BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY

VACET: Deploying Technology for Visualizing and Analyzing Astrophysics Simulations

Gunther H. Weber, Hank Childs

May 19, 2010

Outline

- VACET Overview
- VisIt Overview
- Success stories with the Computational Astrophysics Consortium
- Parallel particle advection (streamlines)
- Topology-based Data Analysis

Outline

- VACET Overview
- VisIt Overview
- Success stories with the Computational Astrophysics Consortium
- Parallel particle advection (streamlines)
- Topology-based Data Analysis

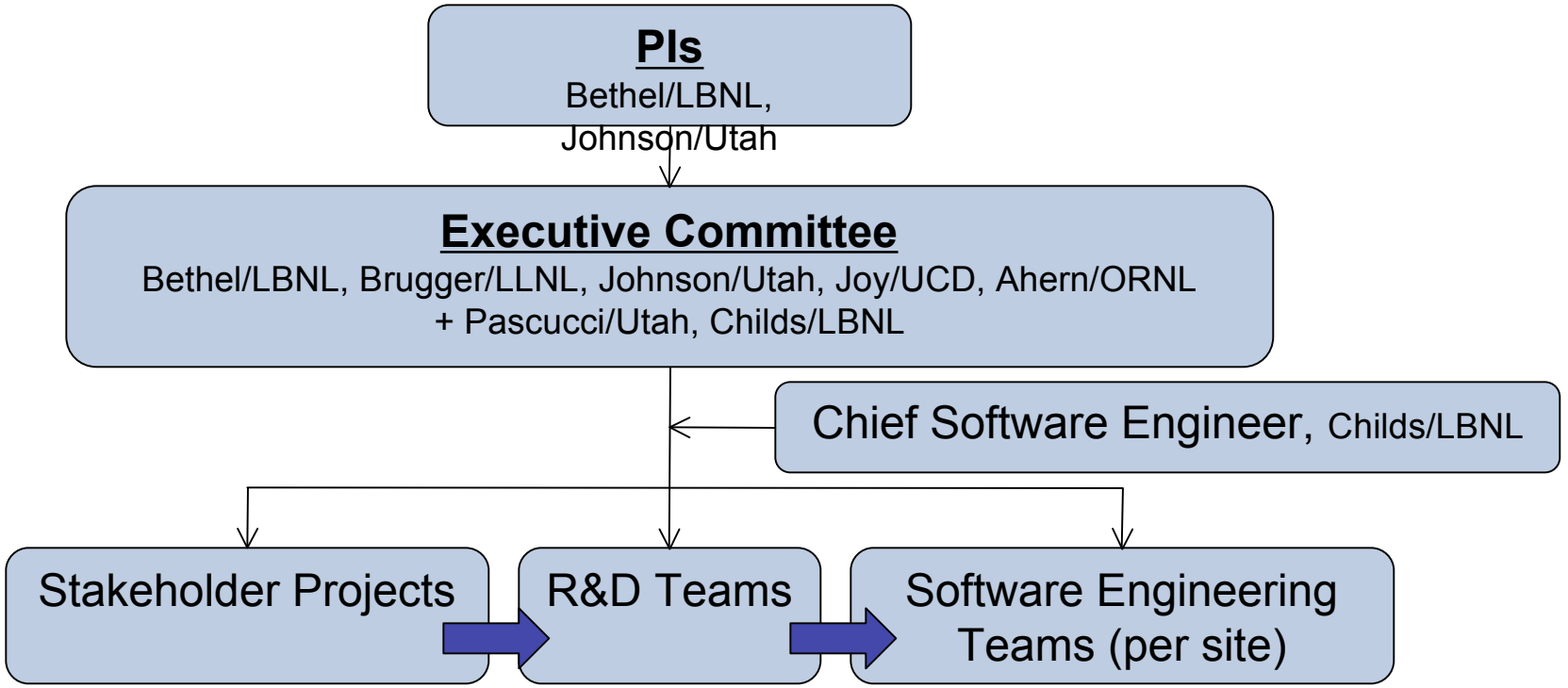
VACET Mission and Vision

- **Mission**: enable **scientific insight** for **petascale** data.
- **Strategy**: Leverage scientific visualization and analytics software technology as an enabling technology.
- **Vision**: adapt, extend, create, and deploy data understanding technologies for science stakeholders

- Why are we a center?
  - As a center, well positioned to respond to diverse needs/objectives through coordinated R&D, software engineering, outreach efforts.

VACET Organization

- Teams: stakeholder projects, R&D projects, software engineering projects.
- Executive committee: cross-institutional, cross-team coordination.

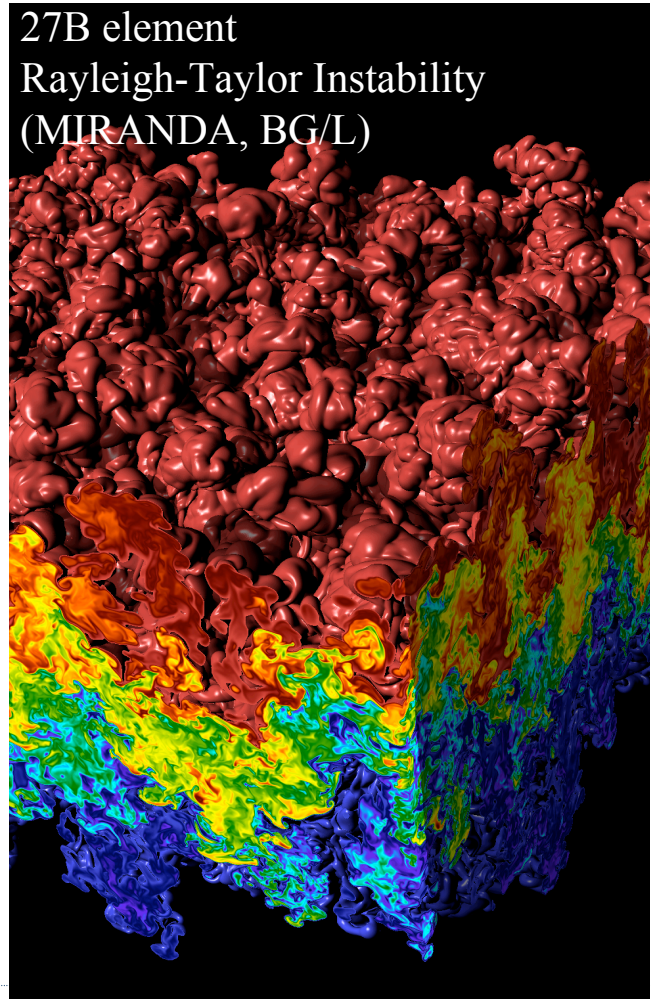


Outline

- VACET Overview
- VisIt Overview
- Success stories with the Computational Astrophysics Consortium
- Parallel particle advection (streamlines)
- Topology

VisIt is the primary deployment tool of VACET technologies to the Office of Science.

