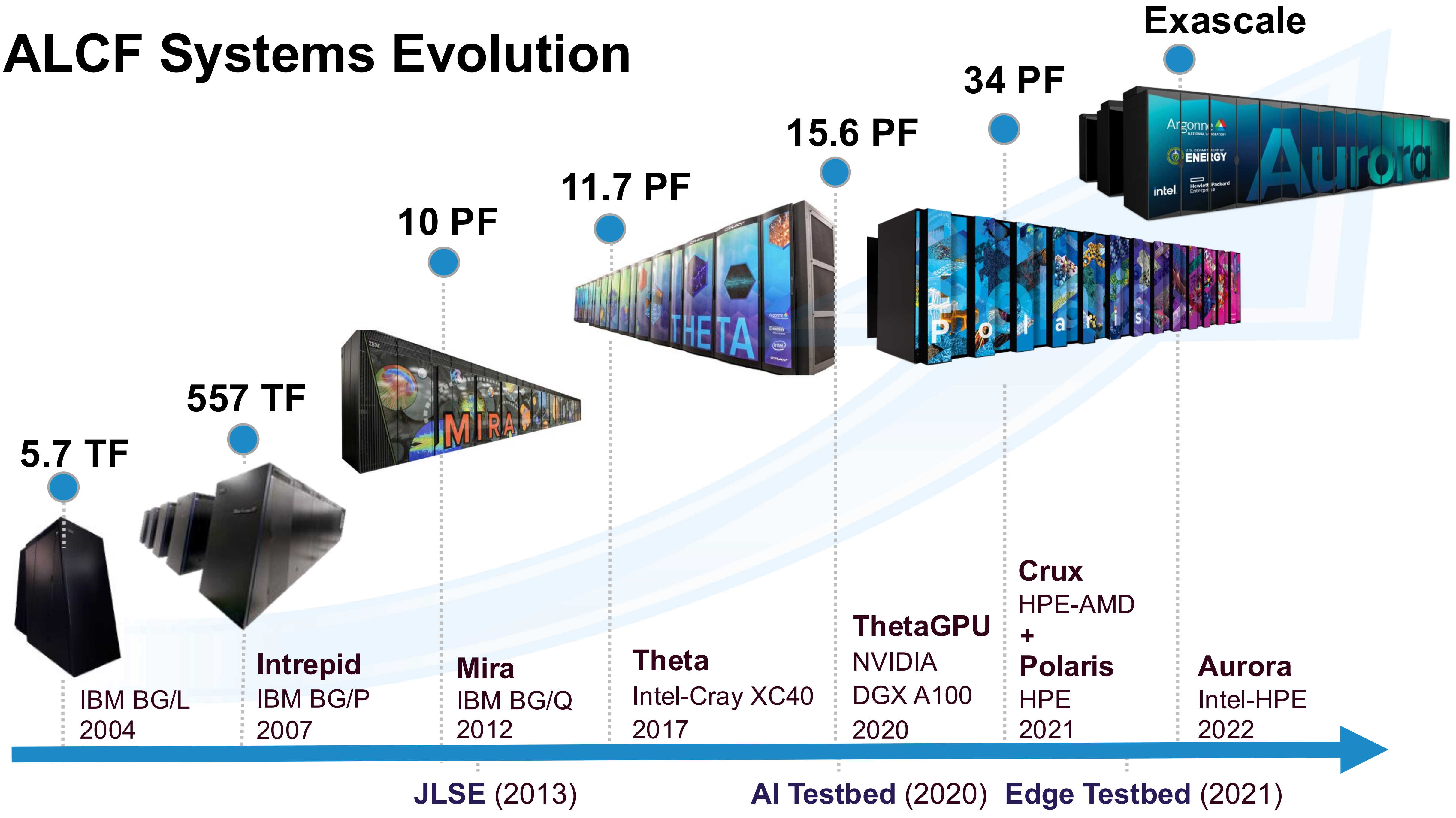
To represent attributes of the data

- Temperature
- Pressure
- Velocity
- Density
- Etc



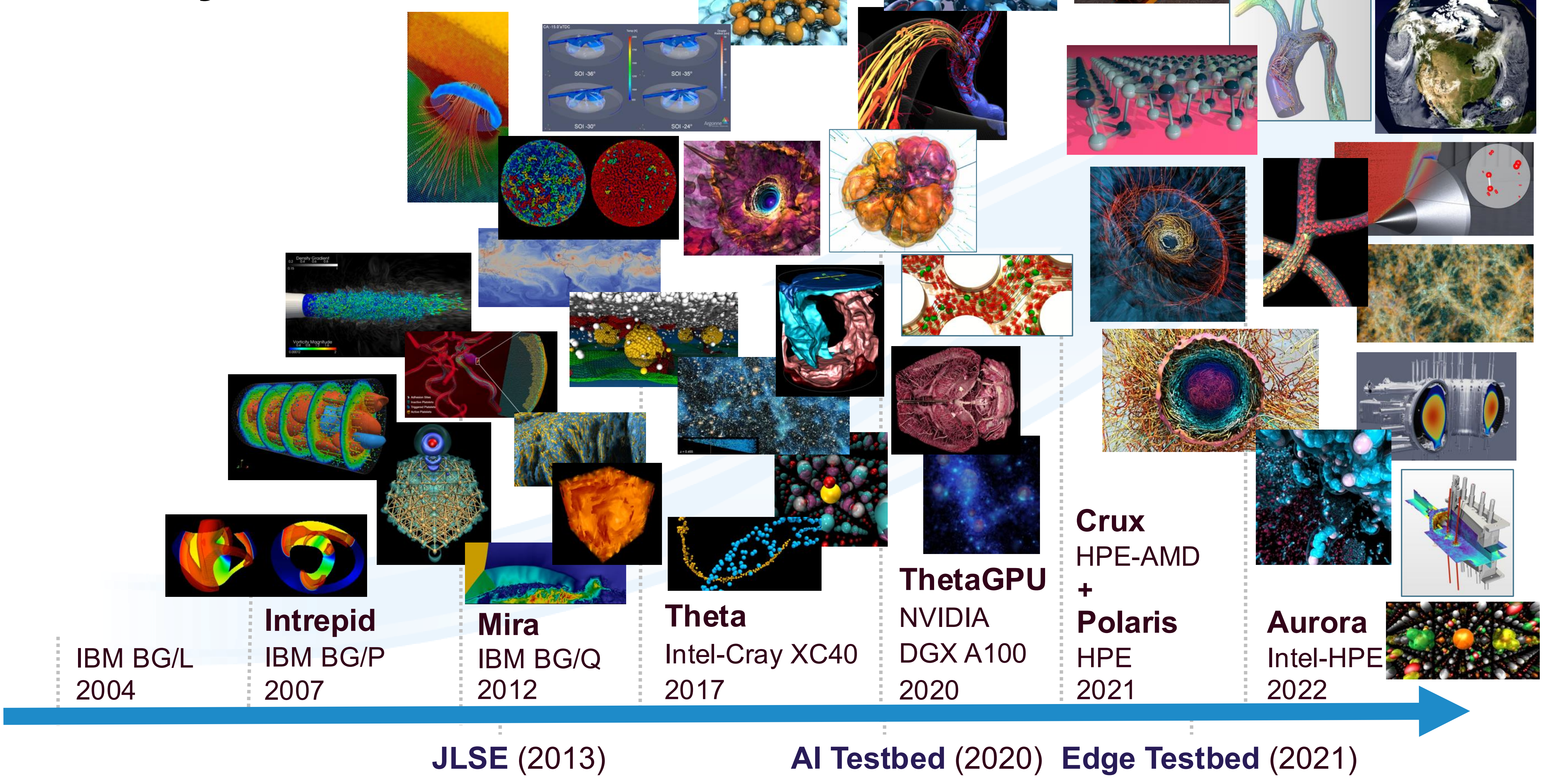


ALCF Systems Evolution



Argonne NATIONAL LABORATORY

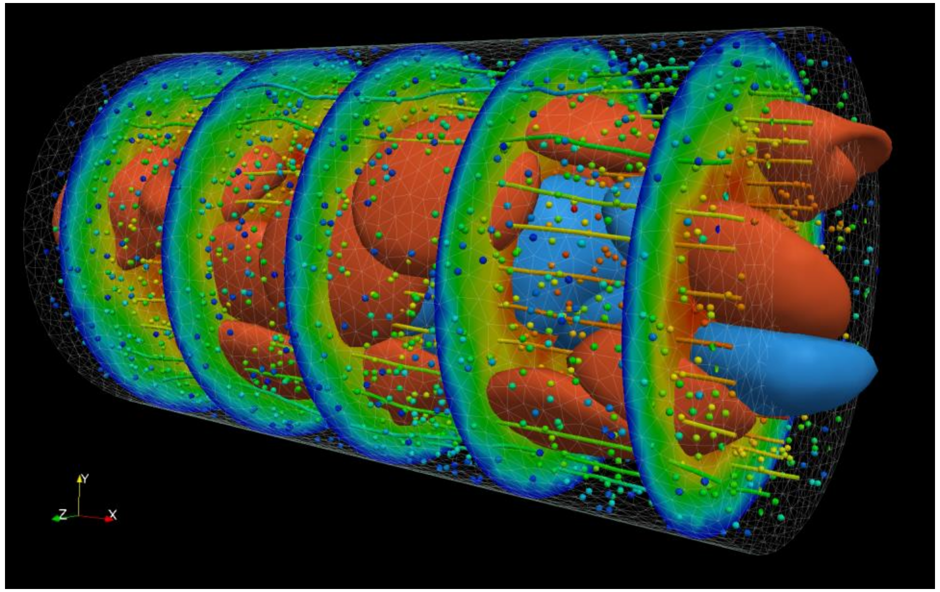
ALCF Systems Evolution



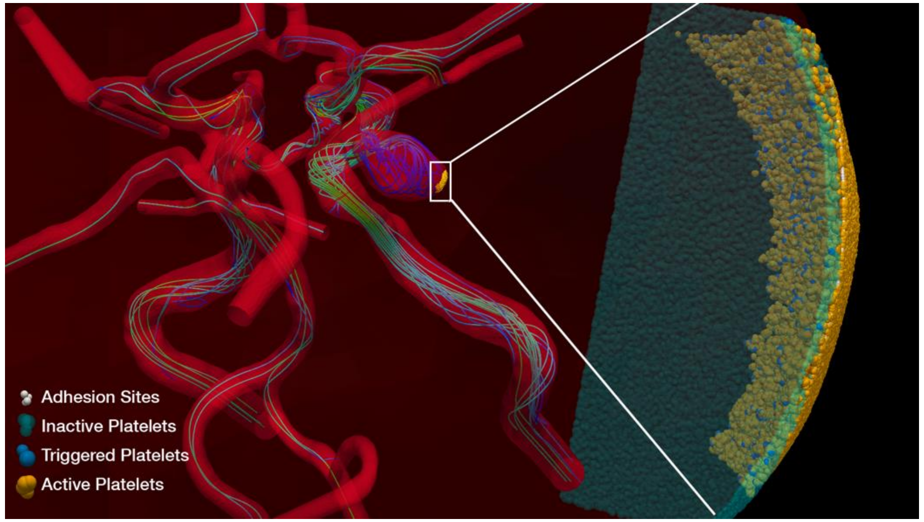
Multi-Scale Simulation / Visualization Arterial Blood Flow

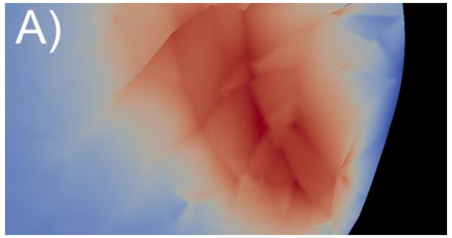
PI: George Karniadakis, Brown University