- Robust tool for end users + top notch support for big data
- VisIt is an open source, end user visualization and analysis tool for simulated and experimental data
  - Used by: physicists, engineers, code developers, vis experts
  - >100K downloads on web
- R&D 100 award in 2005
- Used “heavily to exclusively” on 8 of world’s top 12 supercomputers

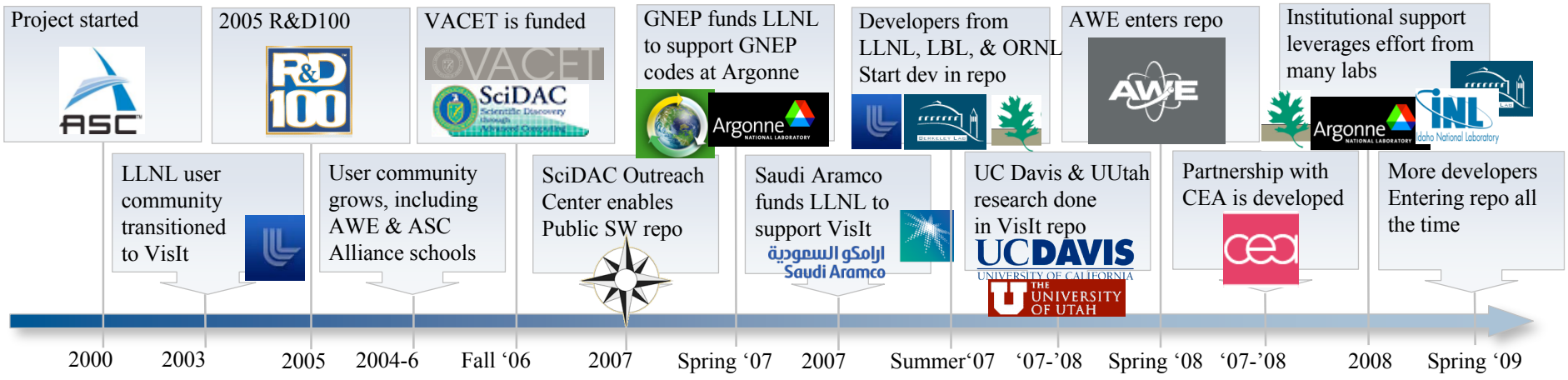


VisIt has a rich feature set that can impact many science areas.

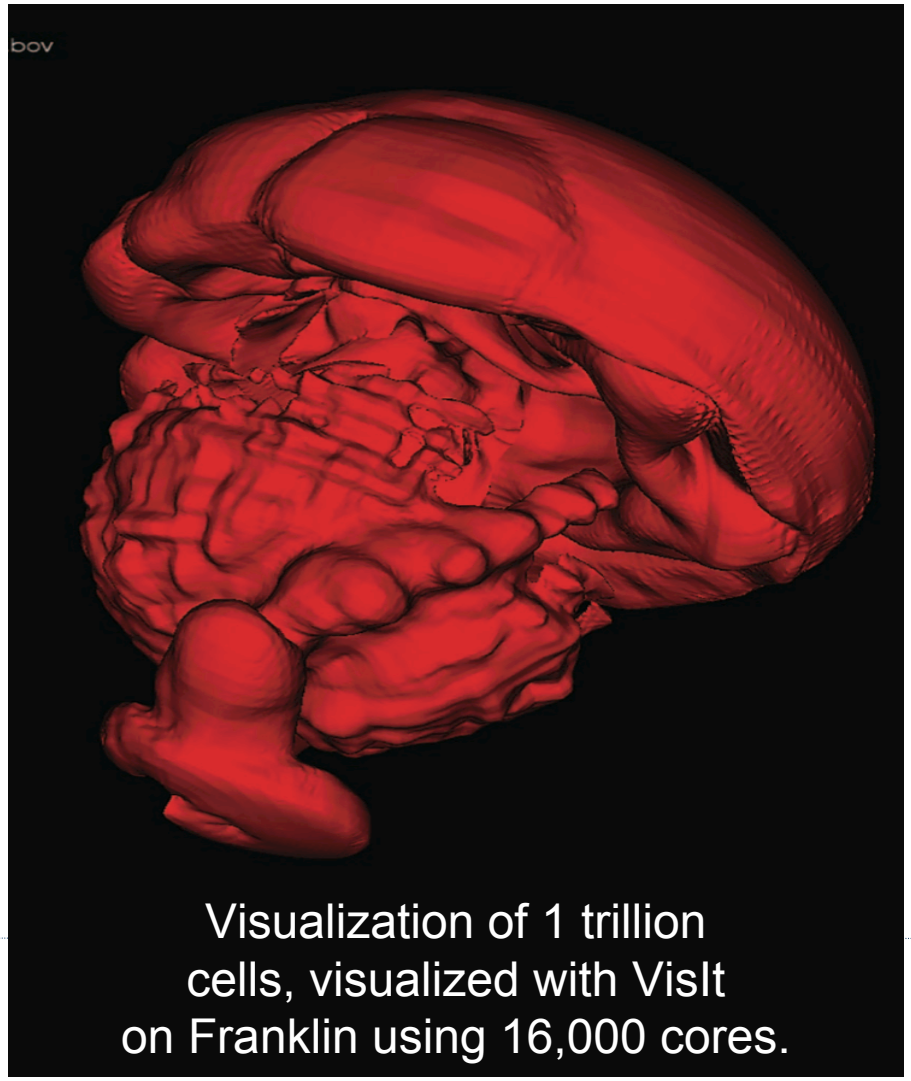
- *Meshes*: rectilinear, curvilinear, unstructured, point, AMR
- *Data*: scalar, vector, tensor, material, species
- *Dimension*: 1D, 2D, 3D, time varying
- *Rendering (~15)*: pseudocolor, volume rendering, hedgehogs, glyphs, mesh lines, etc…
- *Data manipulation (~40)*: slicing, contouring, clipping, thresholding, restrict to box, reflect, project, revolve, …
- *File formats (~85)*: incl. BoxLib (MAESTRO & CASTRO)
- *Derived quantities*: >100 interoperable building blocks
- *Many general features*: position lights, make movie, etc
- *Queries (~50)*: ways to pull out quantitative information, debugging, comparative analysis

VisIt is a vibrant project with many developers and users.

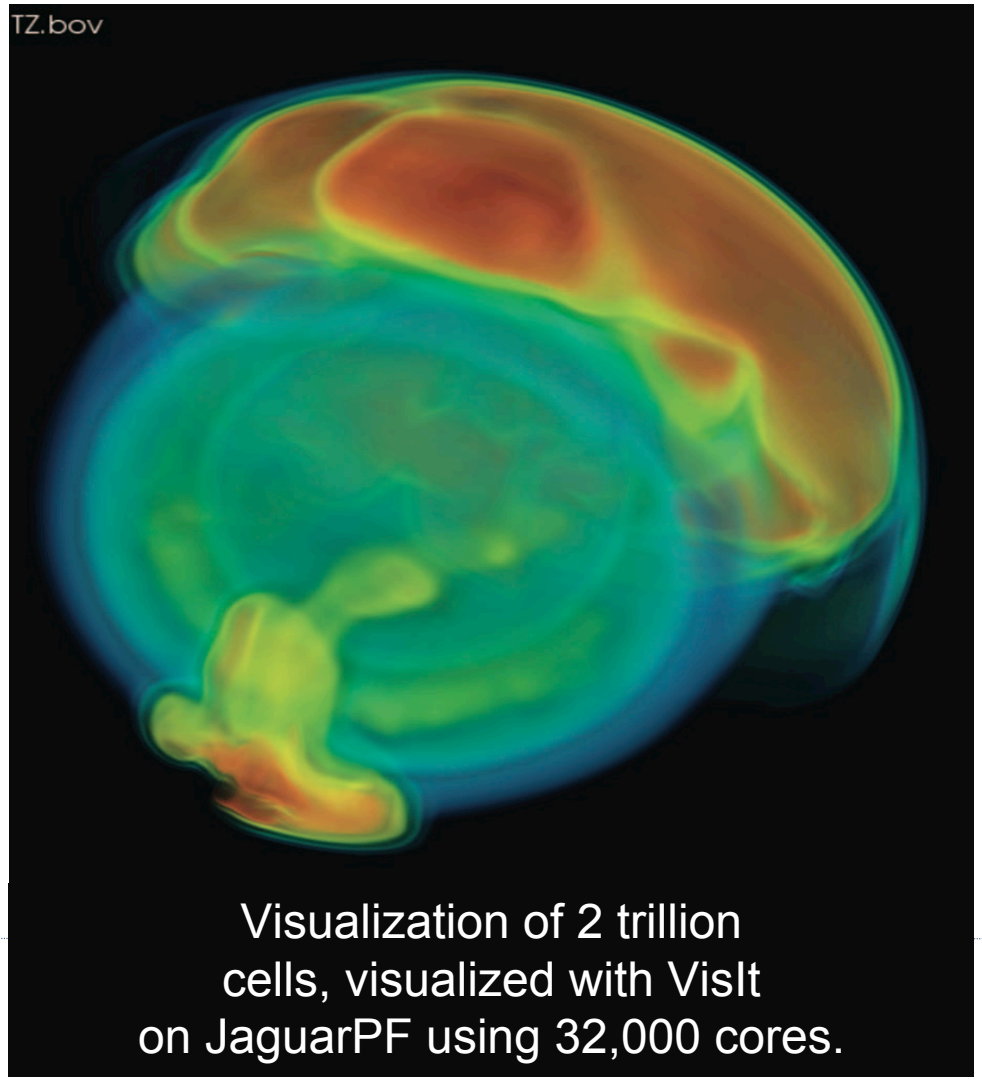
- VisIt represented over 50 person-years of effort at the time of VACET being funded
  - VisIt contains over one million lines of code
- More developers entering software repository all the time
  - Technology adopted by NSF XD centers



VACET has performed scaling studies to identify and fix bottlenecks when running VisIt on massive data sets.



Visualization of 1 trillion cells, visualized with VisIt on Franklin using 16,000 cores.



Visualization of 2 trillion cells, visualized with VisIt on JaguarPF using 32,000 cores.

We demonstrated that VisIt performs well on tens of thousands of cores with trillions of cells.

- Goal was to uncover bottlenecks on tomorrow's data.
- Experiments varied over supercomputing environment, data generation patterns, and I/O pattern.

Machine	No. of cores	Data set size (TCells)	Total I/O time (sec.)	Contour time (sec.)	Total pipeline execution time (sec.)†	Rendering time (sec.)
Purple	8,000	0.5	53.4	10.0	63.7	2.9
Dawn	16,384*	1.0	240.9	32.4	277.6	10.6
Juno	16,000	1.0	102.9	7.2	110.4	10.4
Ranger	16,000	1.0	251.2	8.3	259.7	4.4
Franklin	16,000	1.0	129.3	7.9	137.3	1.6
JaguarPF	16,000	1.0	236.1	10.4	246.7	1.5
Franklin	32,000	2.0	292.4	8.0	300.6	9.7
JaguarPF	32,000	2.0	707.2	7.7	715.2	1.5

* *Dawn requires that the number of cores be a power of two.*
† *This measure indicates the time to produce the surface.*

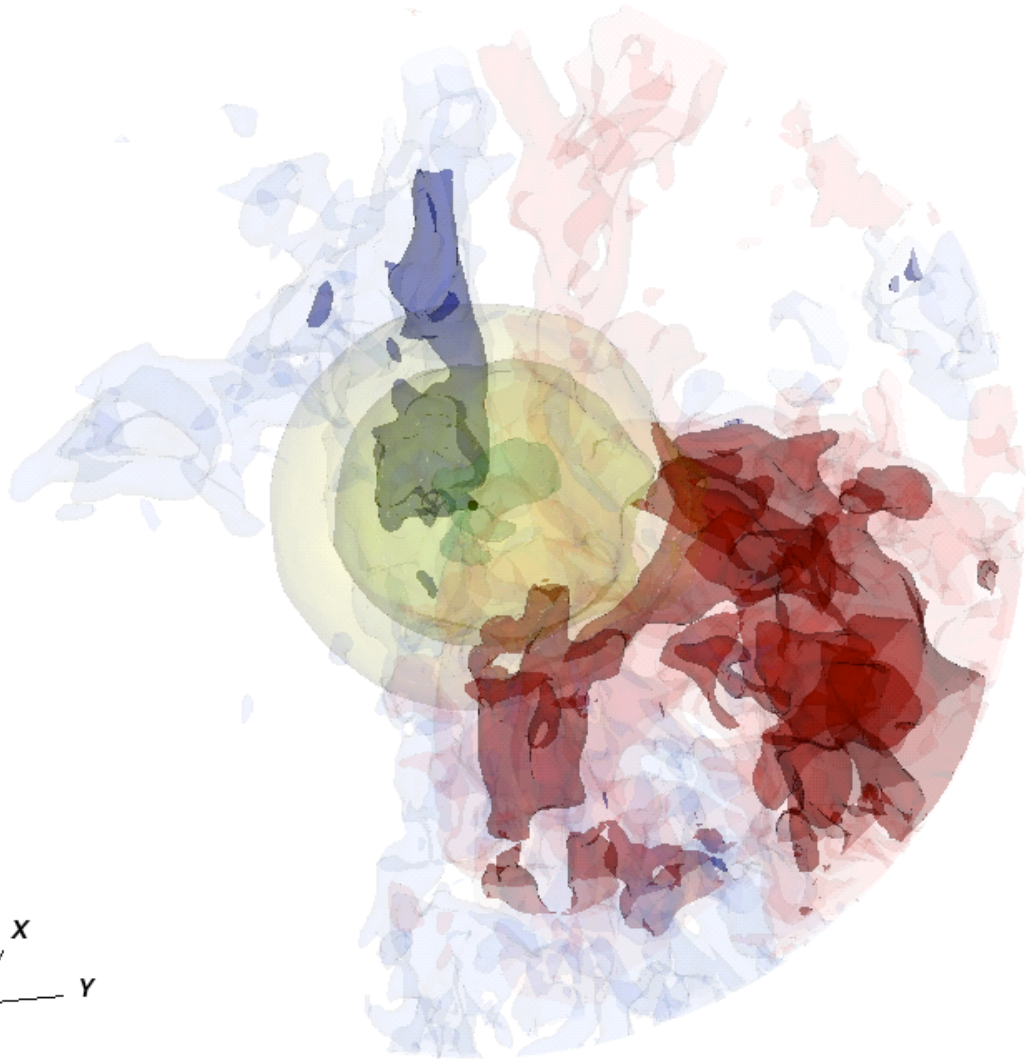
Outline

- VACET Overview
- VisIt Overview
- Success stories with the Computational Astrophysics Consortium
- Parallel particle advection (streamlines)
- Topology