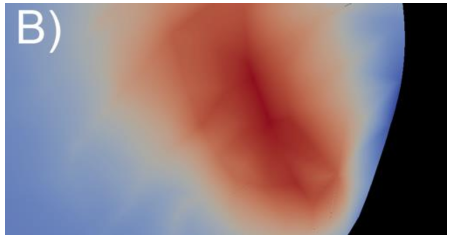
2010



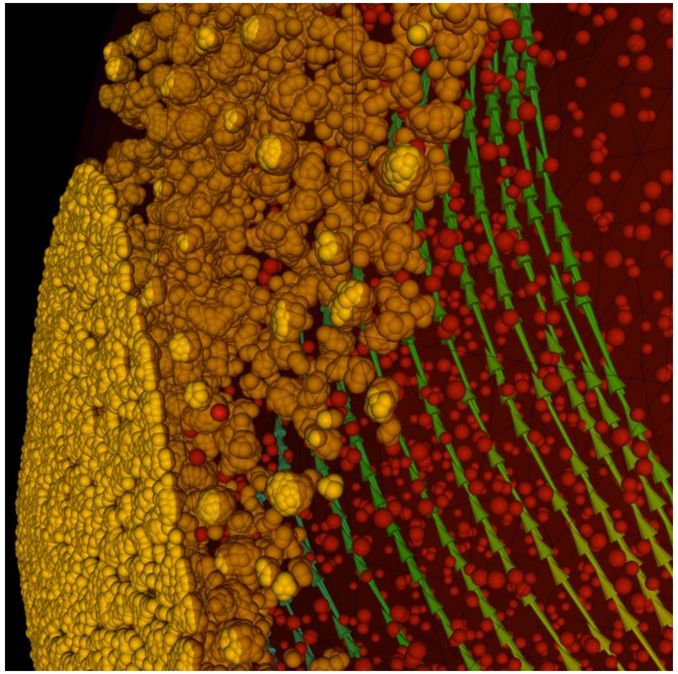
2011



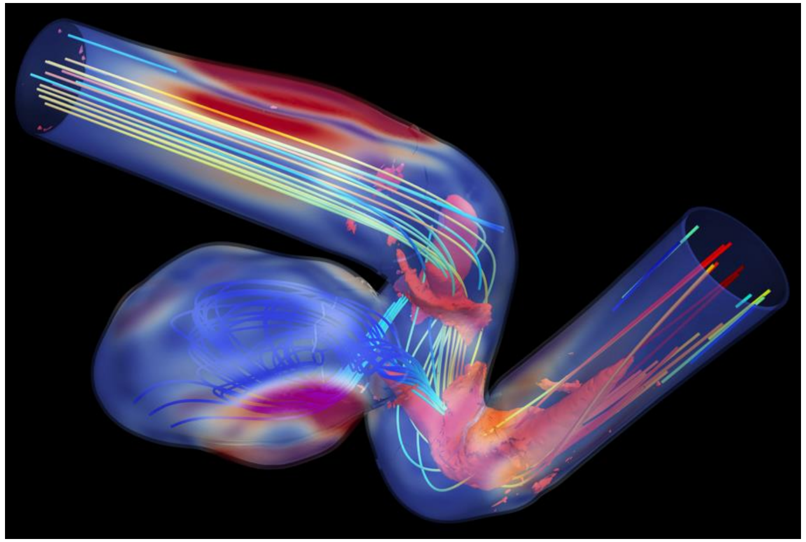




2012

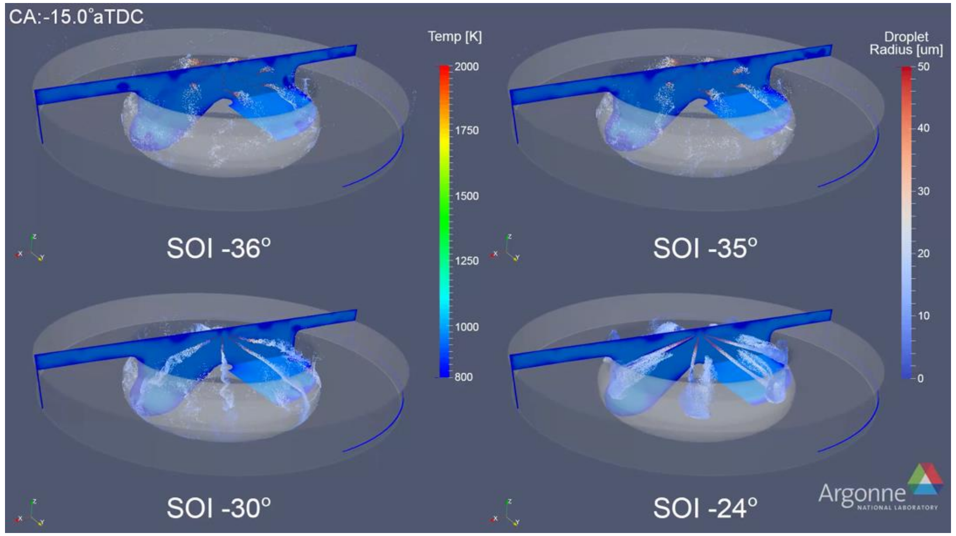


2014

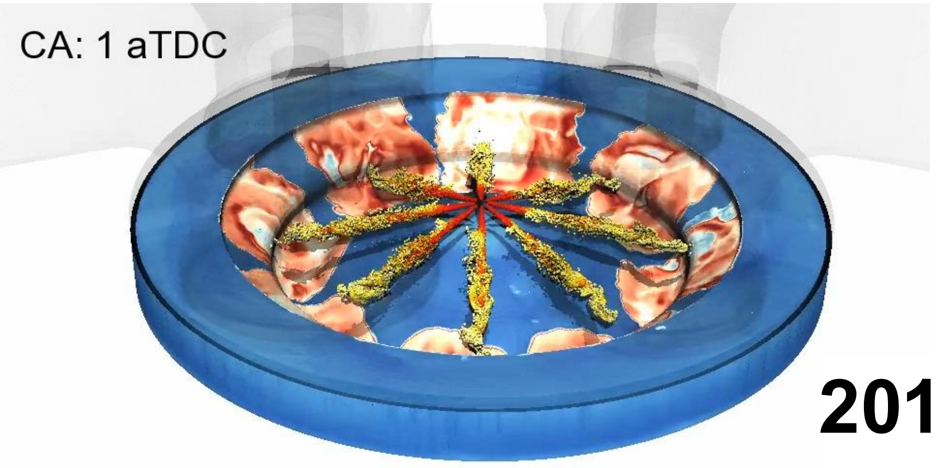


Engineering / Combustion / Biofuels

PI: Sibendu Som, Argonne National Laboratory

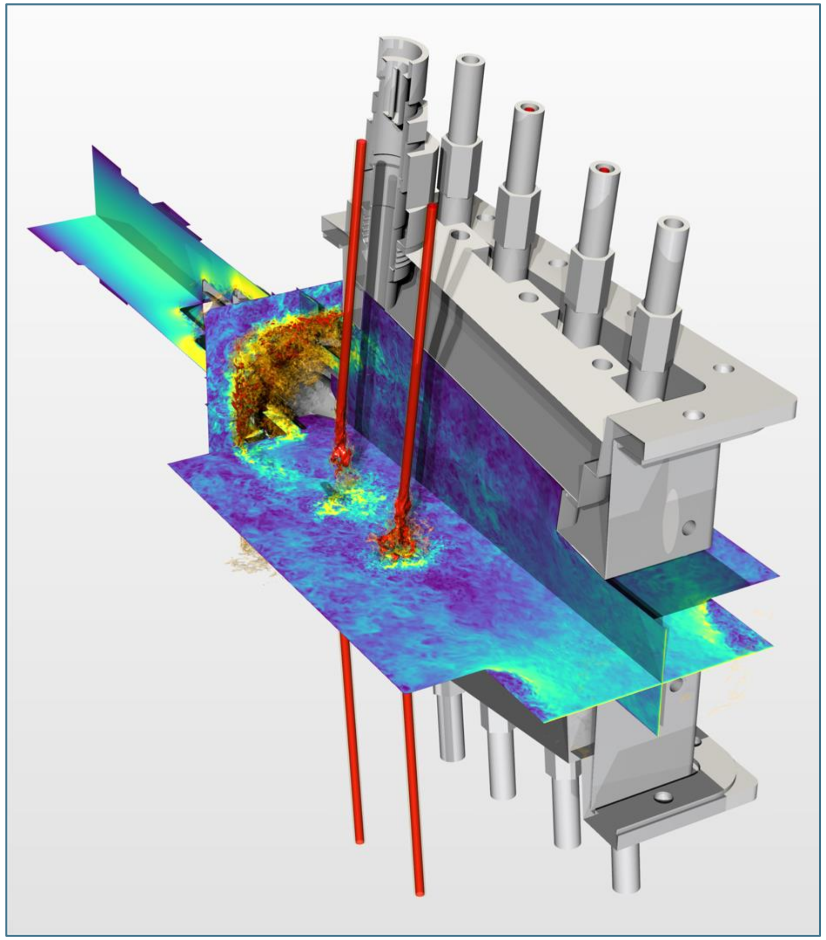


2015



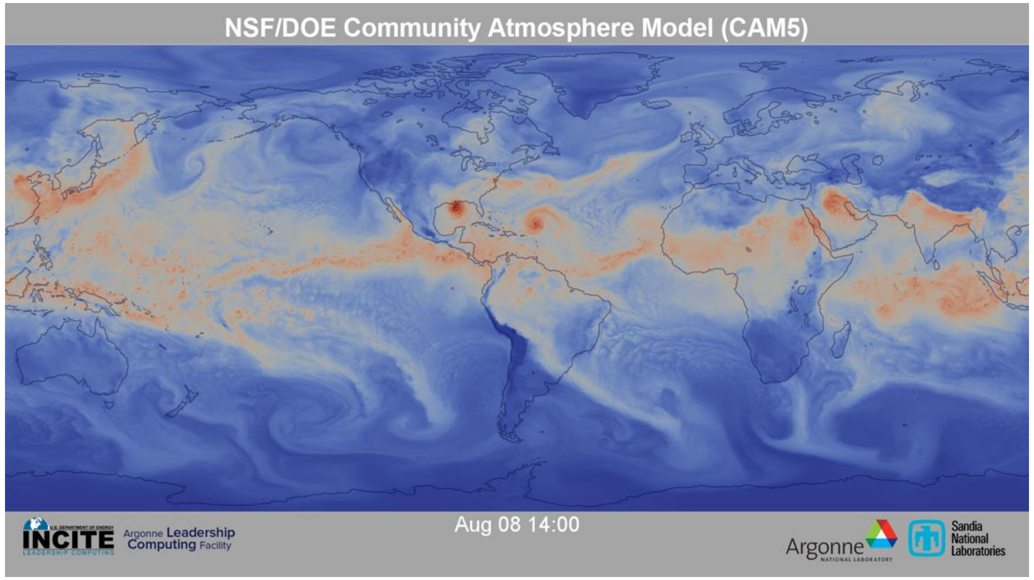
2017

2021



2023

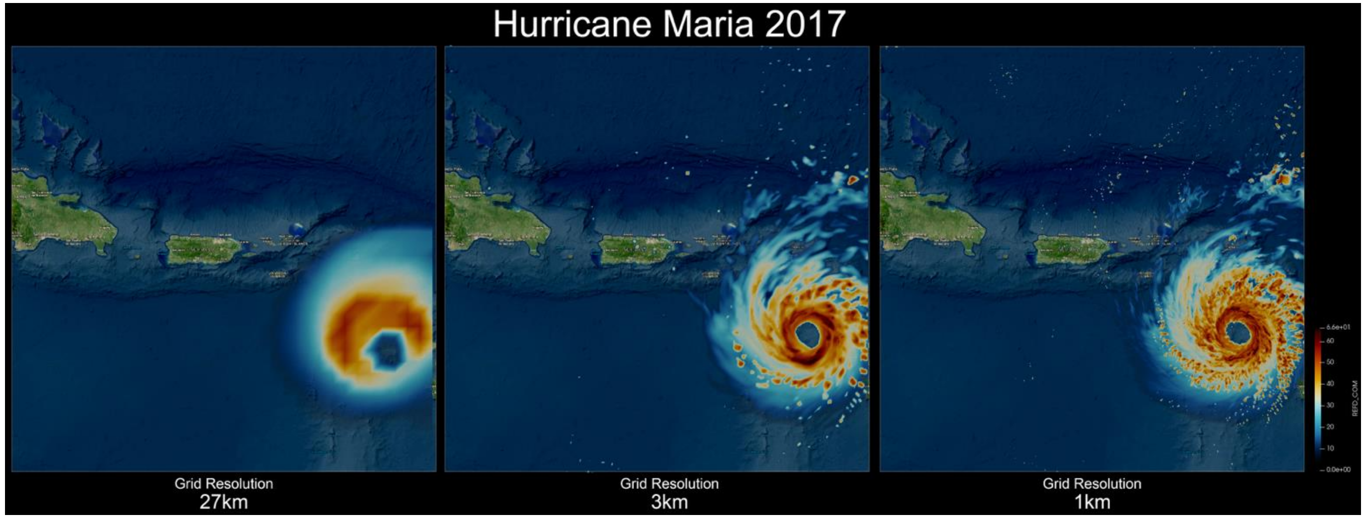
Climate

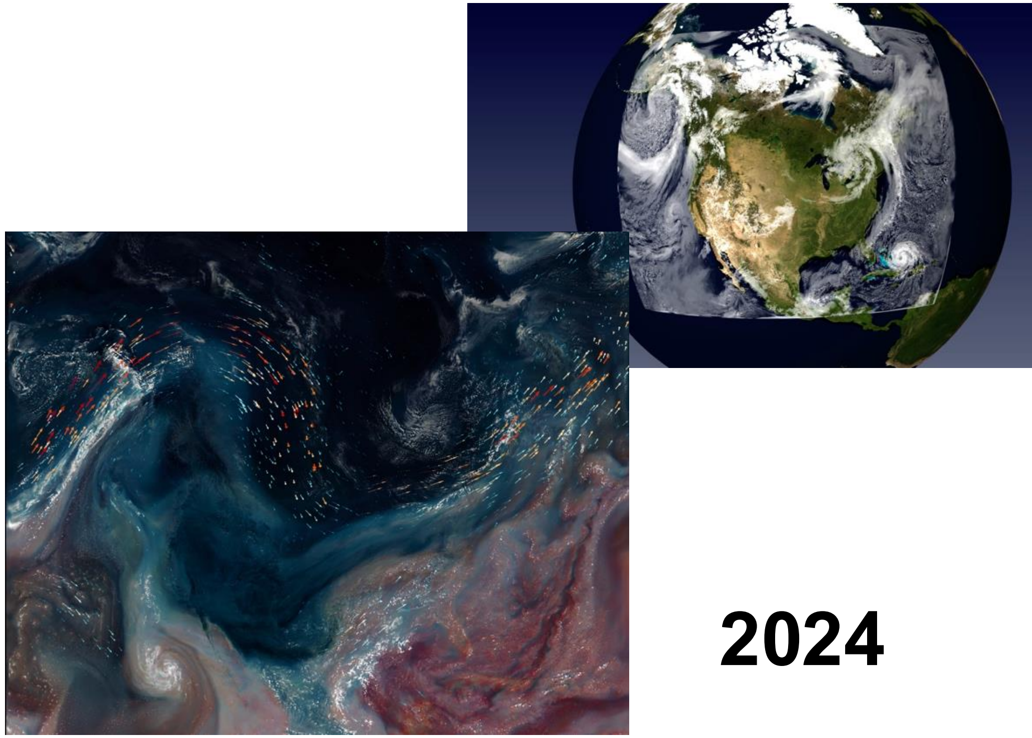


PI: Warren Washington, National Center for Atmospheric Research

2012

2022



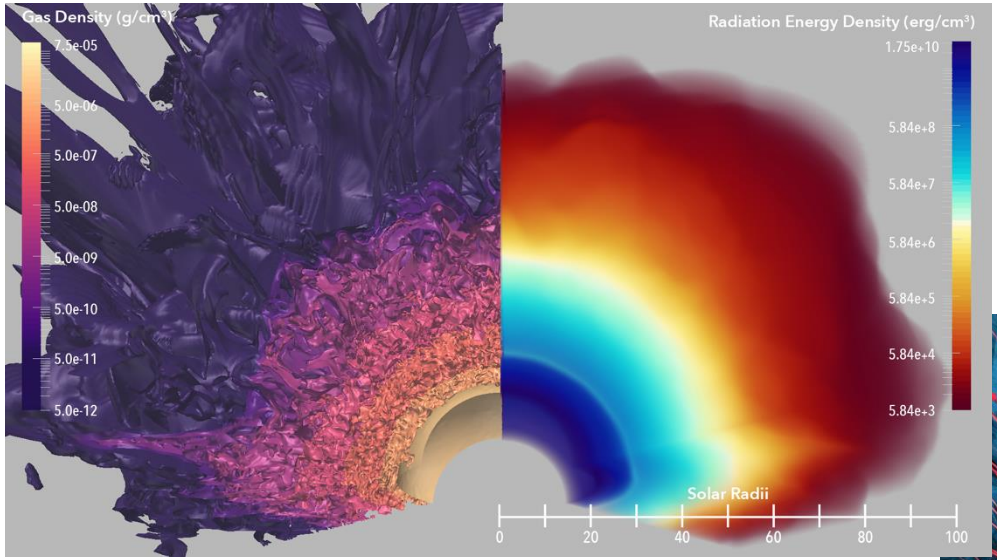


2024

PI: Rao Kotamarthi, Argonne National Laboratory

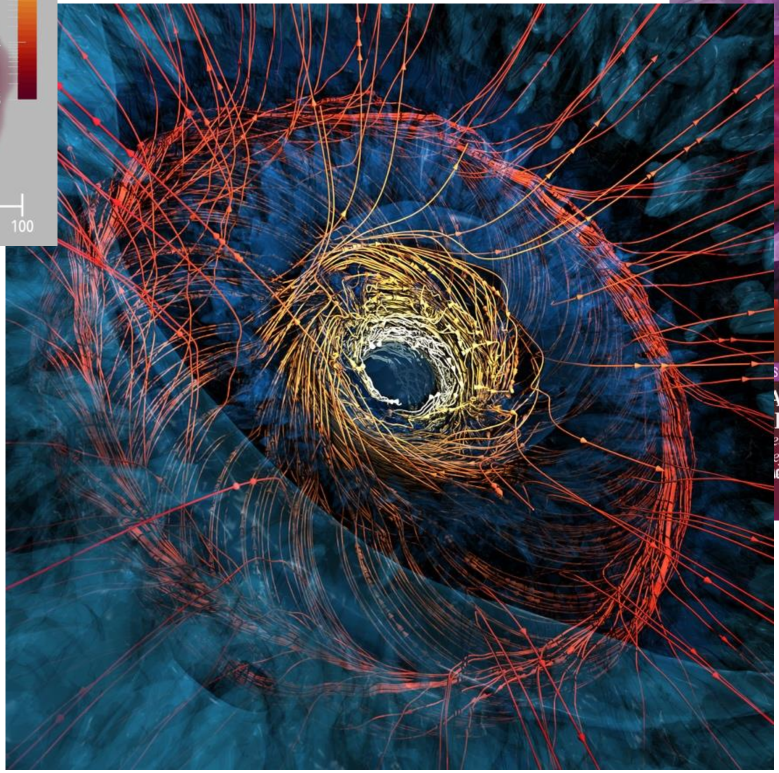
Physics: Stellar Radiation

PI: Lars Bildsten, University of California, Santa Barbara



2017

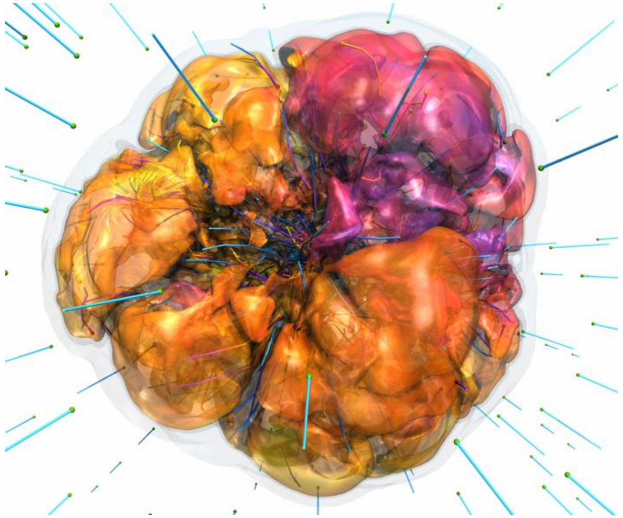
2018



2021

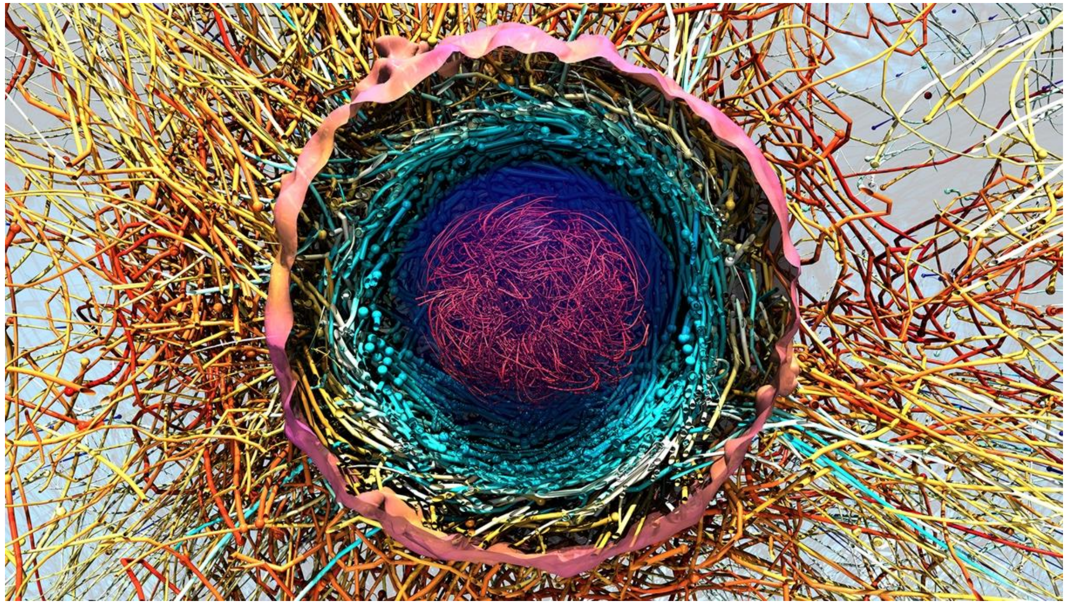
Astrophysics

PI: Adam Burrows, Princeton University

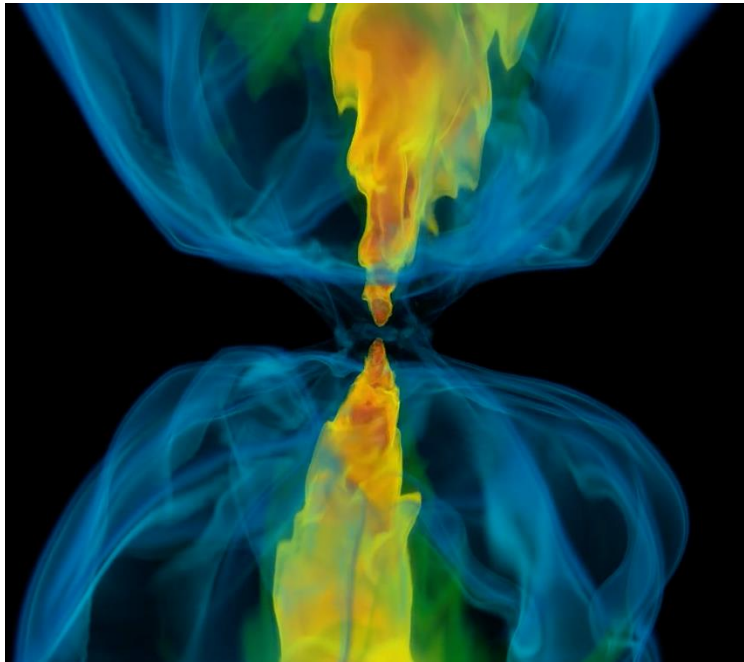