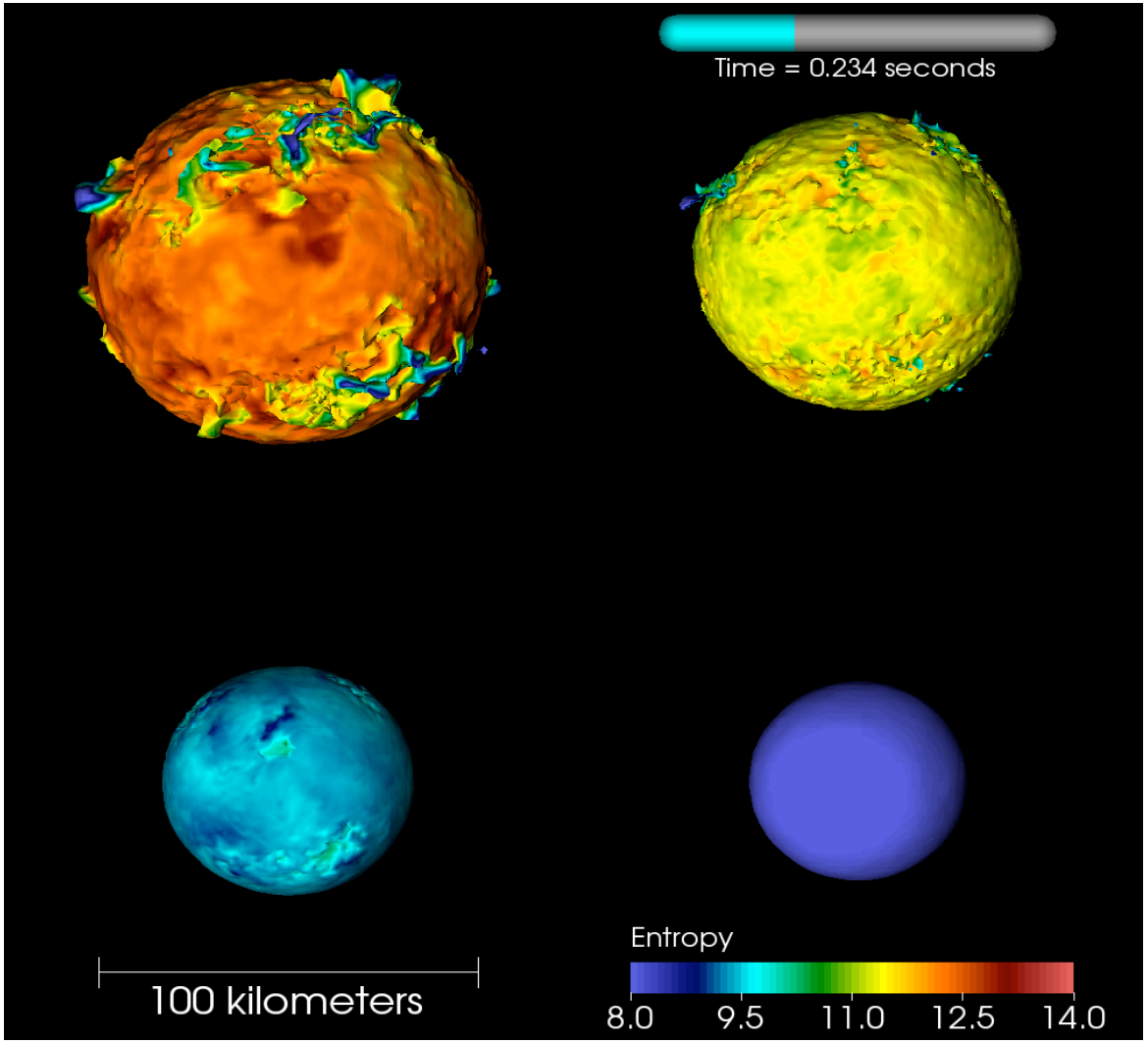
Success Stories with the Computational Astrophysics Consortium

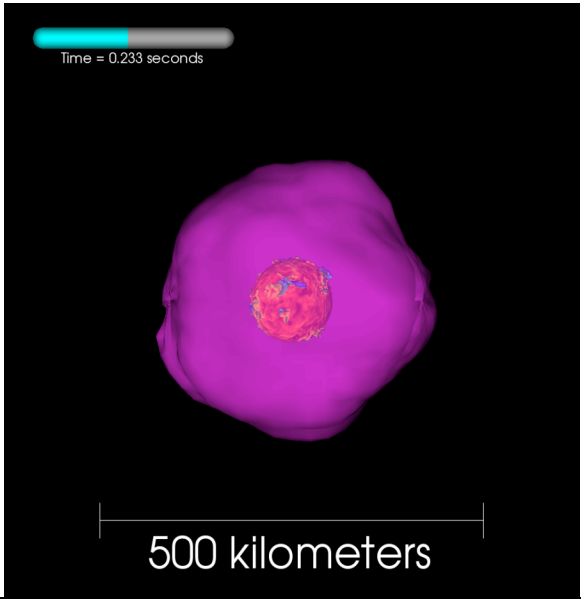
- VisIt already supported AMR data and BoxLib format
- Worked with CCSE on ensuring that VisIt can read CASTRO and MAESTRO files
- VisIt enhancements for AMR data
- Provided VisIt advice, expert support, scripts etc. to CCSE and collaborators
- Efforts leverage VACET / NERSC Analytics connection

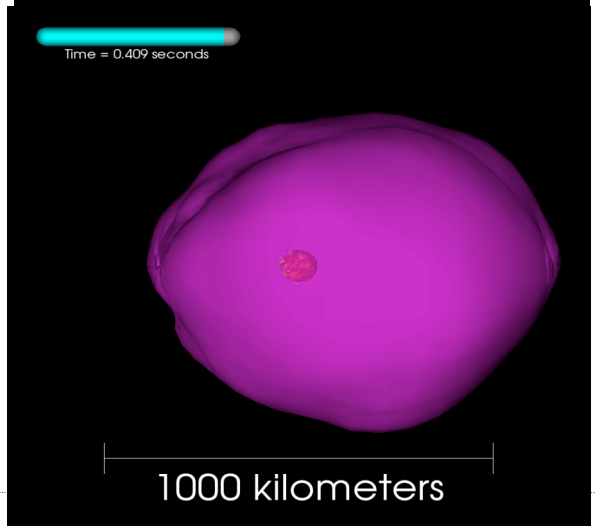
Visualizing Radial Velocity of a MAESTRO Visualization