2019

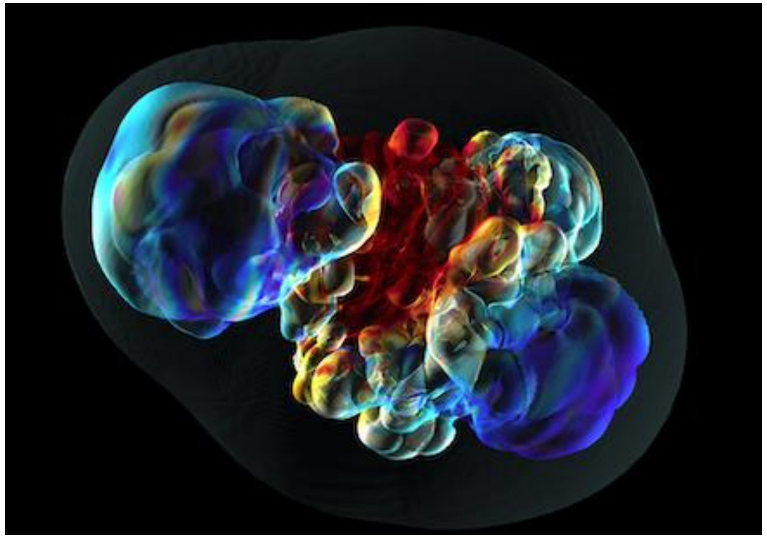
2021



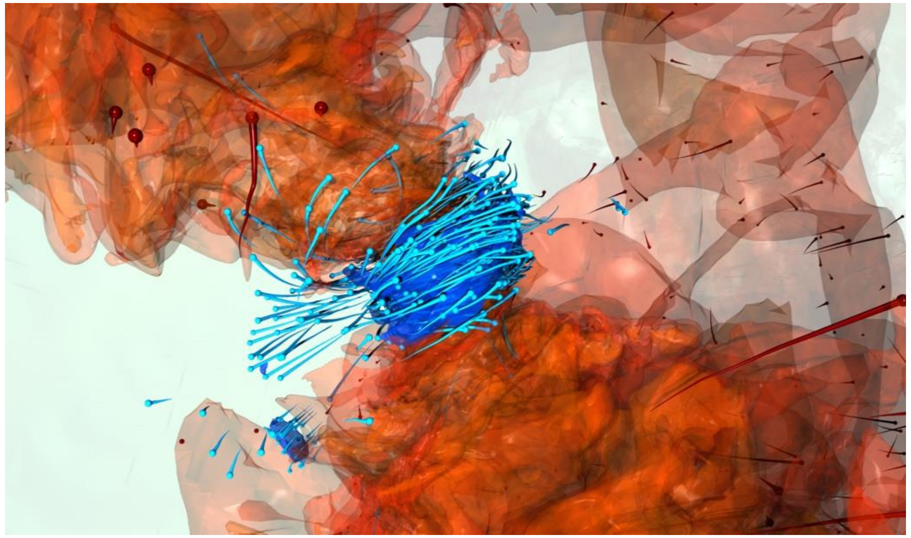
2026



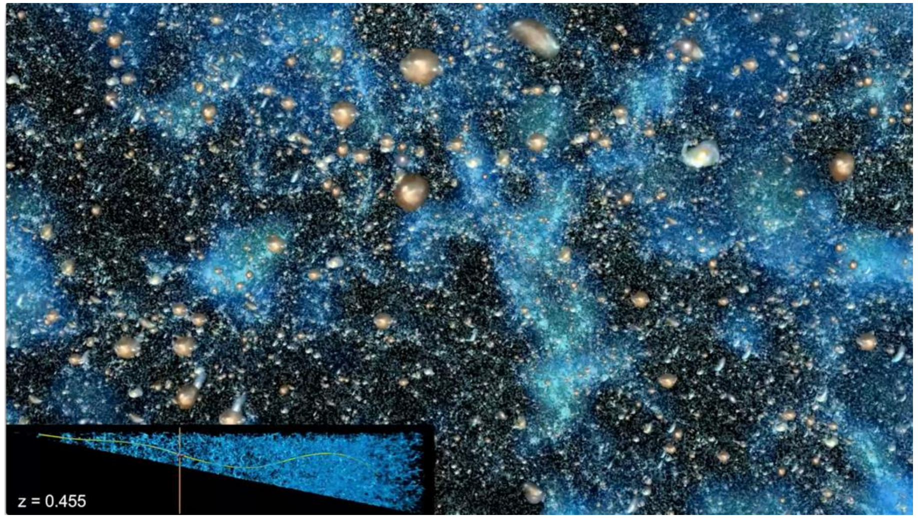
2022



2023

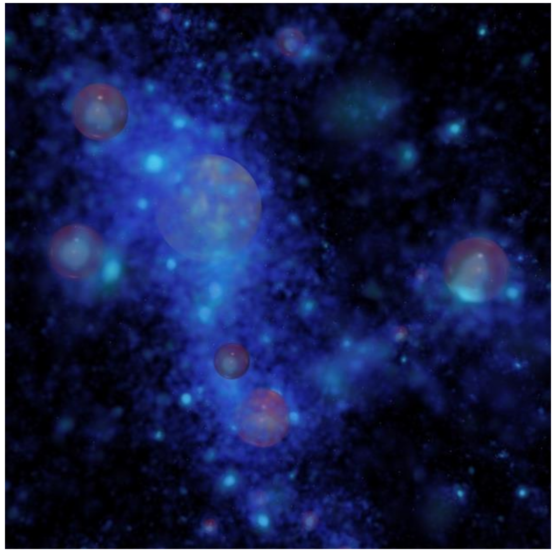


HACC: Cosmology

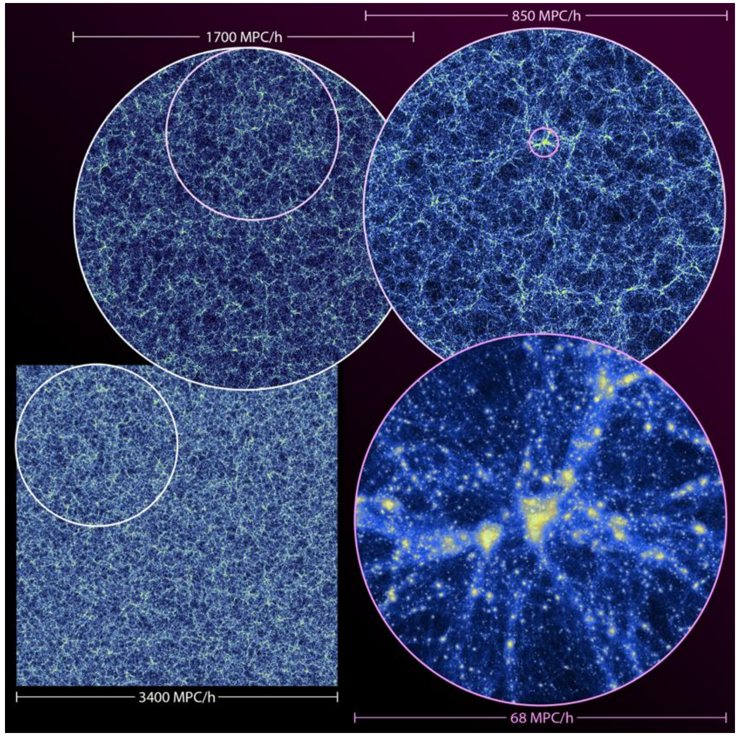


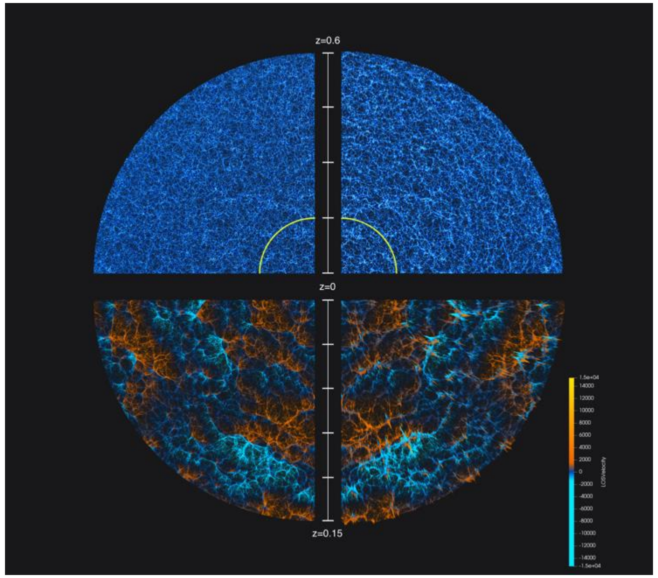
2018

2020

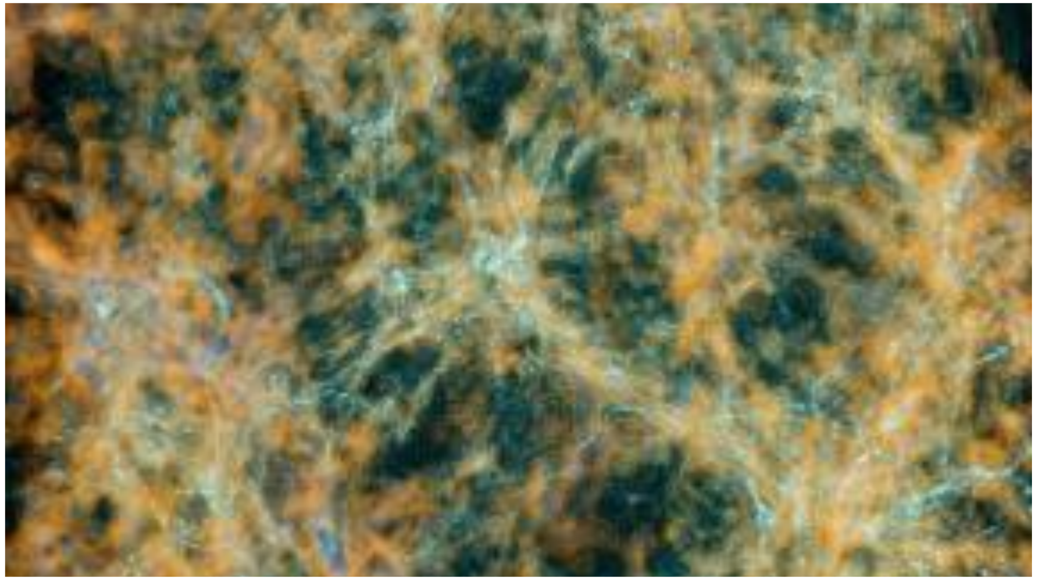


2020





2021



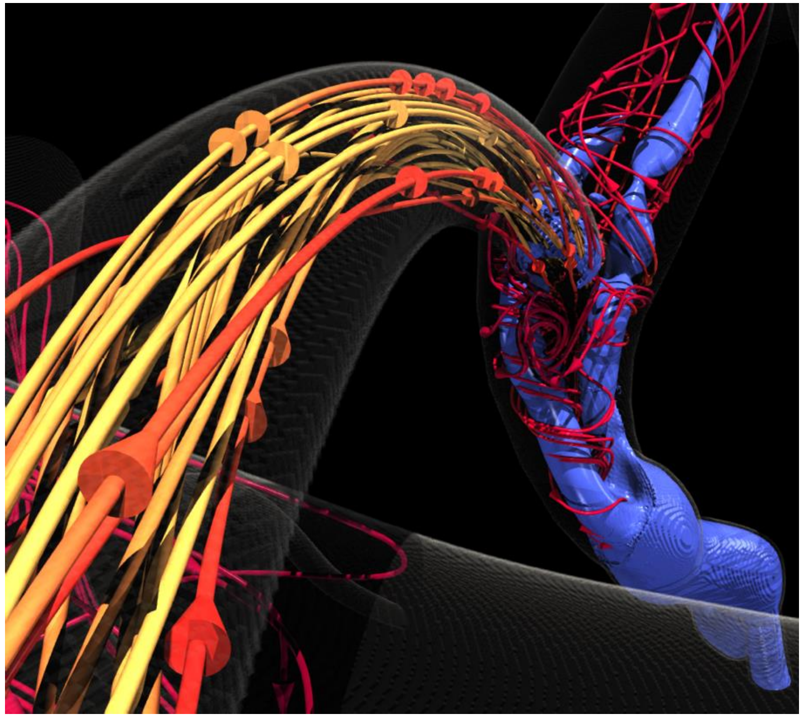
Computed and Rendered on Aurora 2023

PI: Salman Habib and HACC Team, Argonne National Laboratory

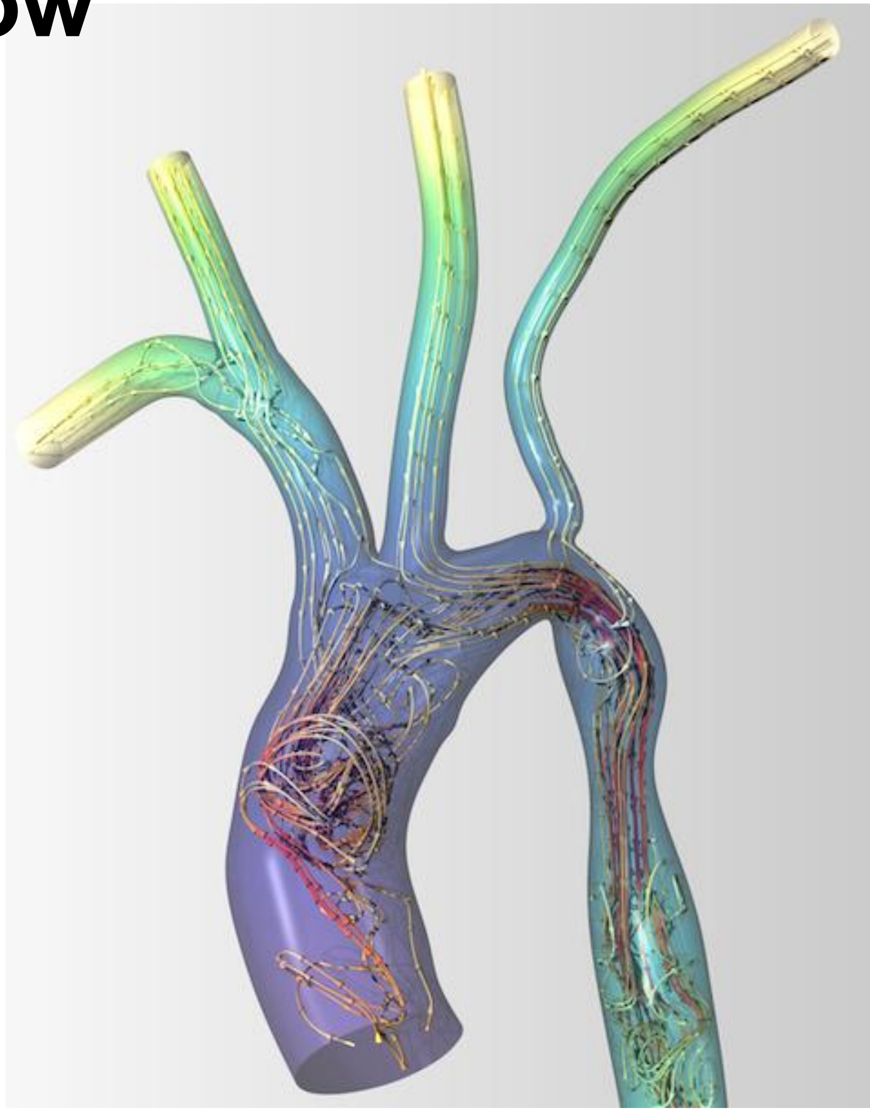
Argonne NATIONAL LABORATORY

Arterial Blood Flow

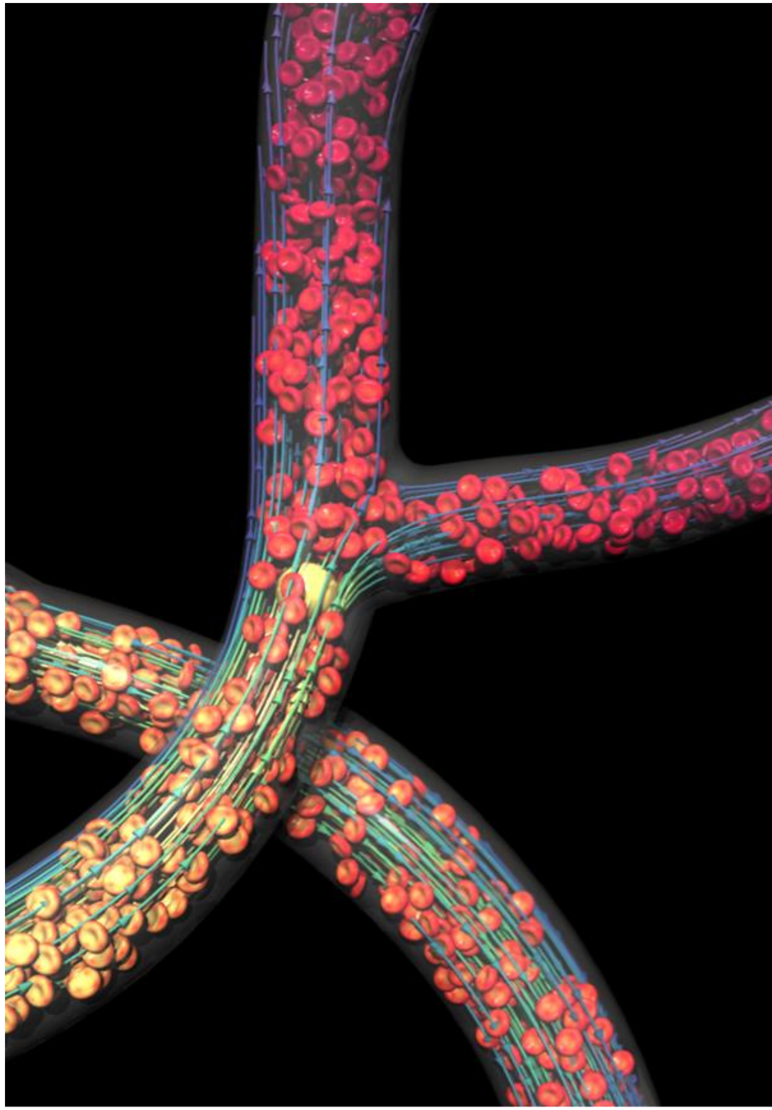
PI: Amanda Randles, Duke University



2020

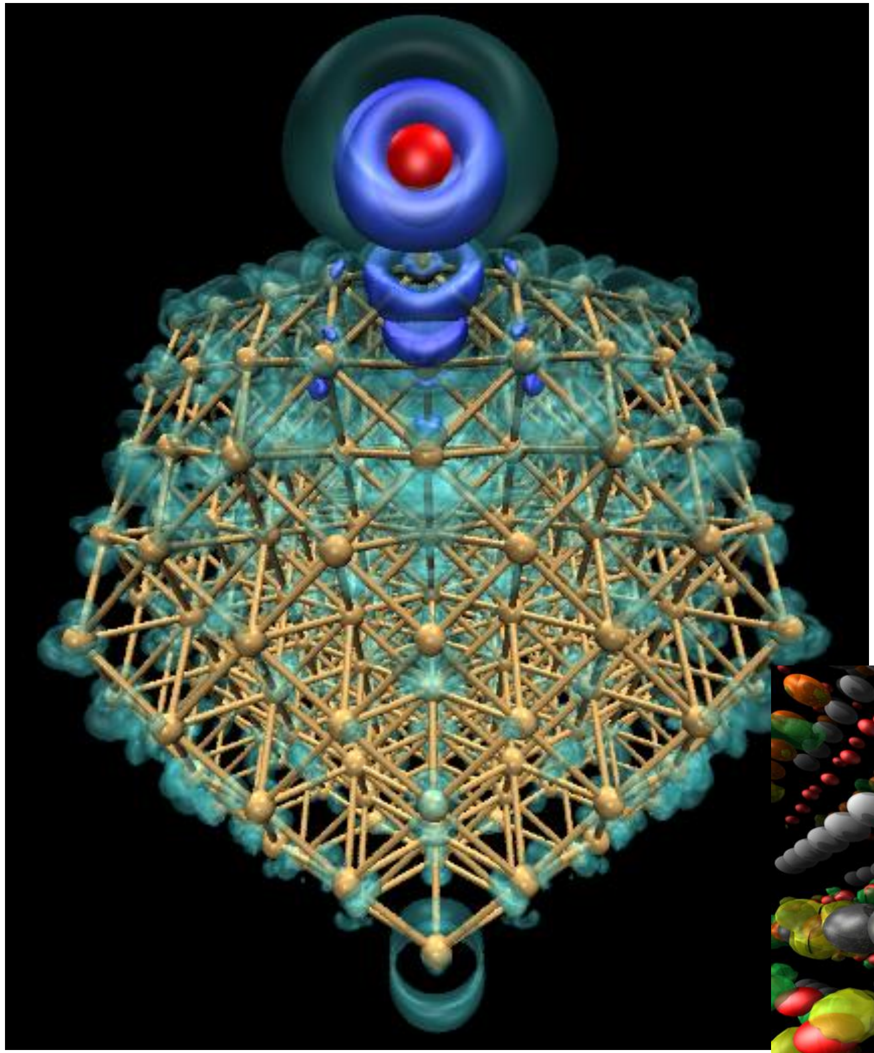


2023 **Rendered on Aurora**



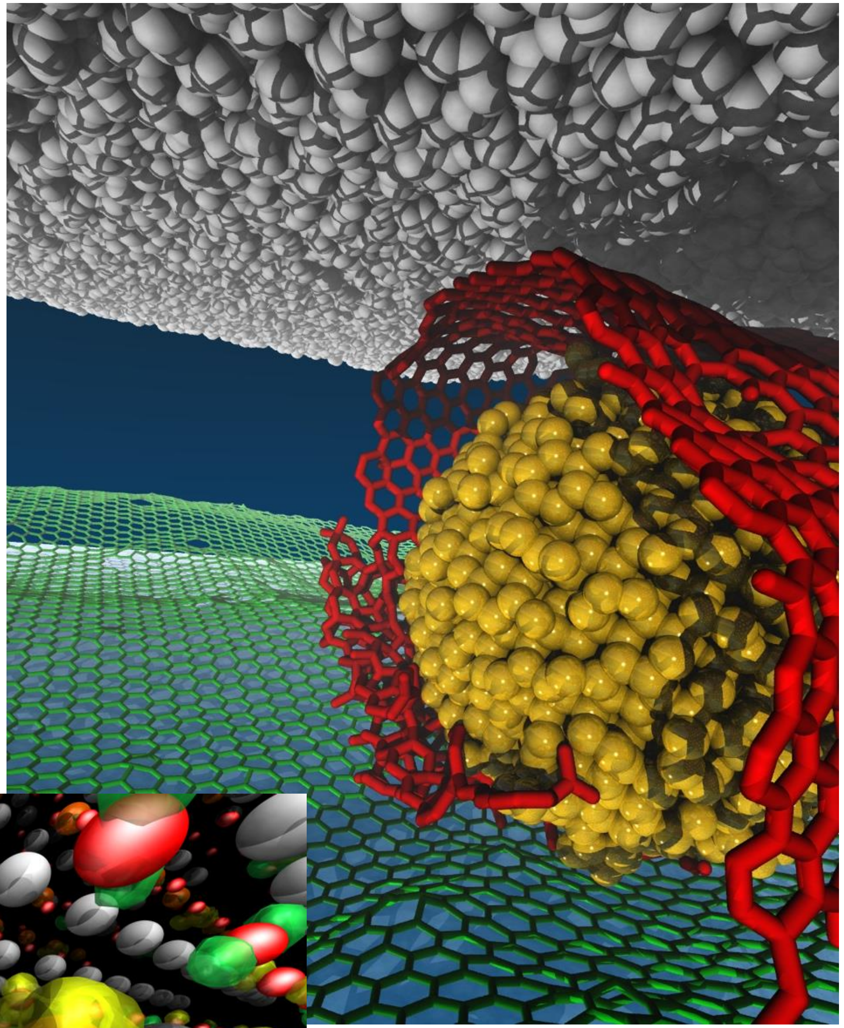
2023 **Rendered on Aurora**

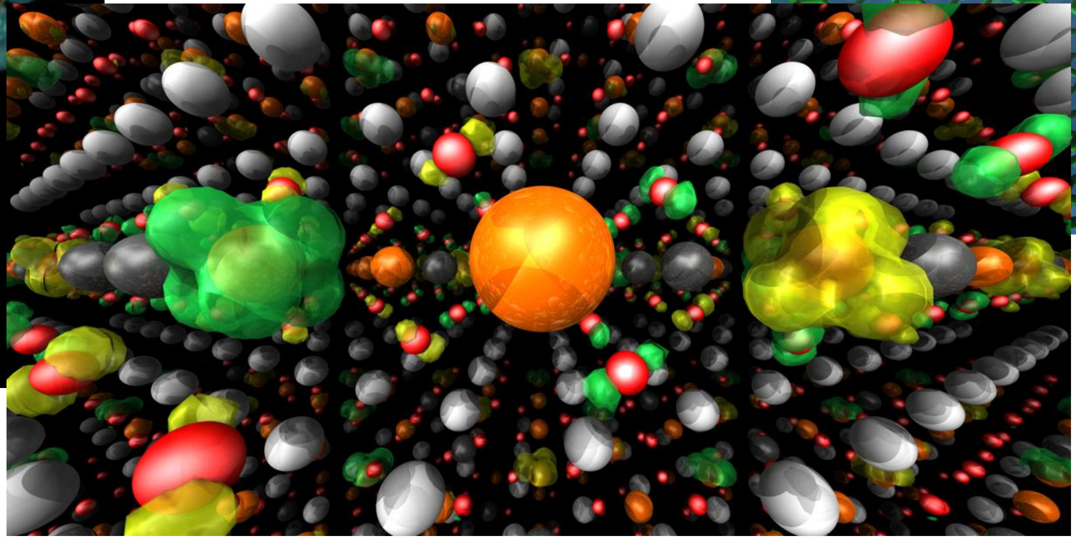
Materials Science / Molecular



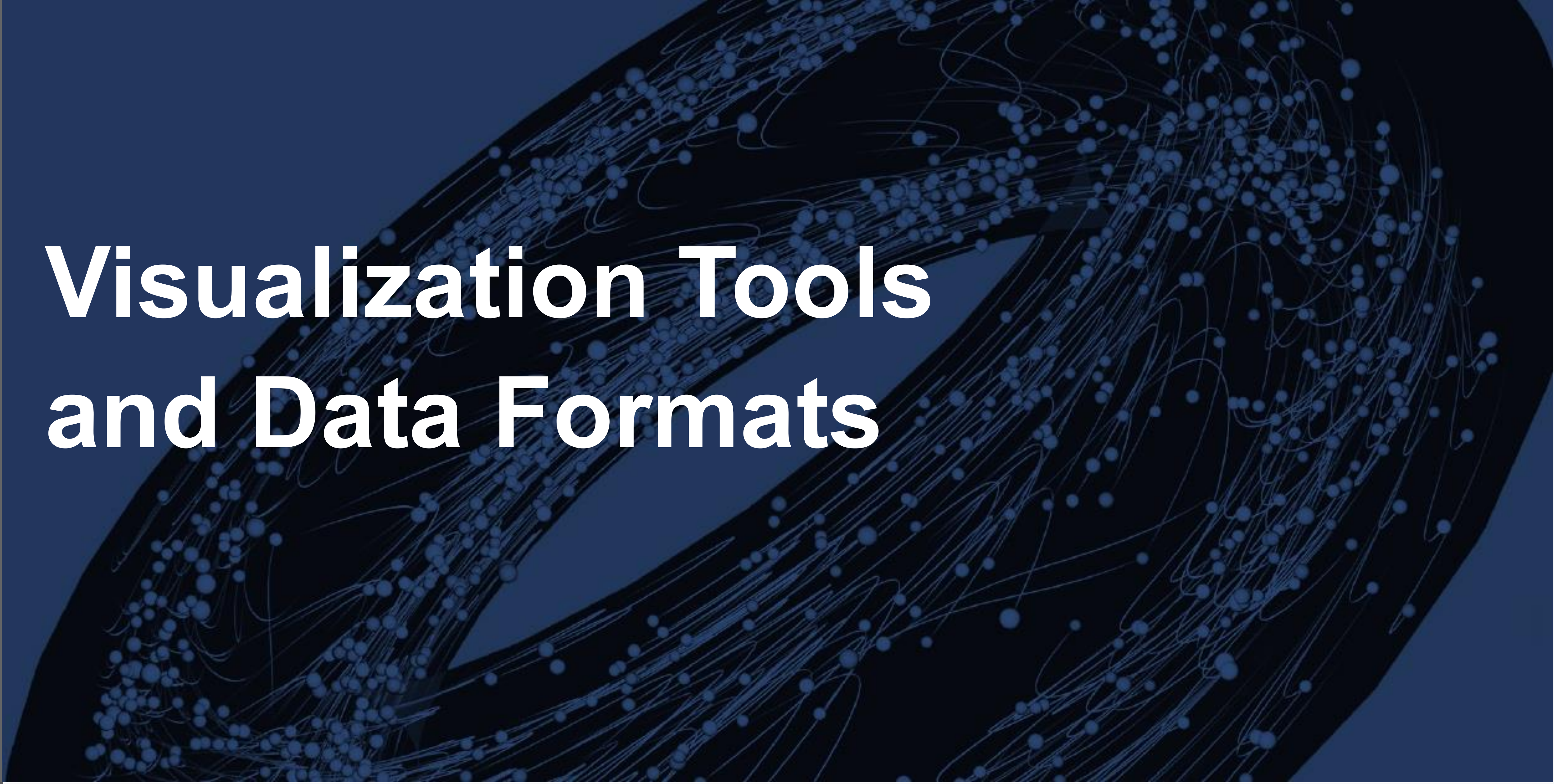
Data courtesy of: Jeff Greeley, Nichols Romero, Argonne National Laboratory