We Have Helped Princeton's CASTRO Users Understanding Their 3D Simulations







Outline

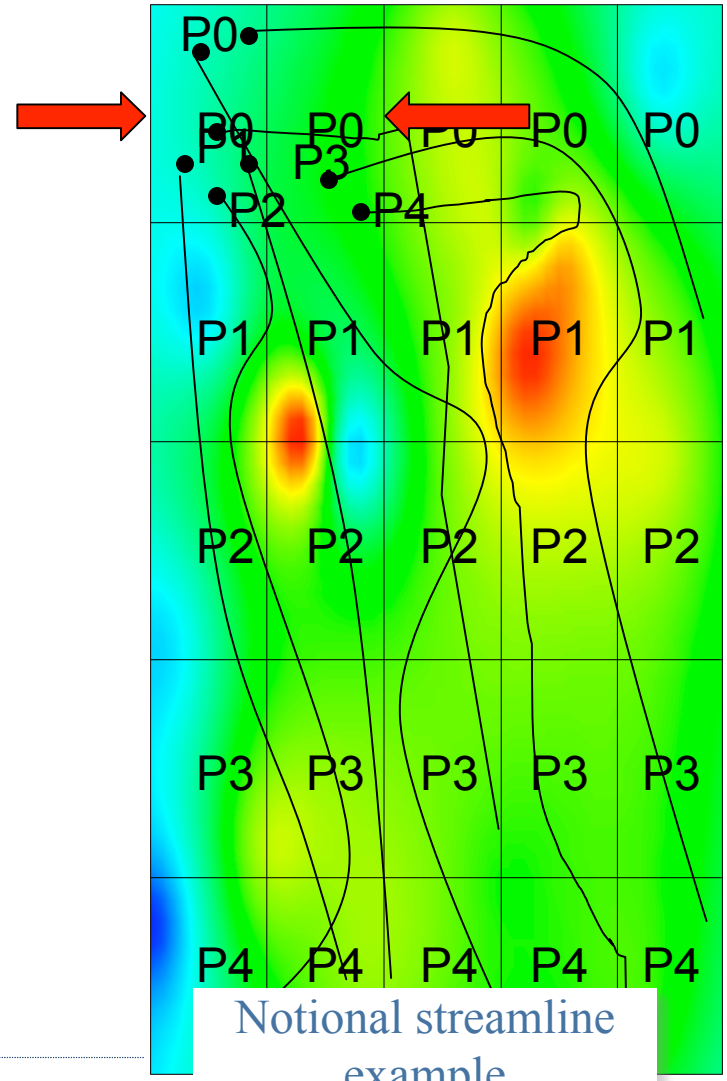
- VACET Overview
- VisIt Overview
- Success stories with Computational Astrophysics Consortium
- Parallel particle advection (streamlines)
- Topology

Particle advection informs flow behavior

- Basic idea: insert massless particle and advect by velocity field (streamlines)
- Fundamental complexity:
  - Efficiently load balancing this activity in a distributed memory, parallel environment is very difficult (see next slide)
- Further complexities:
  - Varying the velocity field in time (pathlines)
  - Integrating over AMR meshes
  - Many potential use cases, from visualization to analysis
- Takeaway:
  - Implemented an efficient streamline algorithm deployed in VisIt
  - We believe algorithm is highly relevant to the astrophysics community for visualization and analysis

Existing parallel integral curve techniques have suffered from load imbalance.

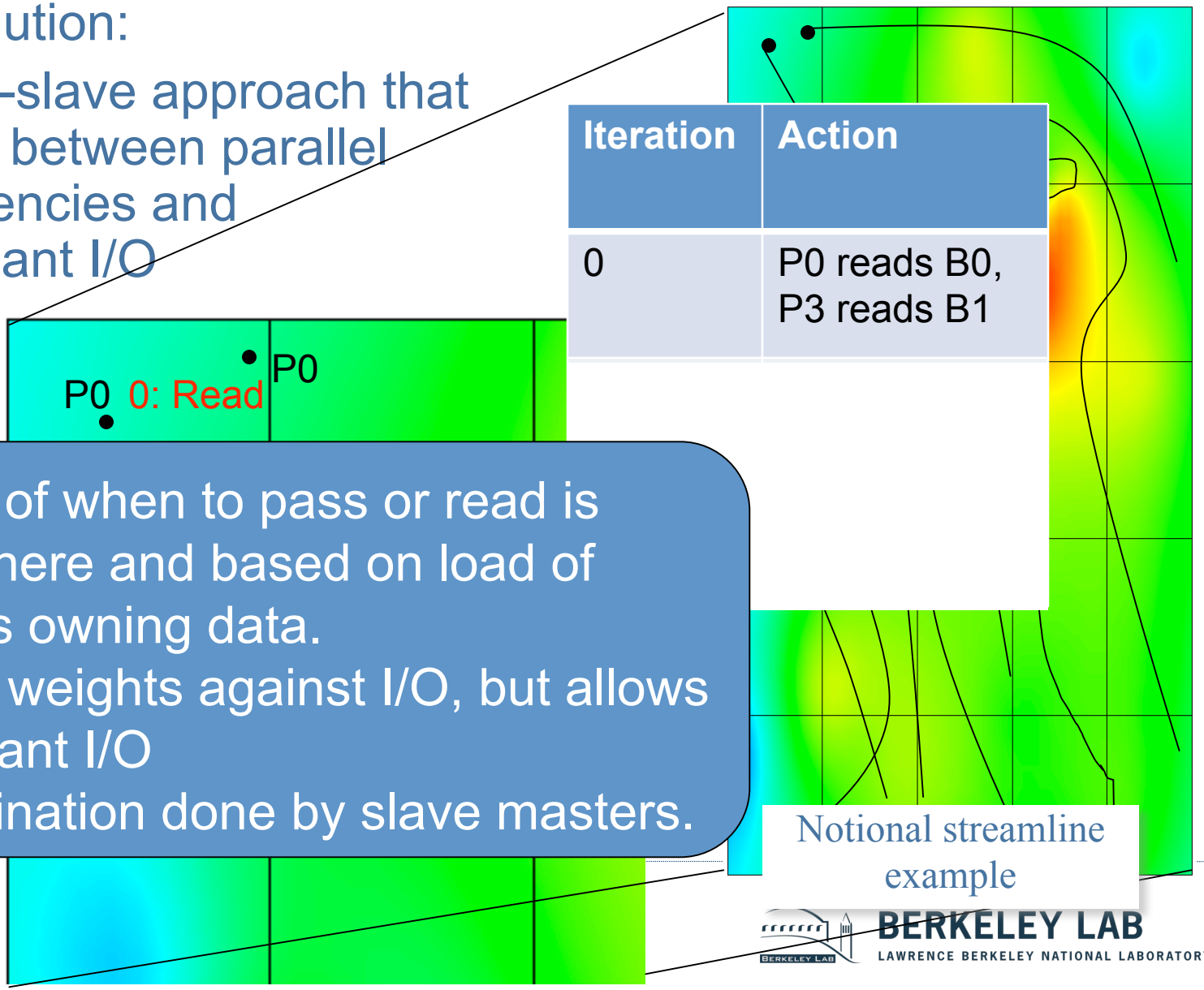
- Two extremes:
  - Partition data over processors and pass particles amongst processors
    - Parallel inefficiency!
  - Partition seed points over processors and process necessary data for advection
    - Redundant I/O!



Notional streamline example

We have greatly advanced the state of the art for parallel integral curve techniques.