Data courtesy of: Subramanian Sankaranarayanan, Argonne National Laboratory





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



Visualization Tools and Data Formats

All Sorts of Tools

Visualization Applications
- VisIt
- ParaView
- EnSight

Domain Specific
- VMD, PyMol, Ovito, Vapor

APIs
- VTK: visualization
- ITK: segmentation & registration

Analysis Environments
- Matlab
- Parallel R

Utilities
- GnuPlot
- ImageMagick
- ffmpeg

Available on Polaris and Aurora

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)

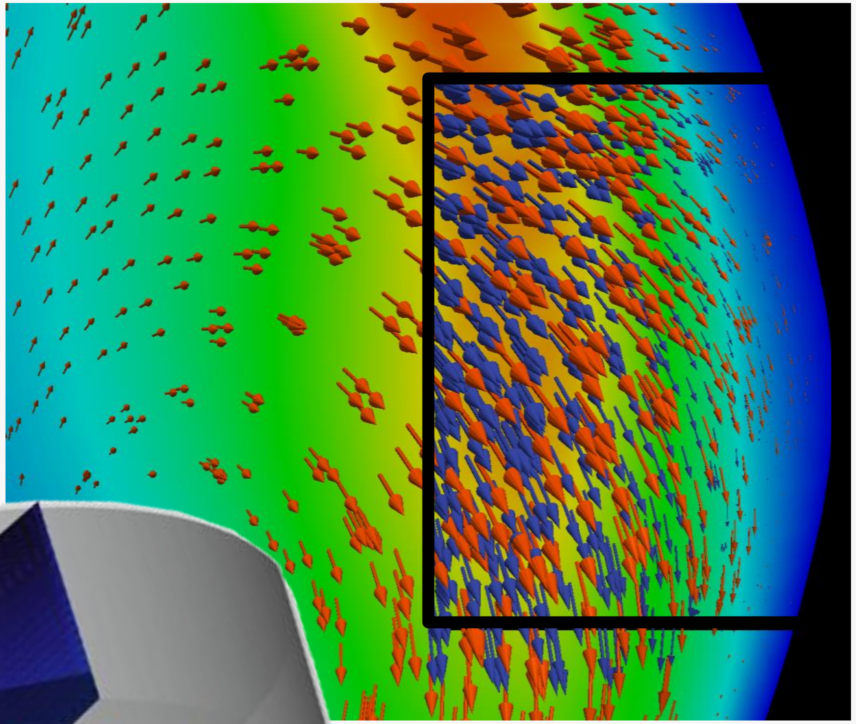
Data File Formats (ParaView & VisIt)

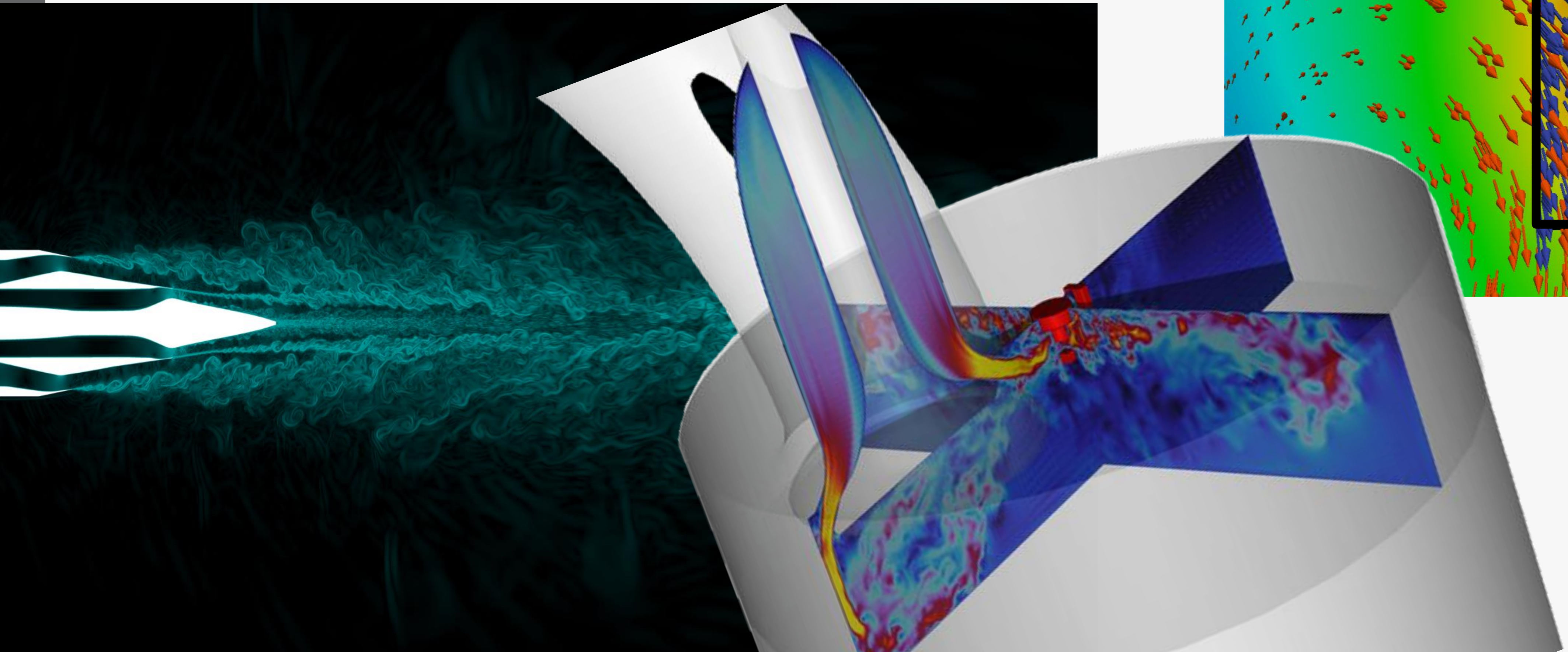
- VTK
- Parallel (partitioned) VTK
- VTK MultiBlock (MultiGroup, Hierarchical, Hierarchical Box)
- Legacy VTK
- Parallel (partitioned) legacy VTK
- EnSight files
- EnSight Master Server
- Exodus
- BYU
- XDMF
- PLOT2D
- PLOT3D
- SpyPlot CTH
- HDF5 raw image data
- DEM
- VRML
- PLY
- Polygonal Protein 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