- Hybrid solution:
  - Master-slave approach that adapts between parallel inefficiencies and redundant I/O

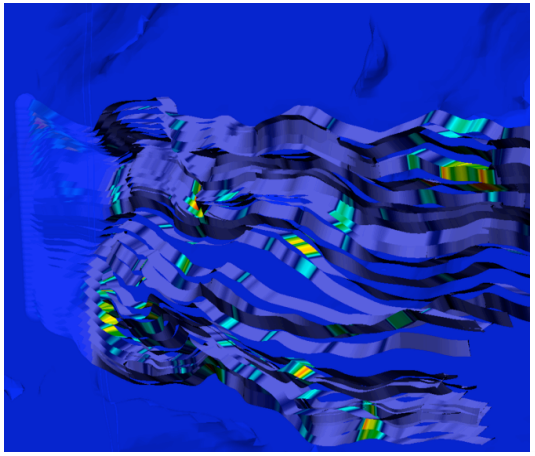


Iteration	Action
0	P0 reads B0, P3 reads B1

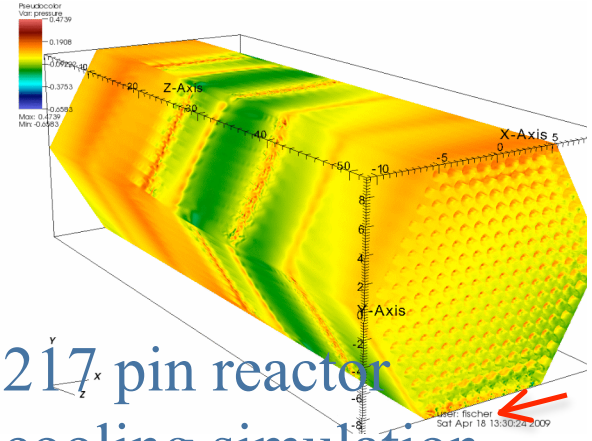
- Decision of when to pass or read is simplified here and based on load of processors owning data.
- Heuristic weights against I/O, but allows for redundant I/O
- All coordination done by slave masters.

Notional streamline example

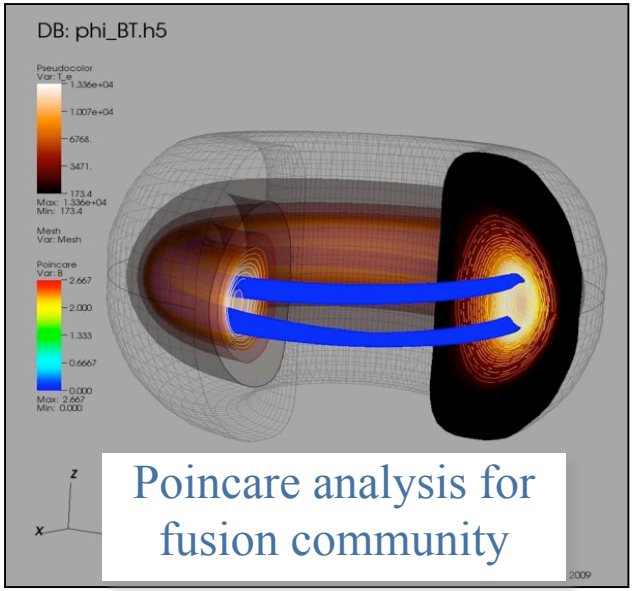
This work has had a broad and varied impact.



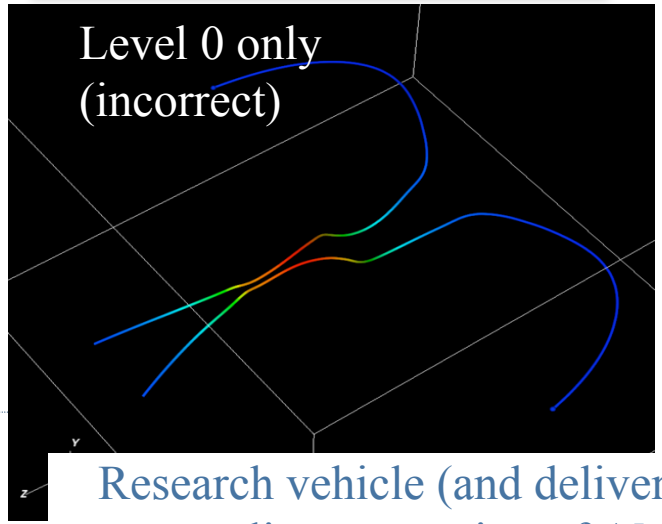
Used in analysis by SHOCKS center (image courtesy SHOCKS website)

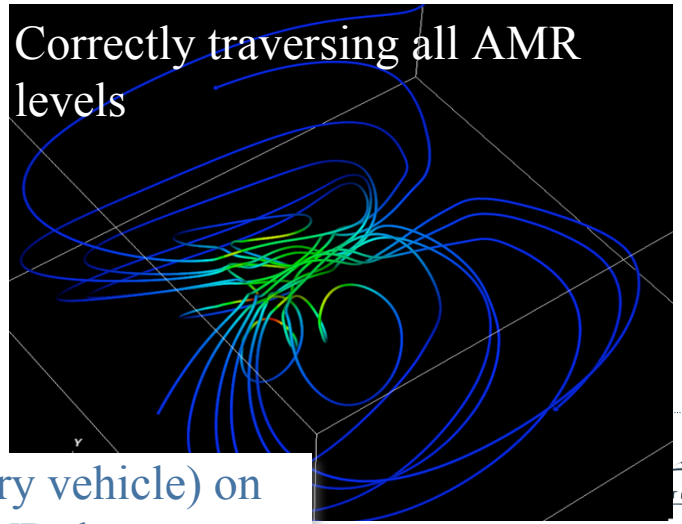


217 pin reactor cooling simulation. Run on ¼ of Argonne BG/P.



Poincare analysis for fusion community





Research vehicle (and delivery vehicle) on streamline generation of AMR data sets.

Streamlines used in visualization of type II supernova collapse

Outline

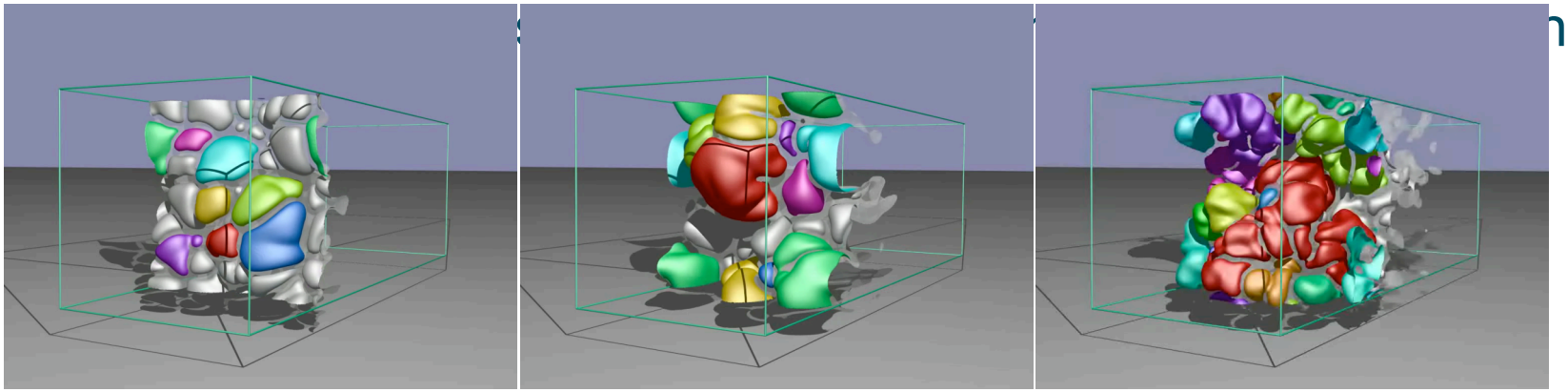
- VACET Overview
- VisIt Overview
- Success stories with Computational Astrophysics Consortium
- Parallel particle advection (streamlines)
- Topology-based Data Analysis