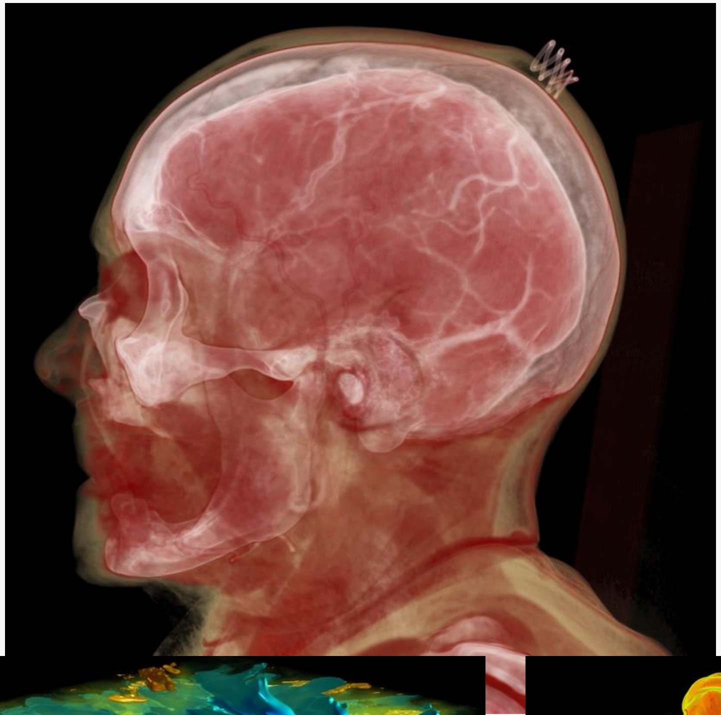
Data Representations: Cutting Planes

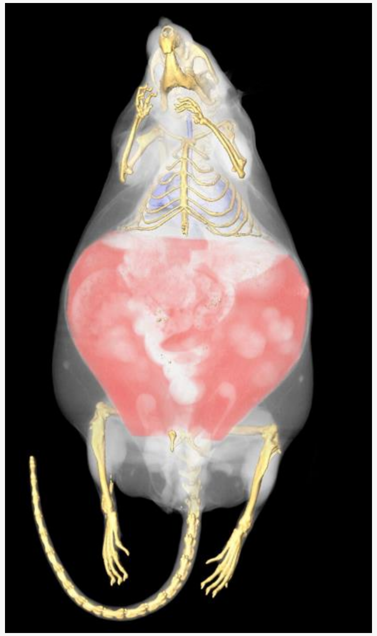
- Slice a plane through the data
  - – Can apply additional visualization methods to resulting plane
- VisIt & ParaView & vtk good at this
- VMD has similar capabilities for some data formats

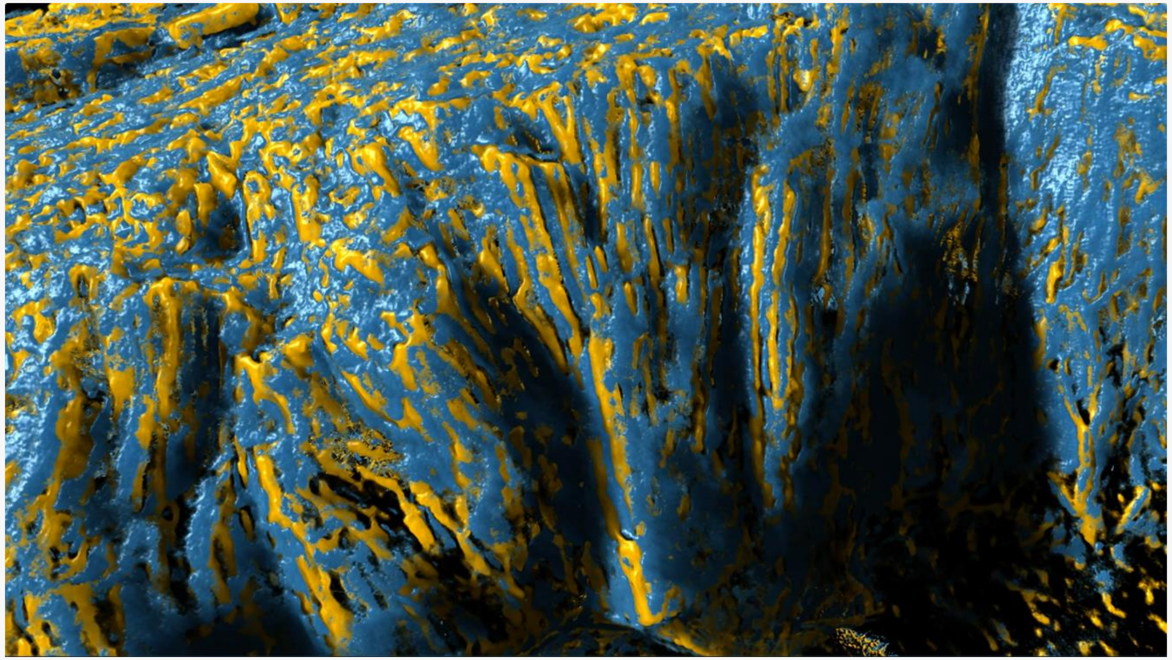


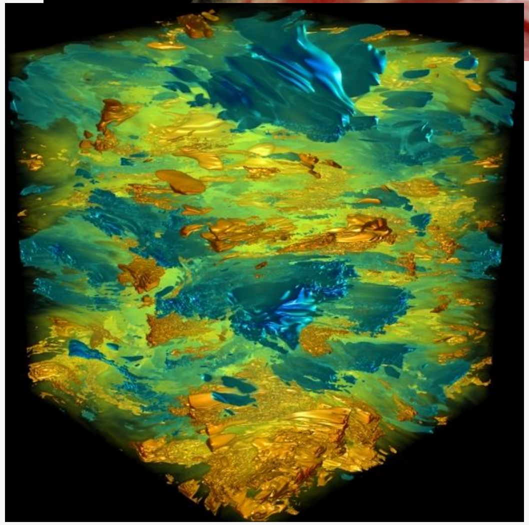


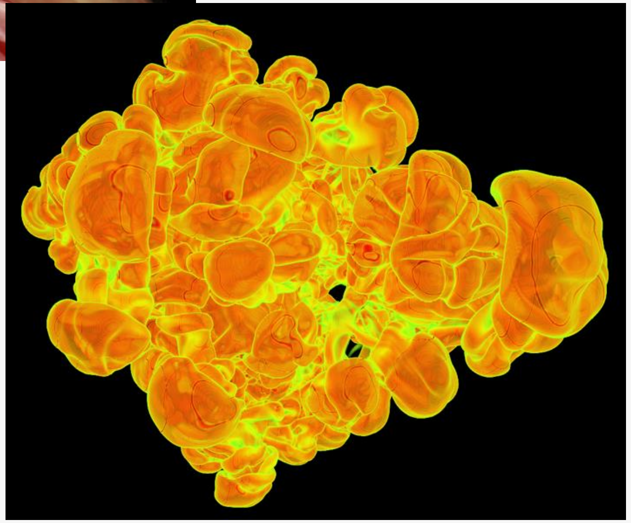
Data Representations: Volume Rendering

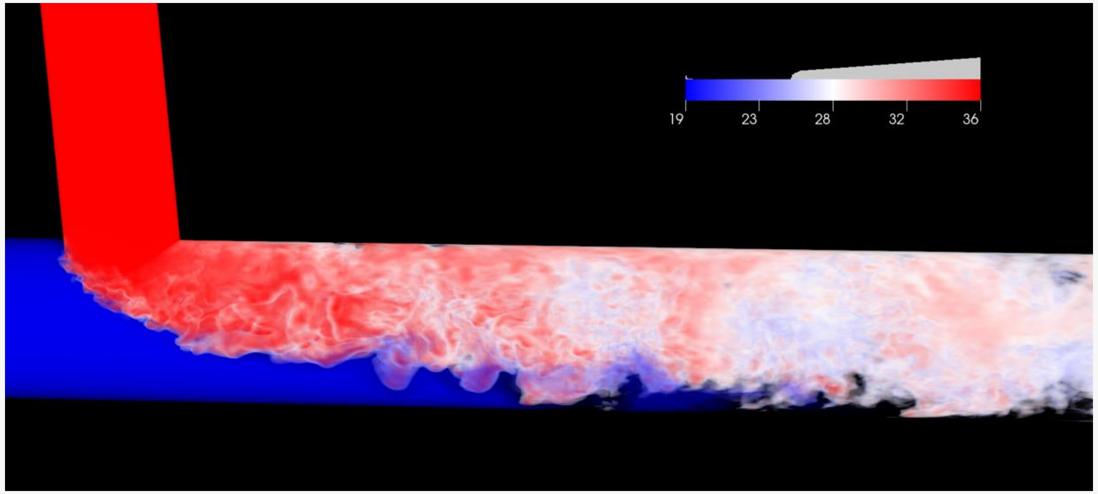












Data Representations: Contours (Isosurfaces)

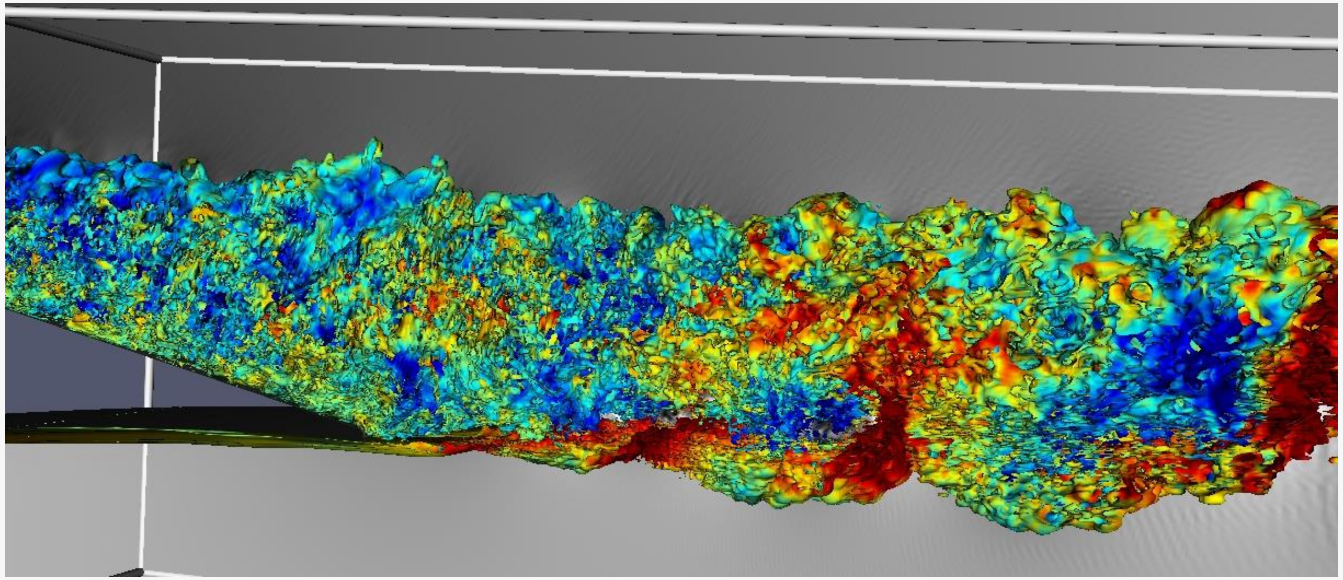
A Line (2D) or Surface (3D), representing a constant value

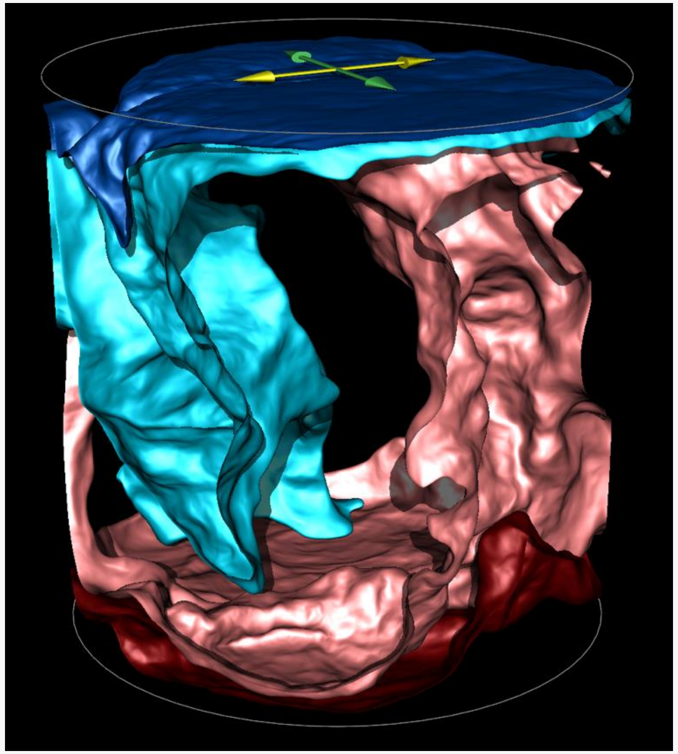
VisIt & ParaView:

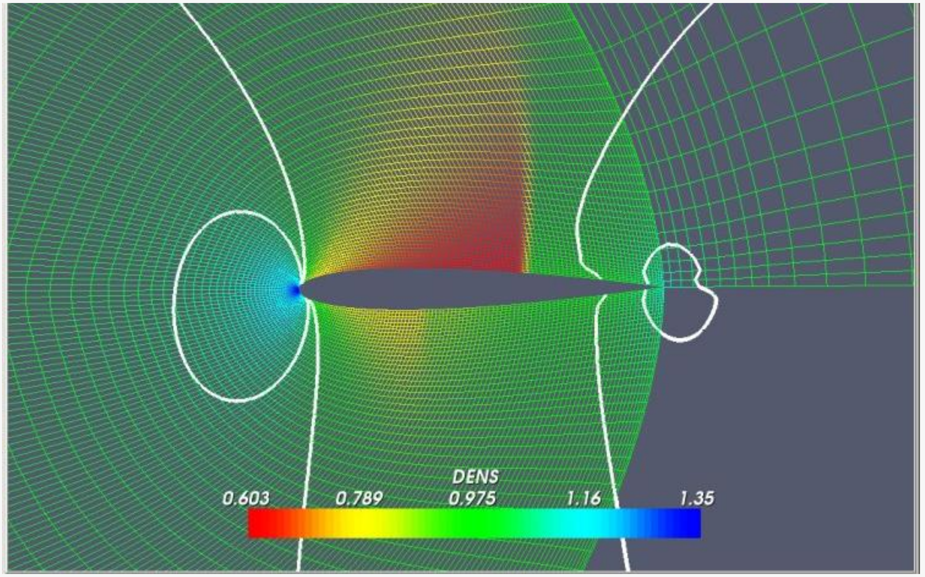
– good at this

vtk:

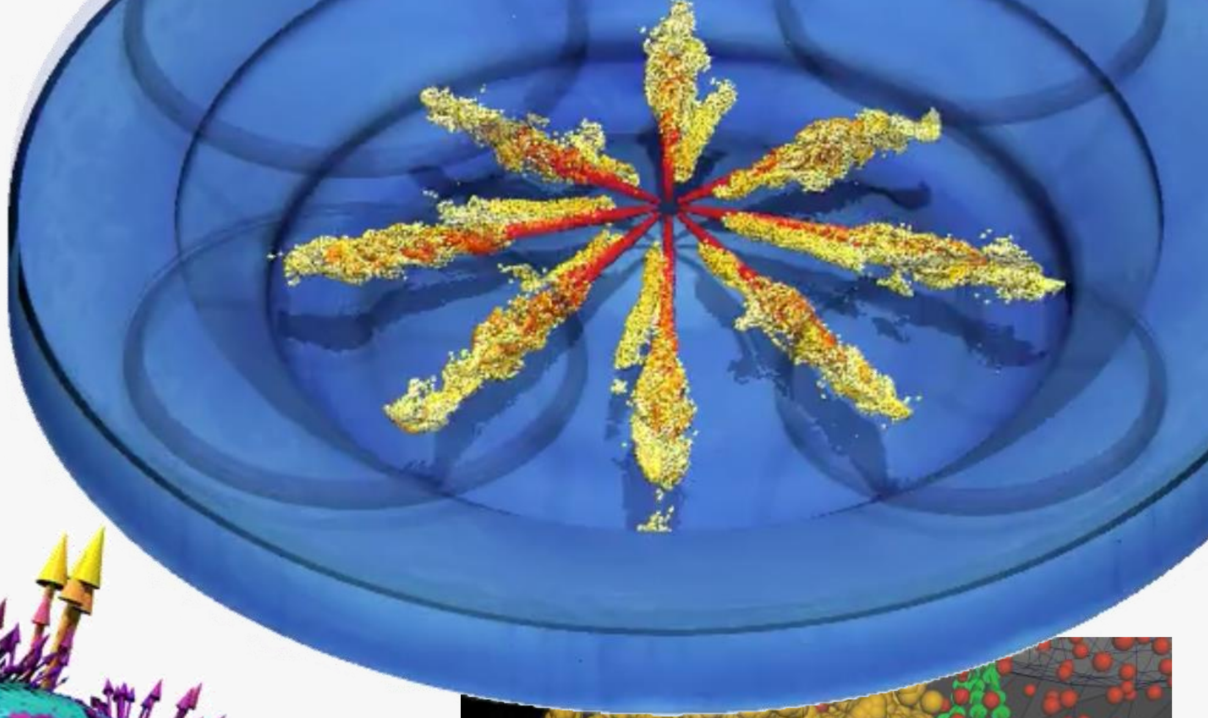
– same, but again requires more effort







Data Representations: Glyphs



2D or 3D geometric object to represent point data

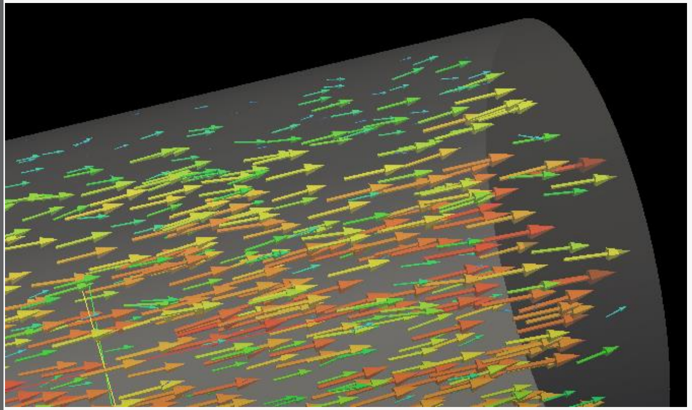
Location dictated by coordinate

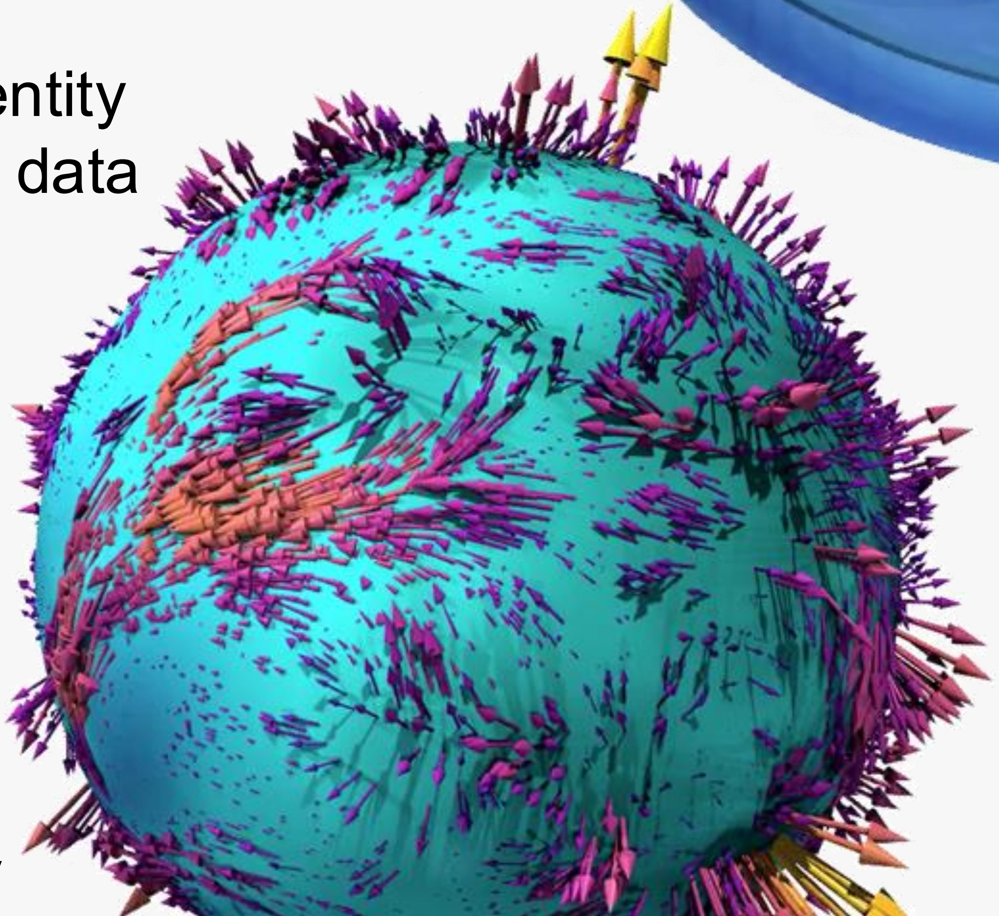
– 3D location on mesh

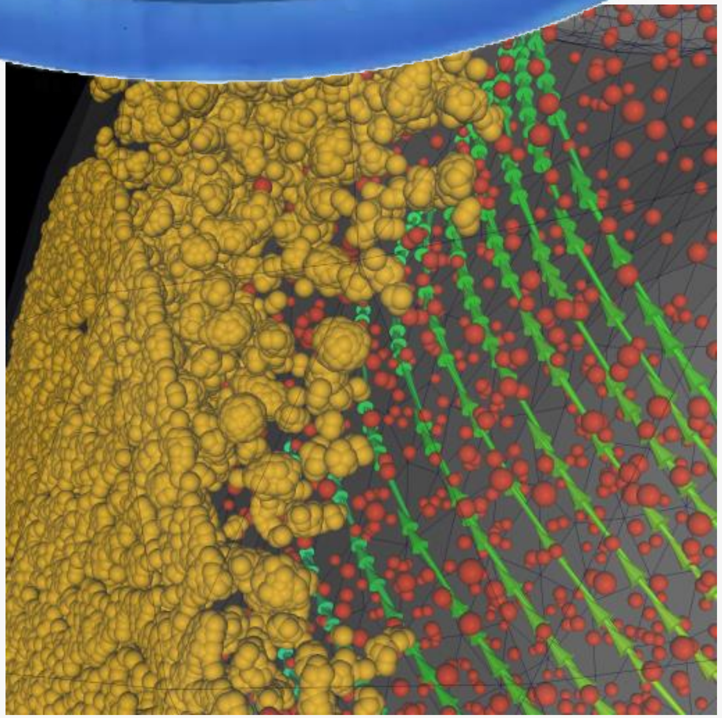
– 2D position in table/graph

Attributes of graphical entity dictated by attributes of data

– color, size, orientation





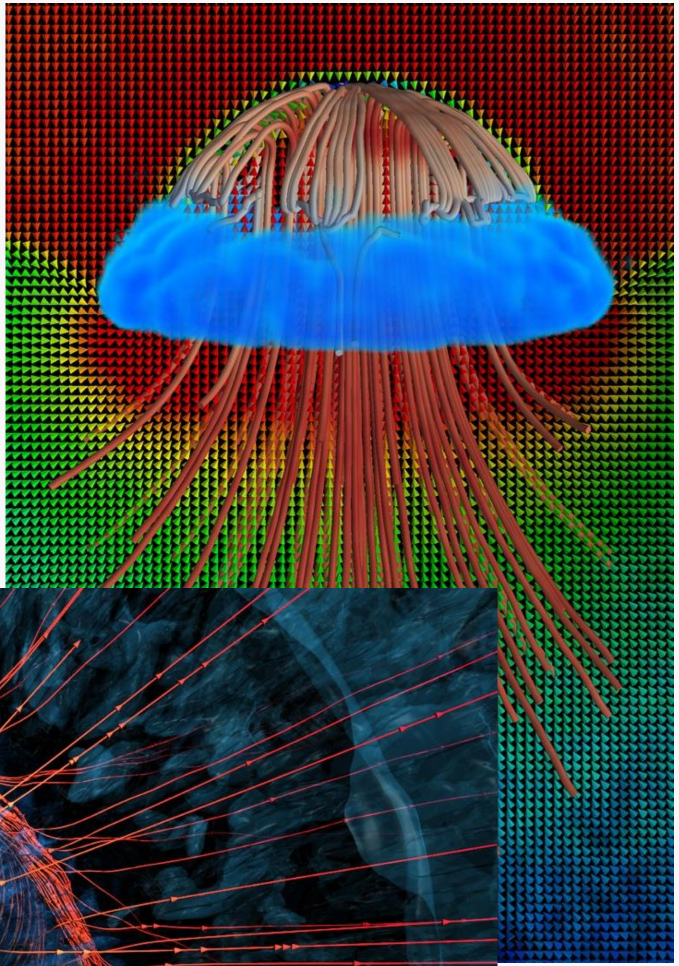


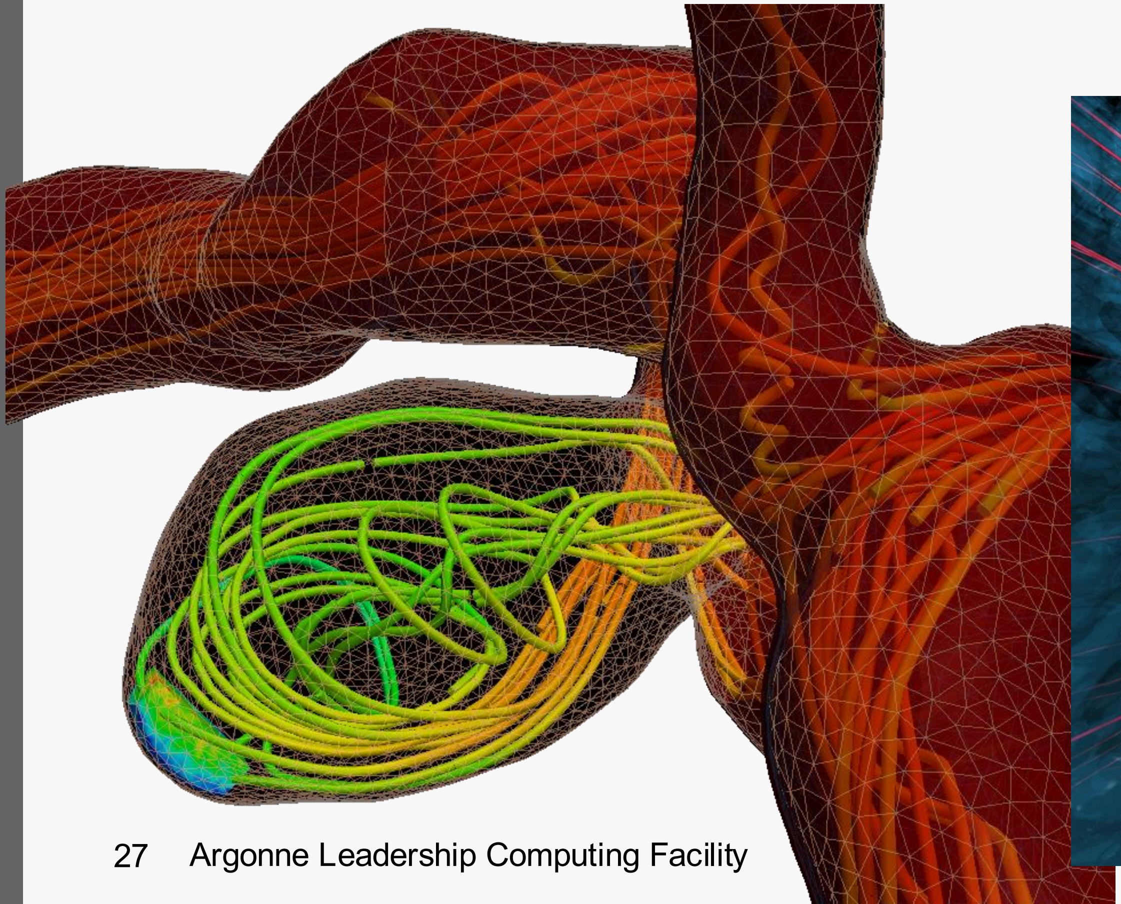
Data Representations: Streamlines

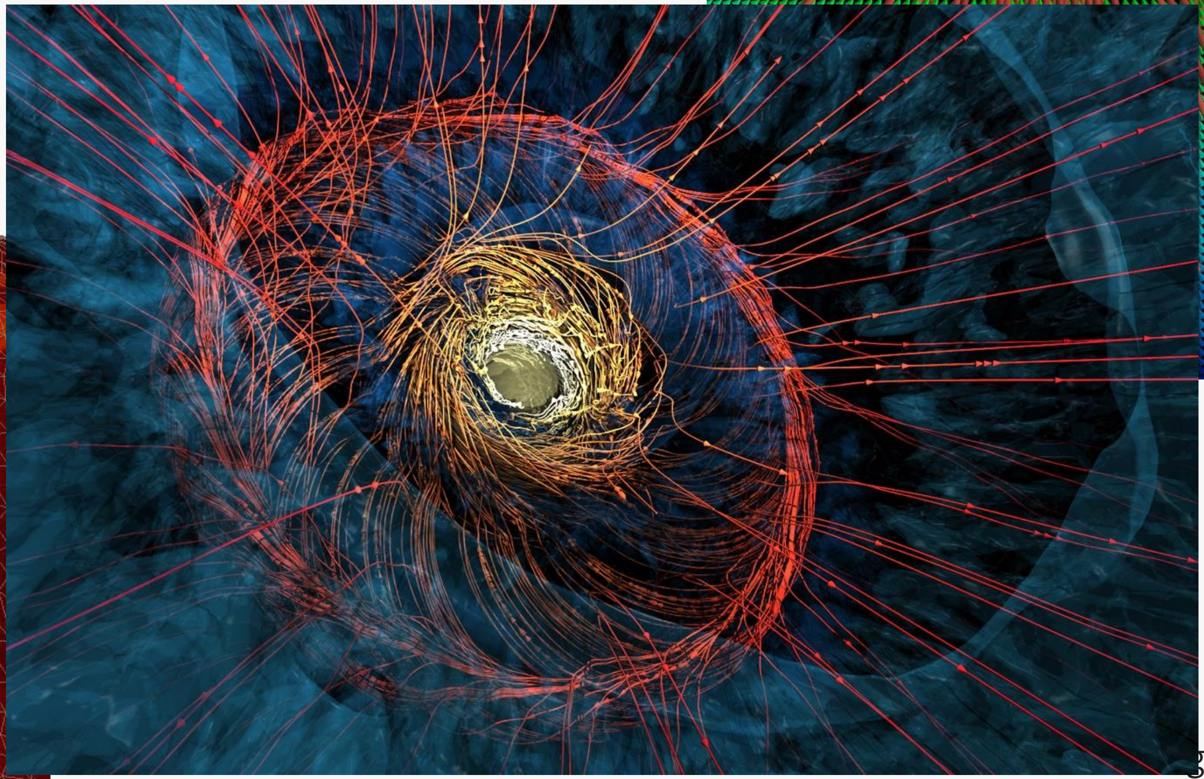
From vector field on a mesh (needs connectivity)

– 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/Ovito:

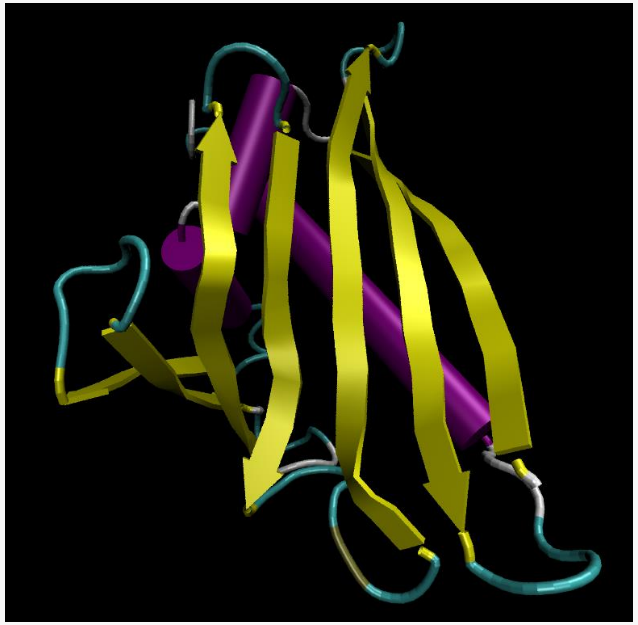
– Lots of domain-specific representations
– Many different file formats
– Animation
– Scriptable

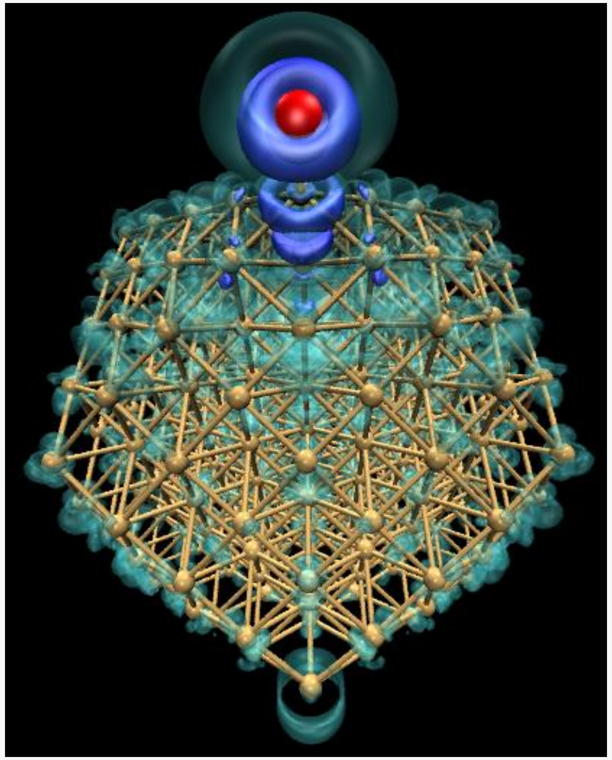
VisIt & ParaView:

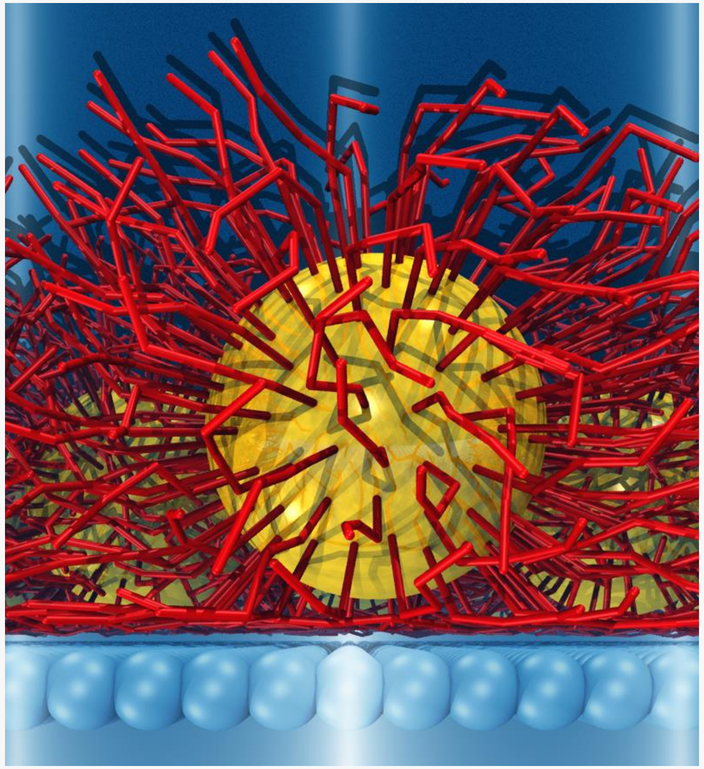
– Limited support for these types of representations, but improving

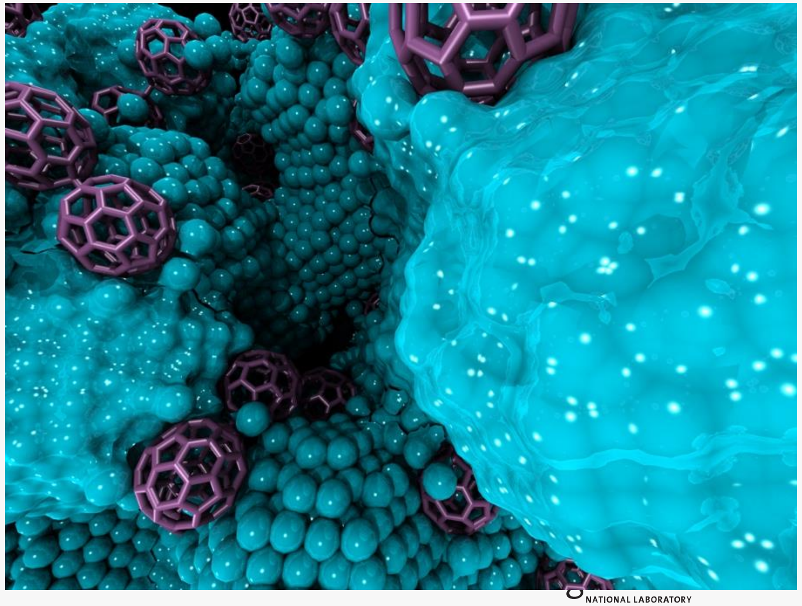
VTK:

– Anything's possible if you try hard enough



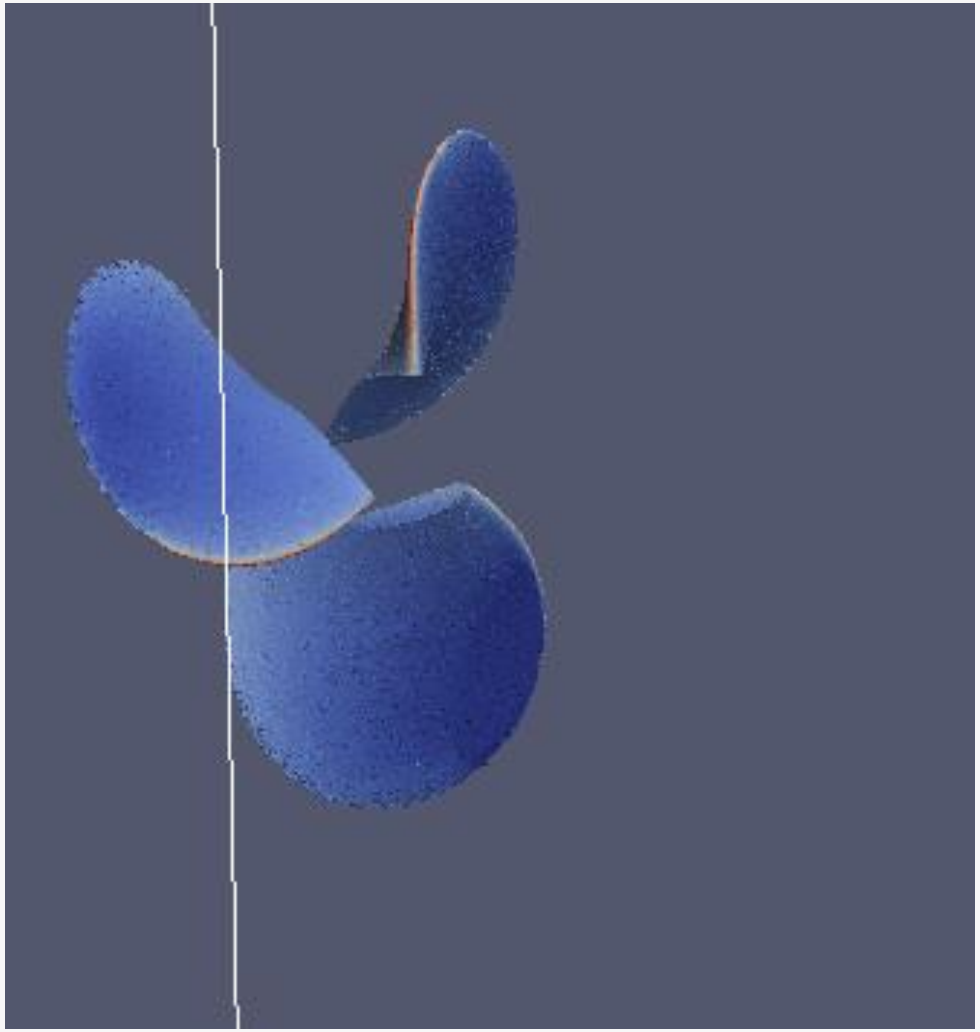




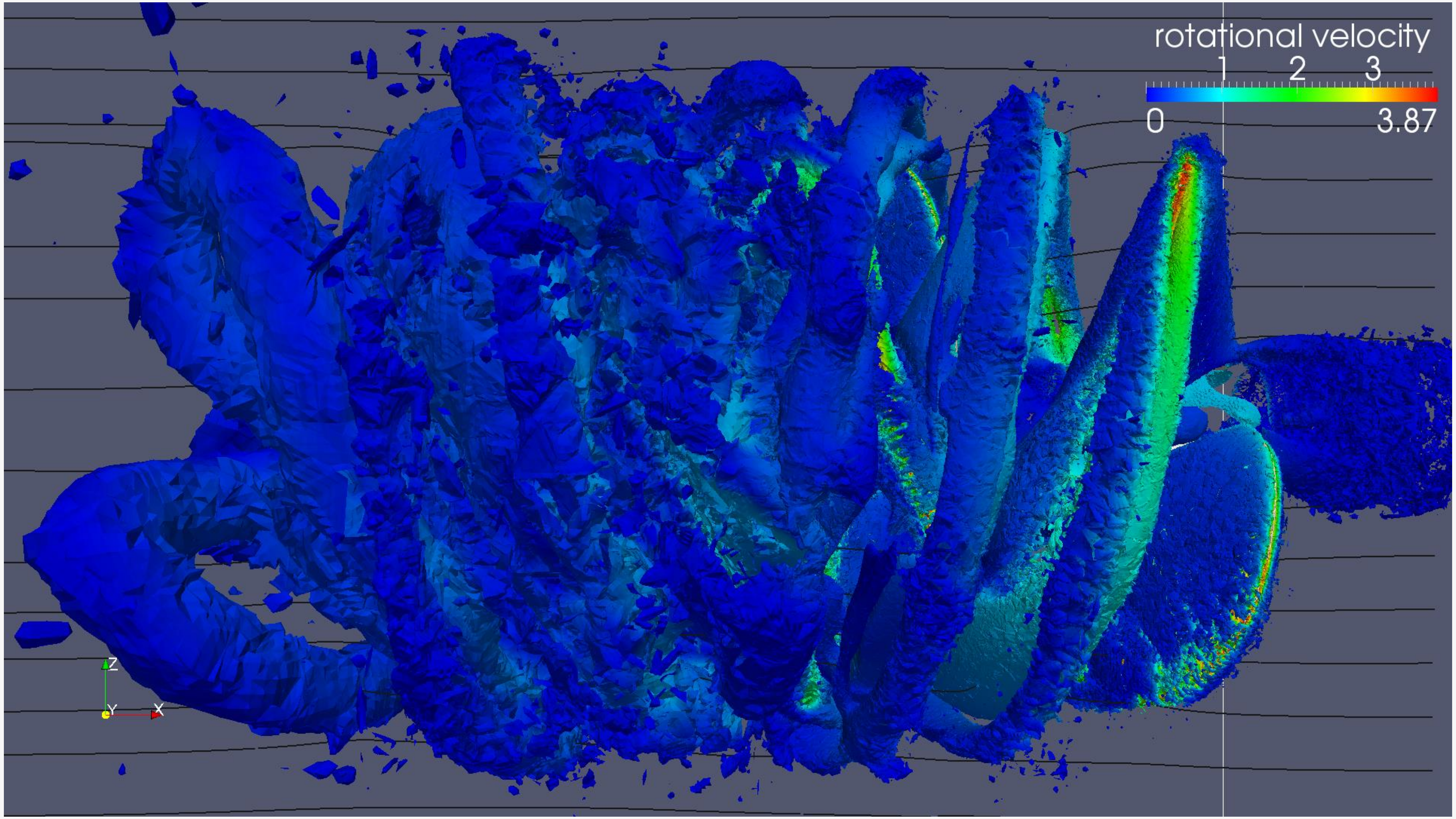


Visualization for Debugging

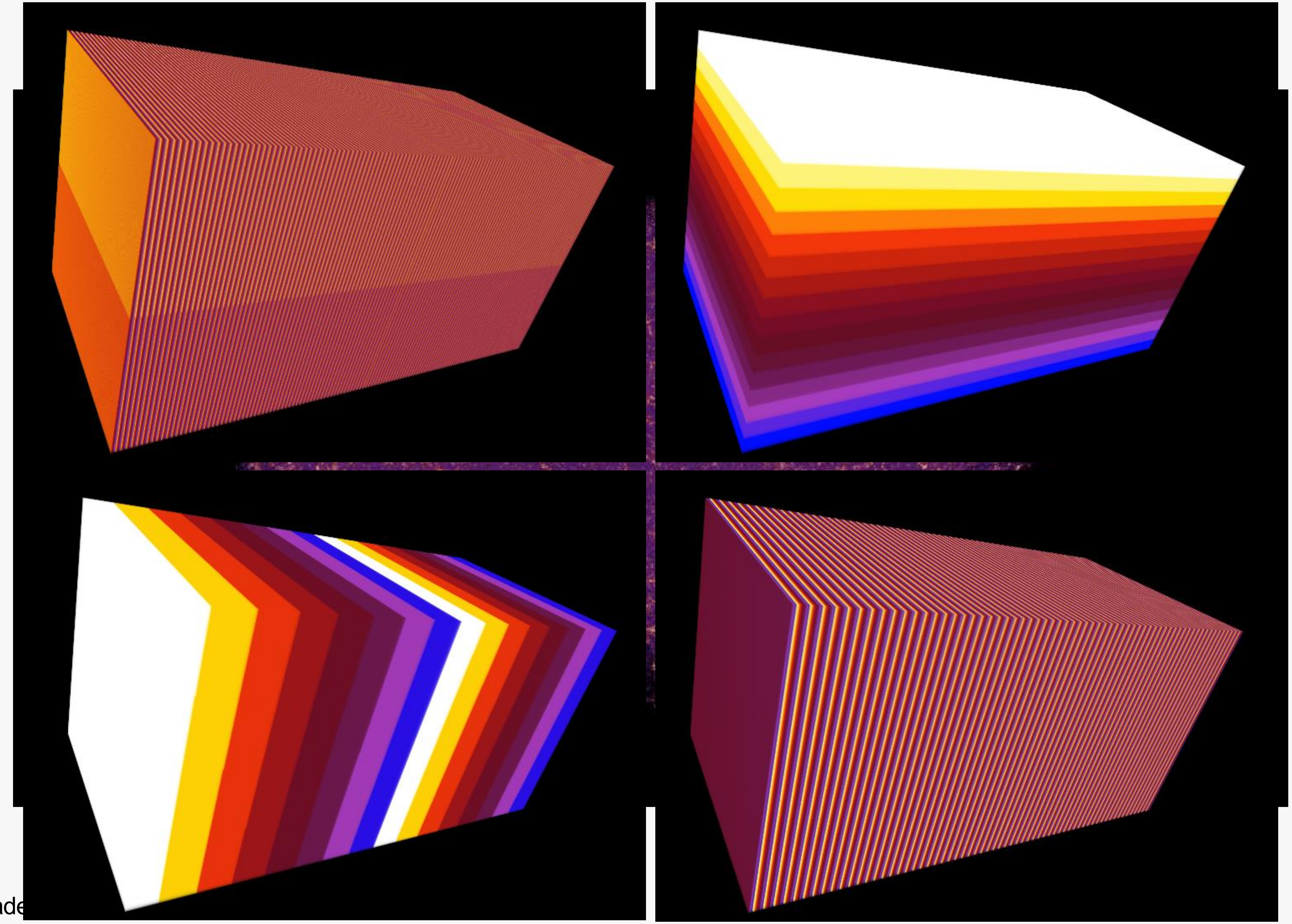
Visualization for Debugging



Visualization for Debugging



Visualization as Diagnostics: Color by Thread ID



In Situ Visualization and Analysis

In Situ vis and Analysis Problem:

FLOPS to I/O Bottleneck

– Frontier
  - Peak Performance: 1.6 EF
  - Storage: 2-4x Summit's I/O 2.5TB/s. At best 10TB/s
  - 5 orders of magnitude difference

– Aurora
  - Peak Performance: 1.012 EF
  - Storage: 31TB/s
  - 5 orders of magnitude difference

Problem

- I/O is too expensive
- Scientists cannot save every timestep, and/or resolution
- Lost cycles: simulation waits while I/O is happening
- Lost science: scientists might miss discoveries

Solution: *In situ* visualization and analysis

What is *IN SITU* vis and analysis

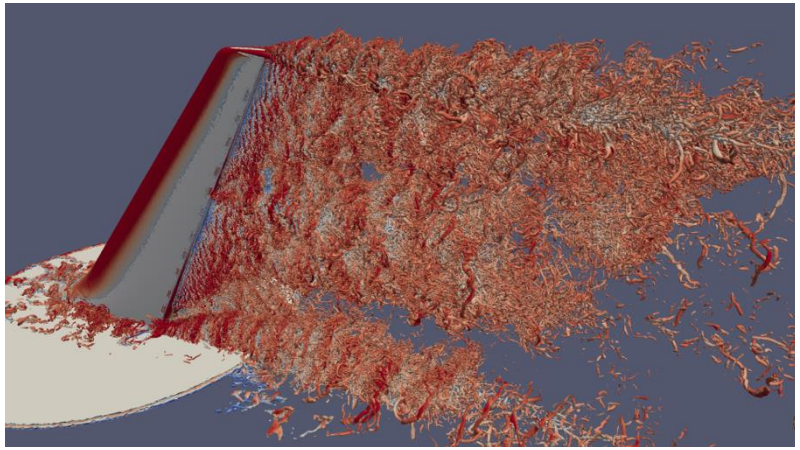
Traditionally visualization and analysis happens post hoc

- aka: Data gets saved to the disk, scientist opens it after the simulation has ended

In situ

- Data gets visualized/analyzed **while** in memory.
- If zero-copy used, there is no data movement
- Ideally the data is on the GPU and stays on the GPU

In Situ

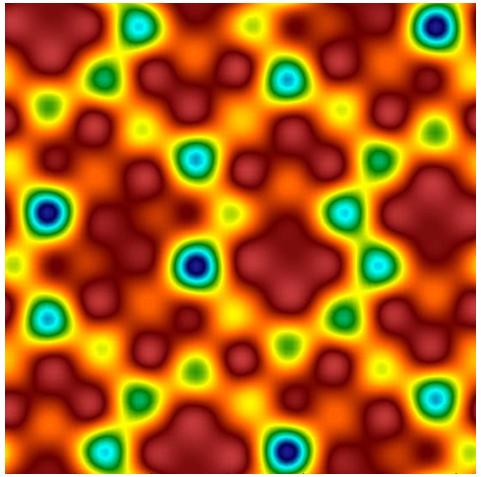


~2014

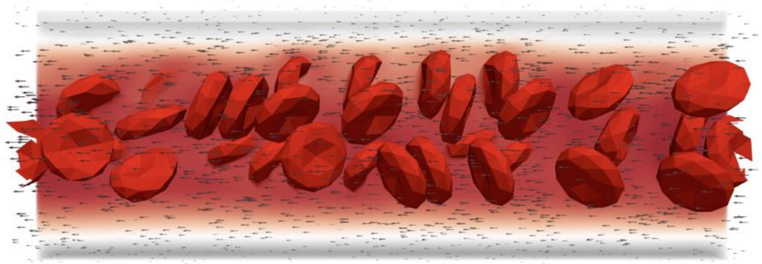
PHASTA, Catalyst, Ken Jansen

2018

Nek5000, SENSEI



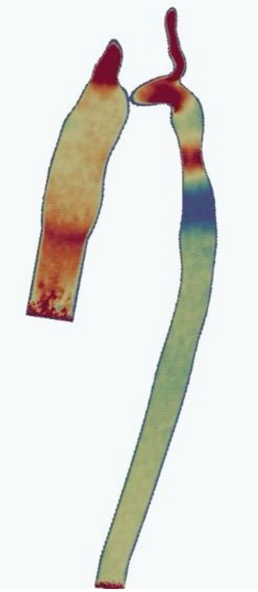
2021 - 2024



Palabos+LAMMPS, SENSEI + Catalyst, bi-directional

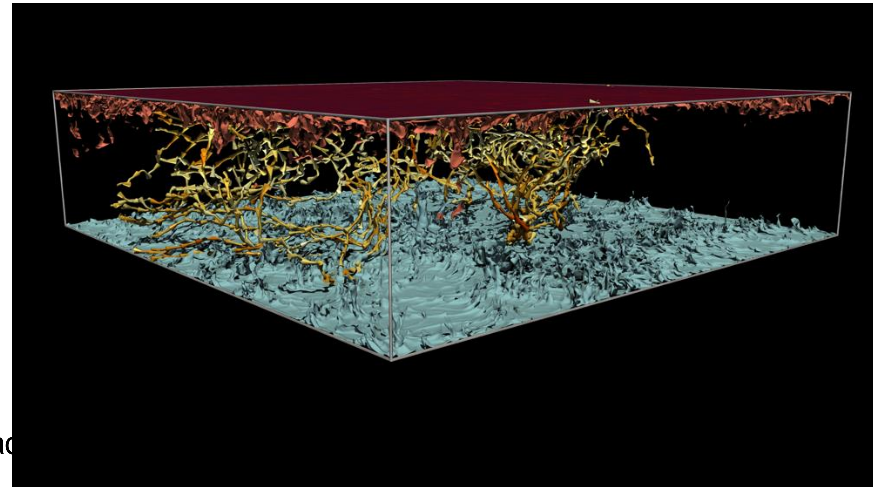
2019

SENSEI + Catalyst



2024

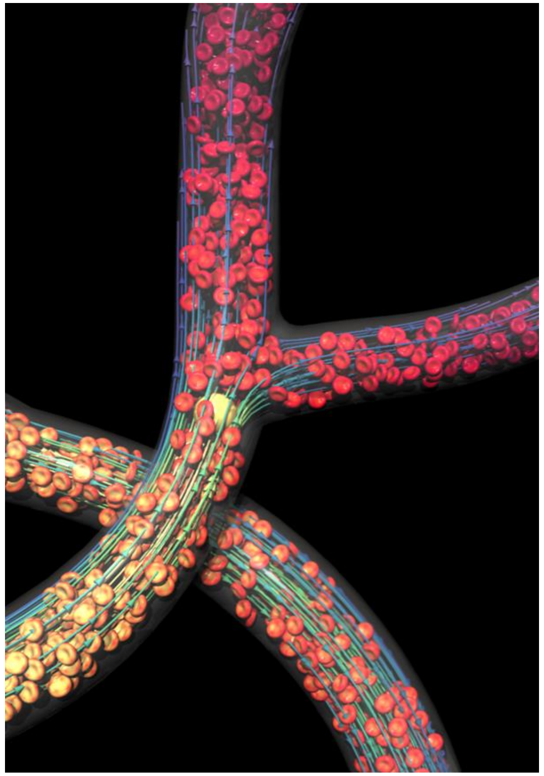
nekRS, Ascent + Catalyst



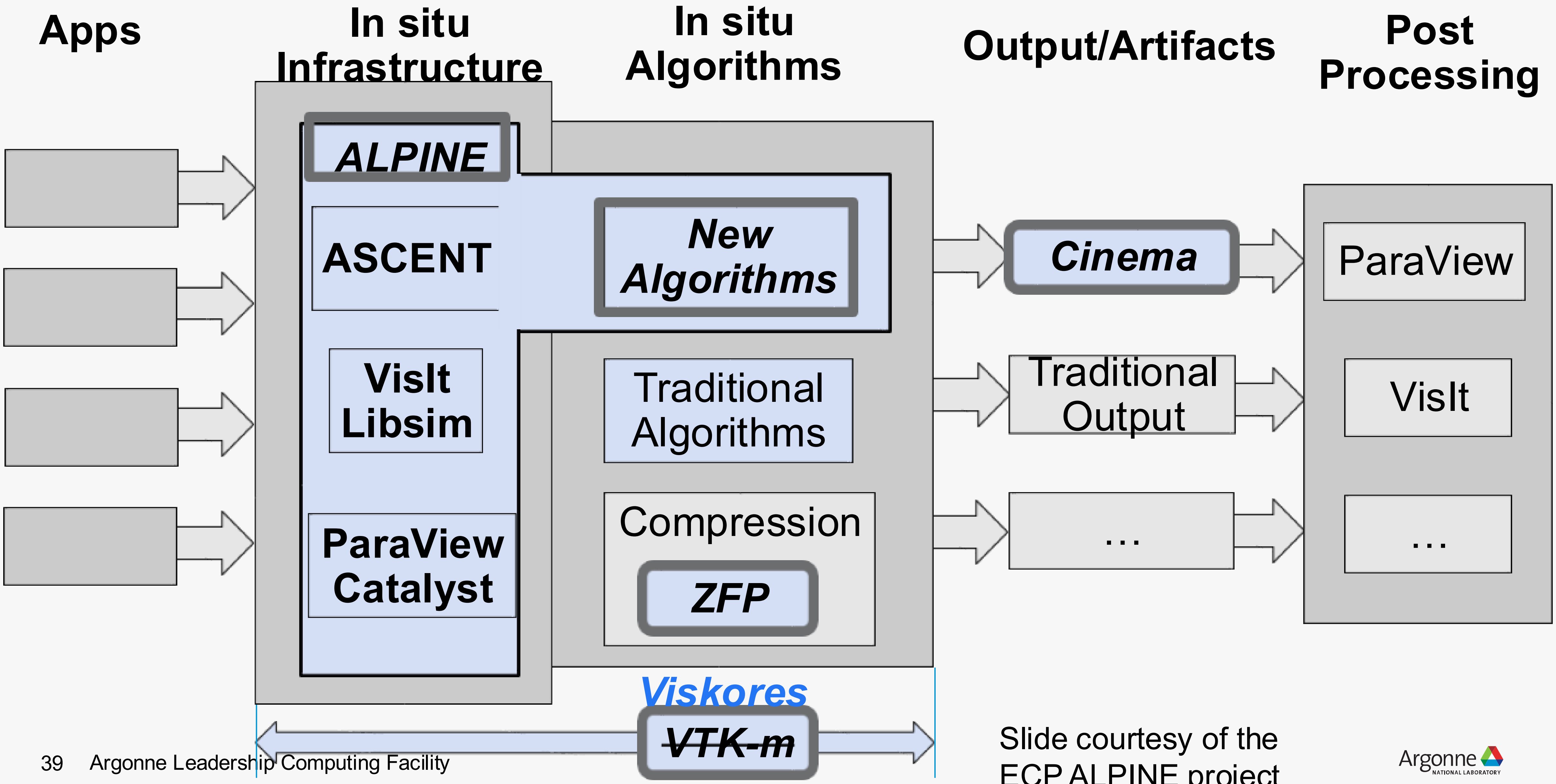
HARVEY

Ascent + Catalyst

2024



Exascale Computing Project
Software Technology Data and Visualization



Slide courtesy of the ECP ALPINE project

In situ infrastructures at ALCF

Ascent

- —GPU->GPU zero copy
- —Lightweight and fast

How?

- —You will need to instrument code with ascent
- —Yaml file for scene/analysis/rendering definition

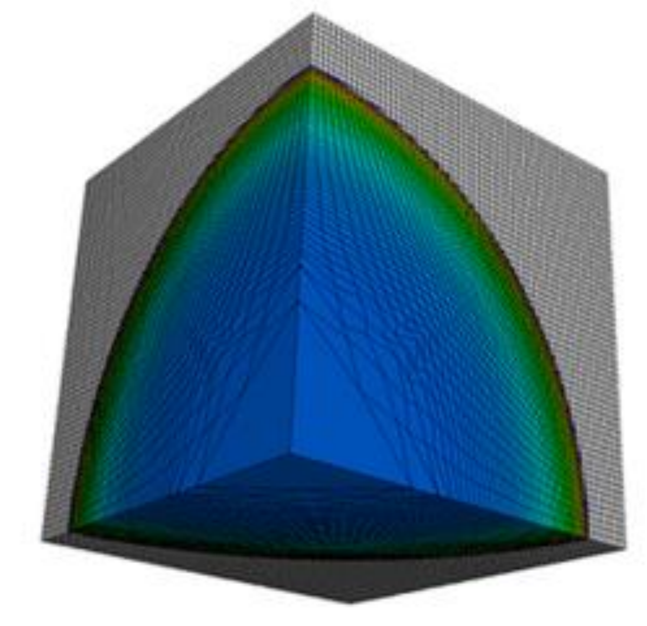
Features

- – Triggers (render only when velocity > 10)
- – Python code execution
- – ML workflows
- – Rendering workflows
- – Bi-directional workflows

Catalyst 2.0 : Rethinking the ParaView in situ data analysis and visualization API

- inspired by MPICH ABI compatibility initiative
- leverages Conduit

- ***Catalyst***: a new project with ***no*** external dependencies: https://gitlab.kitware.com/paraview/catalyst
- Gitlab-CI enabled, ReadTheDocs-enabled
- Provides API ***specification***, ***stub*** implementation, ***SDK*** for developing adaptors and custom implementations



Lulesh, a miniapp representing a typical hydrodynamics simulation code, instrumented with Catalyst

Source: https://gitlab.kitware.com/paraview/catalyst
Docs: https://catalyst-in-situ.readthedocs.io/en/latest/

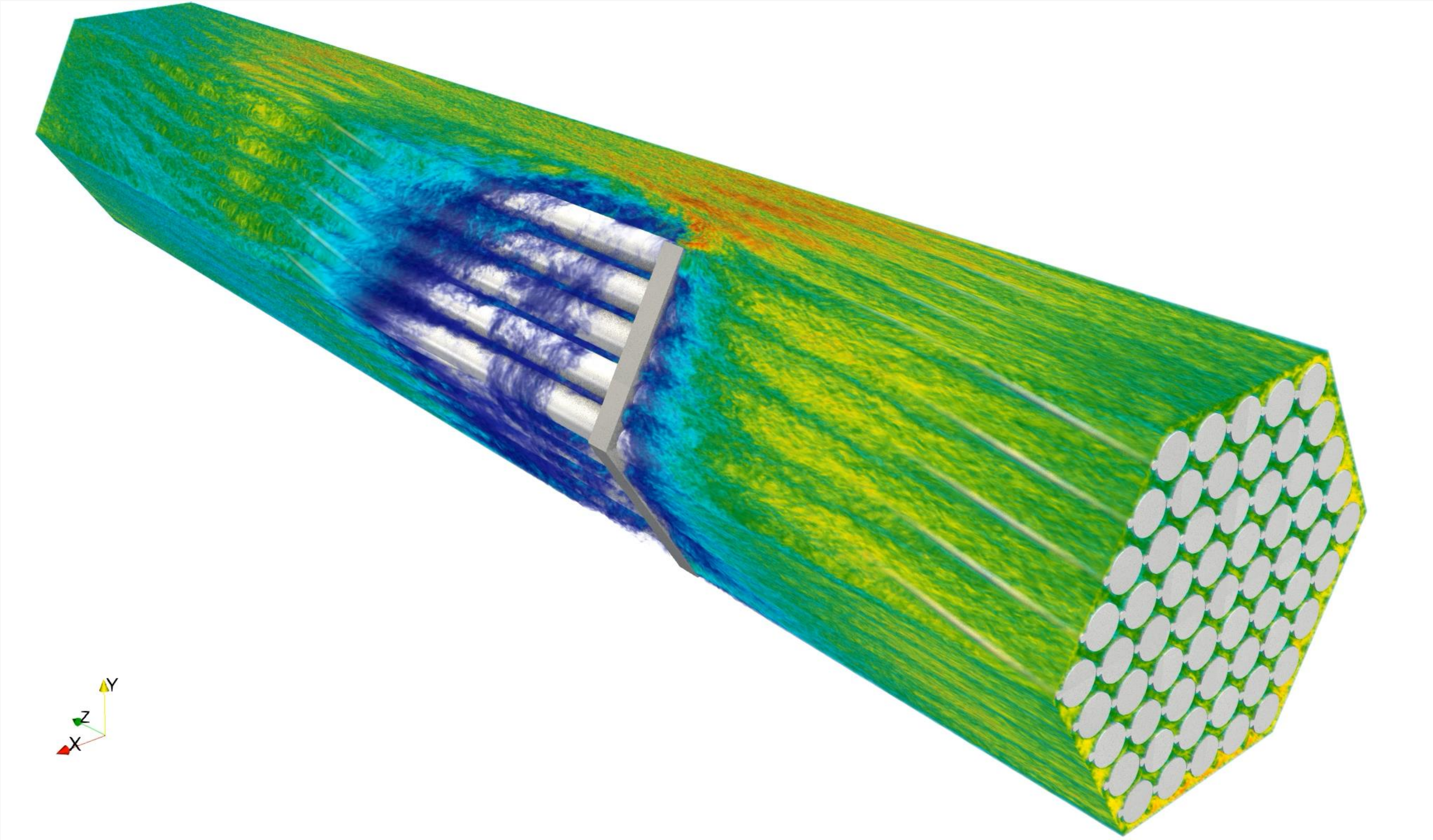
 Contents of this slide courtesy of Utkarsh Ayachit (Nvidia Corp., formerly Kitware, Inc.)

In situ vis from our science users

Animation: CFD Nuclear Reactor Simulation

Credit: Yu-Hsiang Lan and NekRS team at ANL

Rendered in situ with Ascent



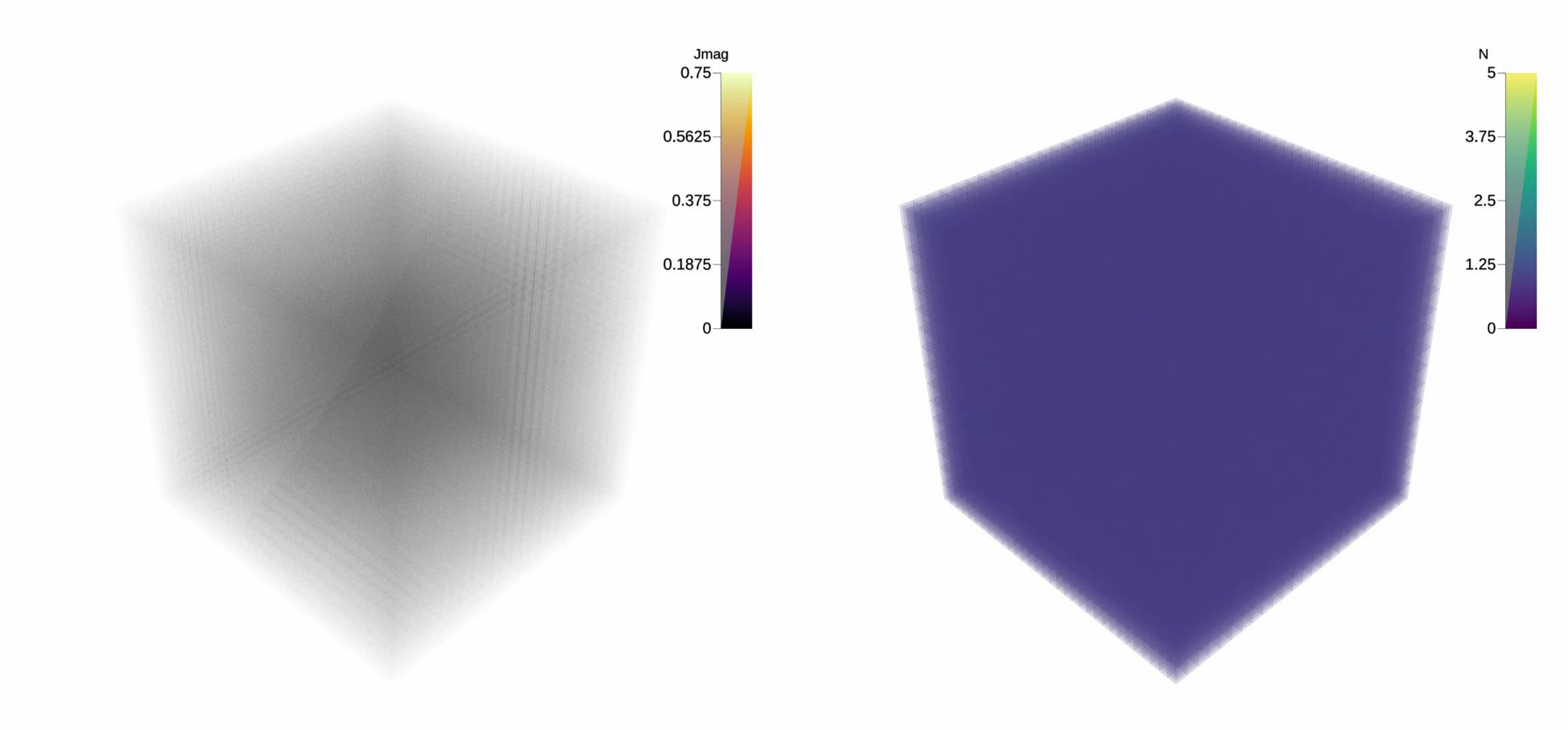
Youtube playlist: https://www.youtube.com/playlist?list=PLWn3pf_cs5UhCZ7xz23DDwPvt-DeLAGy8

In situ vis from our science users

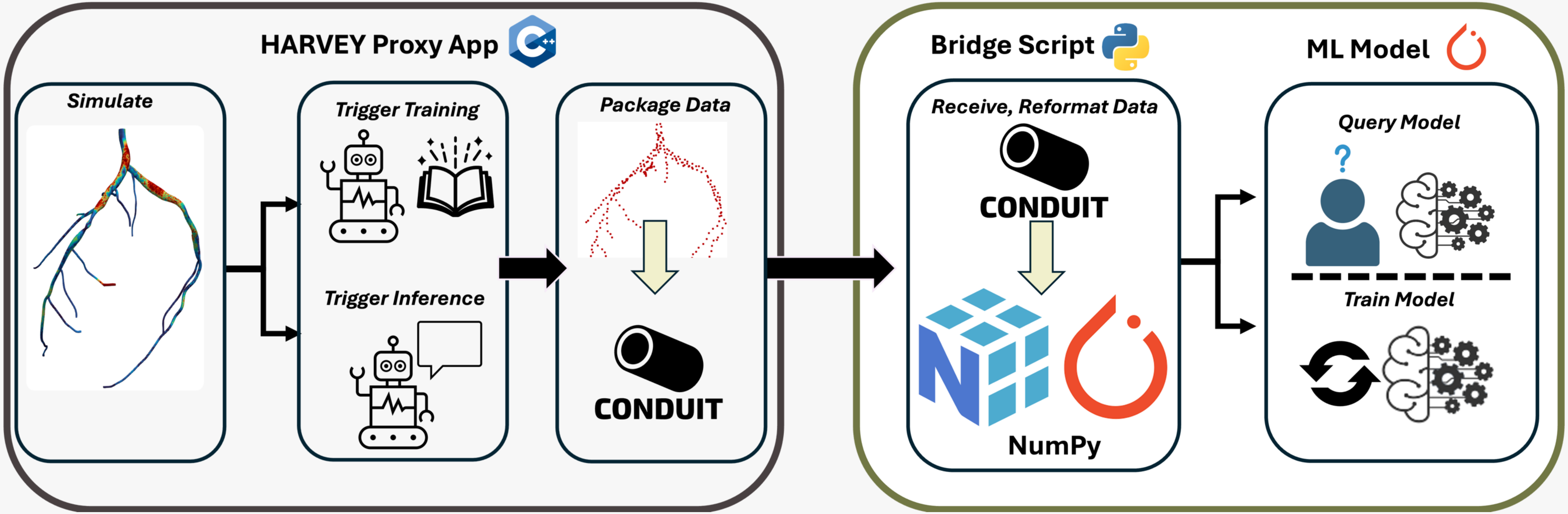
In situ instrumentation with Ascent using Claude during an ALCF hackathon

Credit: Ludwig Böss, Dept. of Astronomy and Astrophysics, Univ. of Chicago

Code: https://github.com/entity-toolkit/entity



Coupling Solvers and Machine Learning for In Situ Training and Inference



Slide courtesy Ayman Yousef, Duke University

Yousef, Ayman, Corey Wetterer-Nelson, Mengjiao Han, Victor Mateevitsi, Joseph Insley, Silvio Rizzi, Janet Knowles, Michael E. Papka, and Amanda Randles. "Instrumenting Lightweight, Modular Machine Learning Training and Inference in Parallel Solvers." In *International Conference on Computational Science*, pp. 123-137. Cham: Springer Nature Switzerland, 2026.

QUESTIONS?

Joe Insley
insley@anl.gov

Silvio Rizzi
srizzi@anl.gov