Topology-based Data Analysis

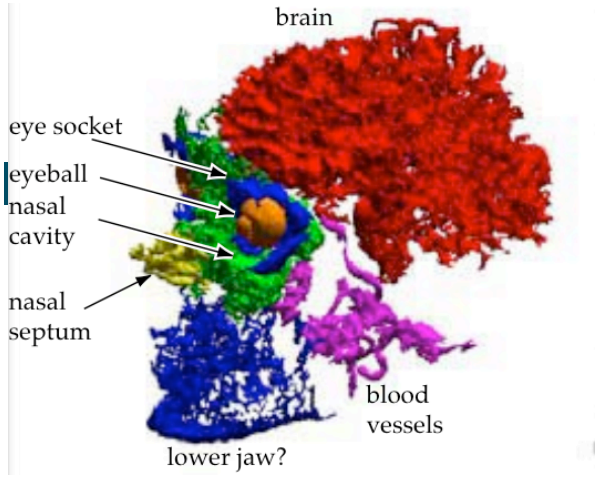
- Significant portion of data analysis feature driven
  - Examples: Isosurfaces or interval volumes
- Topology provides general framework for feature definition
  - Robust: stable under noise, adapted to piece-wise linear and/or constant data;
  - Reliable: strong guarantees wrt. feature dimension and global structure;
  - Hierarchical: inherits natural feature-based hierarchies;
  - Efficient: compute/encode entire feature families; and
  - Flexible: features are sub-sets of more general structures allowing a small set of tools to apply to different applications.

Threshold-Based Features

- Features defined through contours are called *threshold-based*:

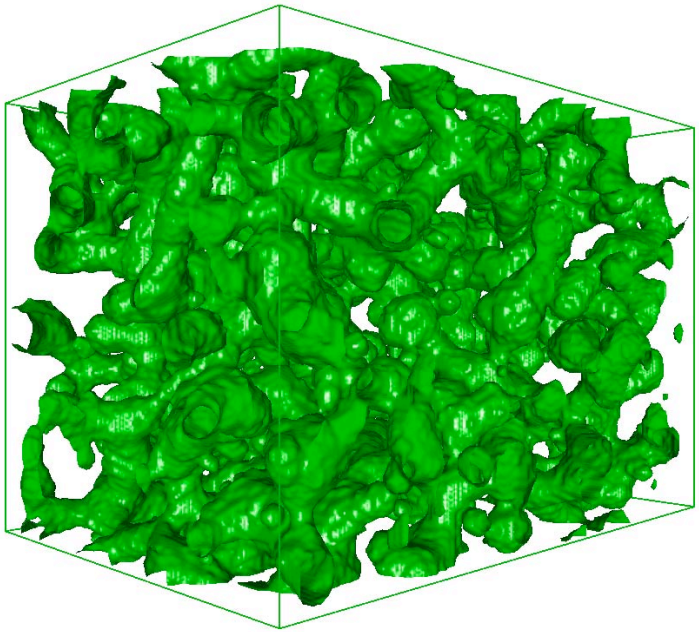


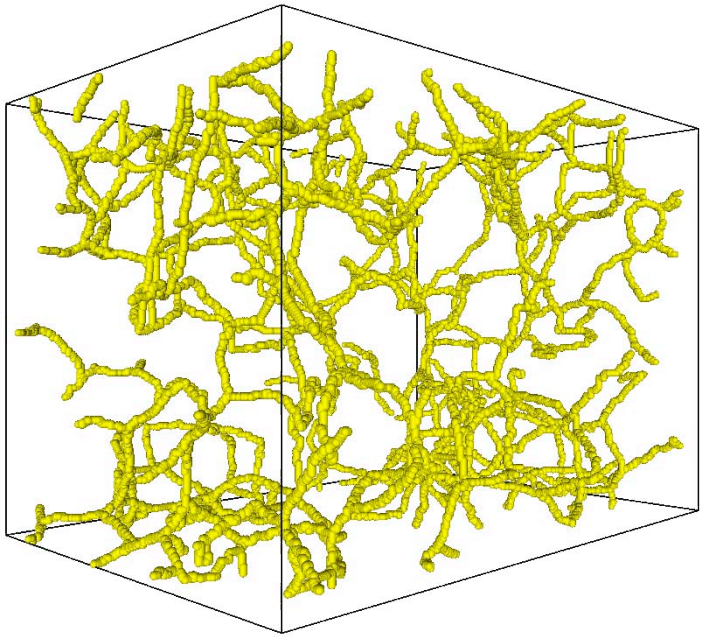
- Flexible iso-surfaces: Regions delimited by Regions local varying iso-values



Gradient-Based Features

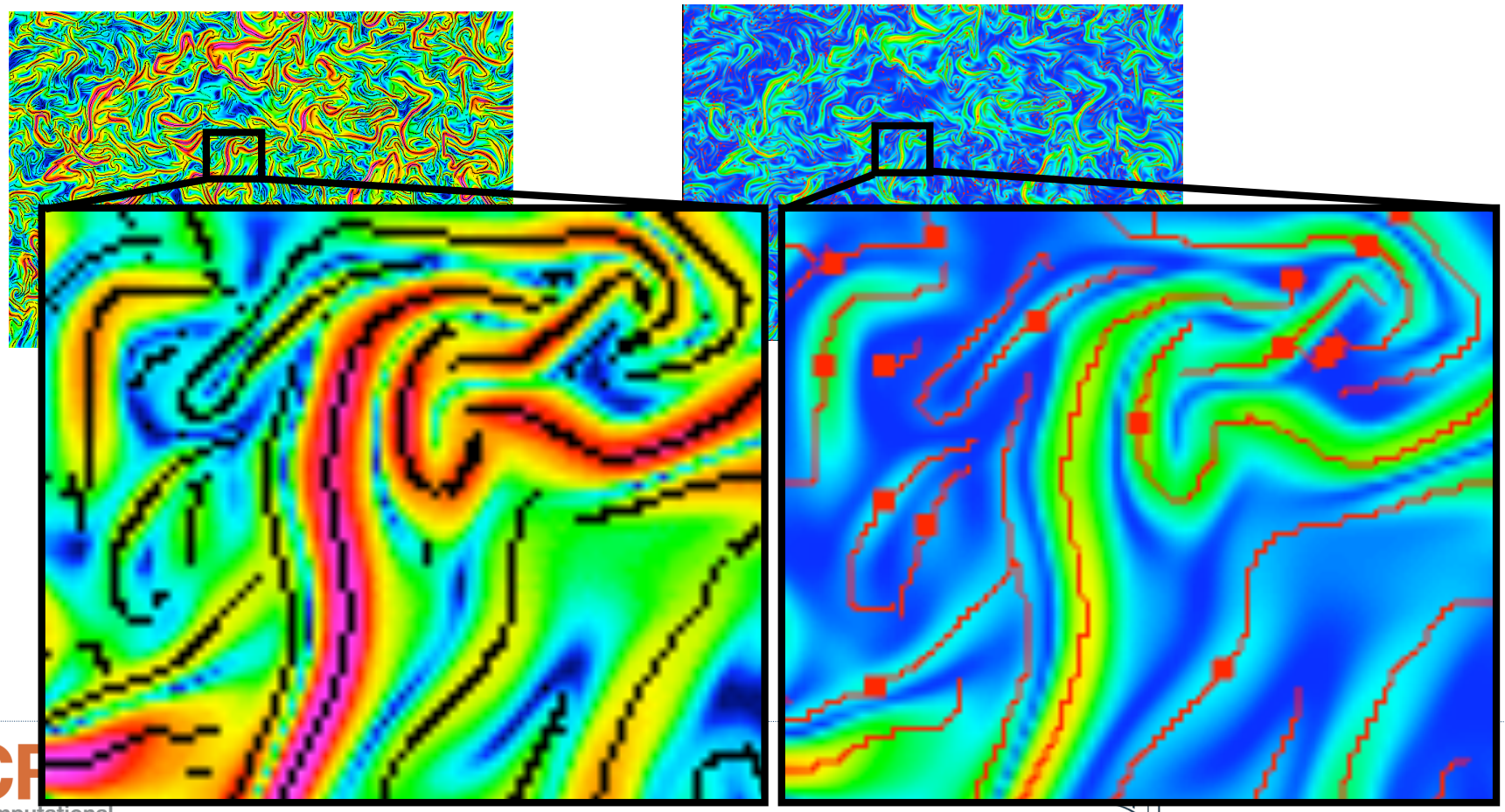
- Features defined as regions of similar (gradient) flow:





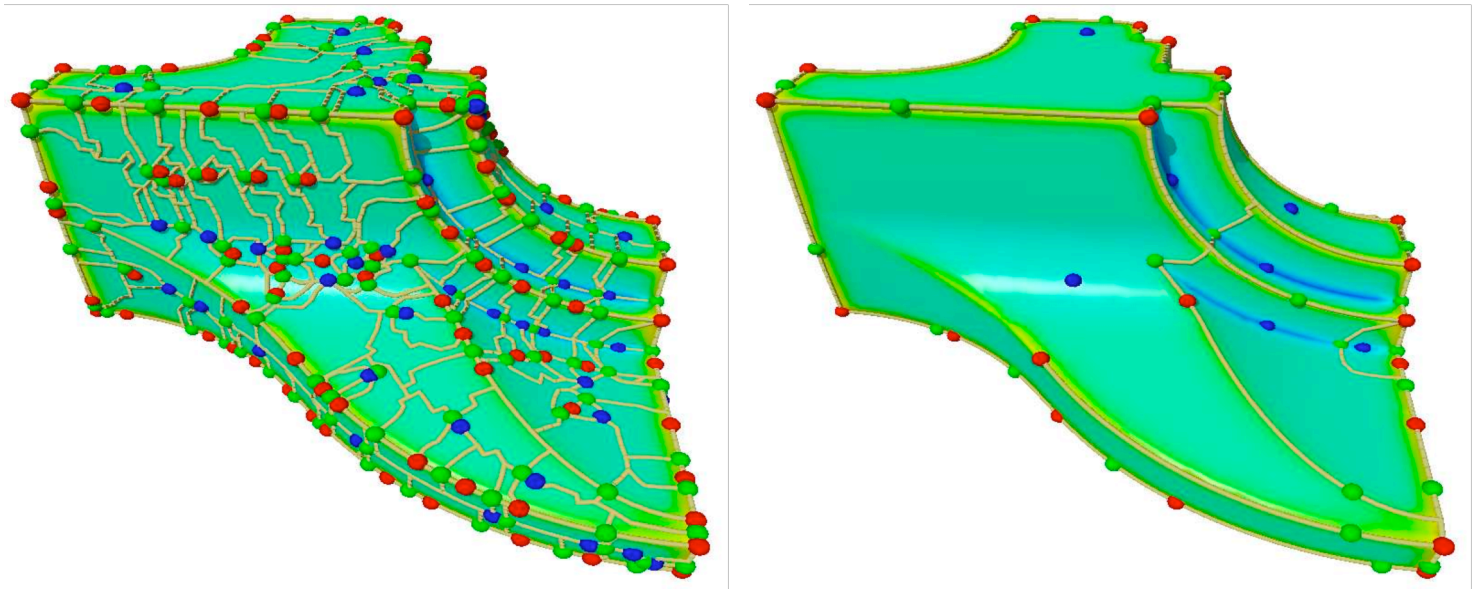
Why Use Topology to Define/Extract Well-Known Features ?

- Structural consistency:
  - Features are guaranteed to be of the expected dimension



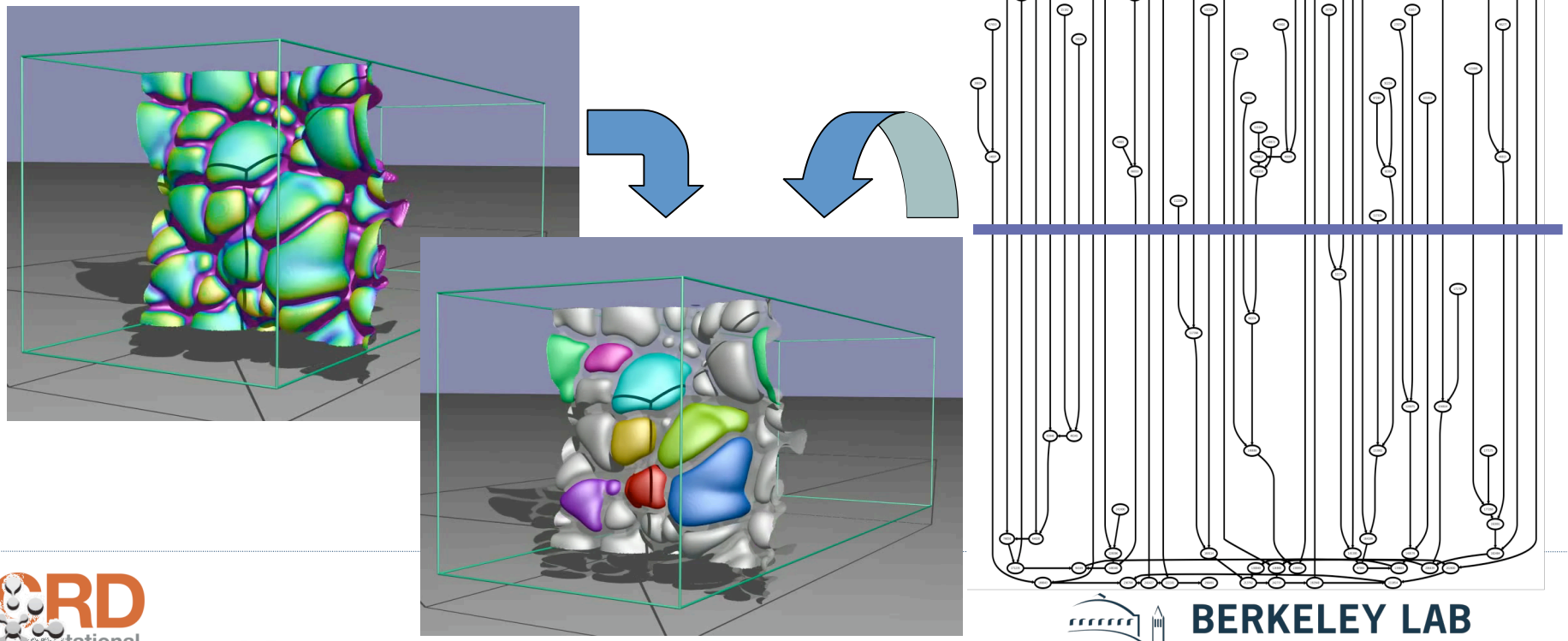
Why Use Topology to Define/Extract Well-Known Features ?

- Structural consistency
  - Features are guaranteed to be of the expected dimension
- Natural feature-based hierarchy:



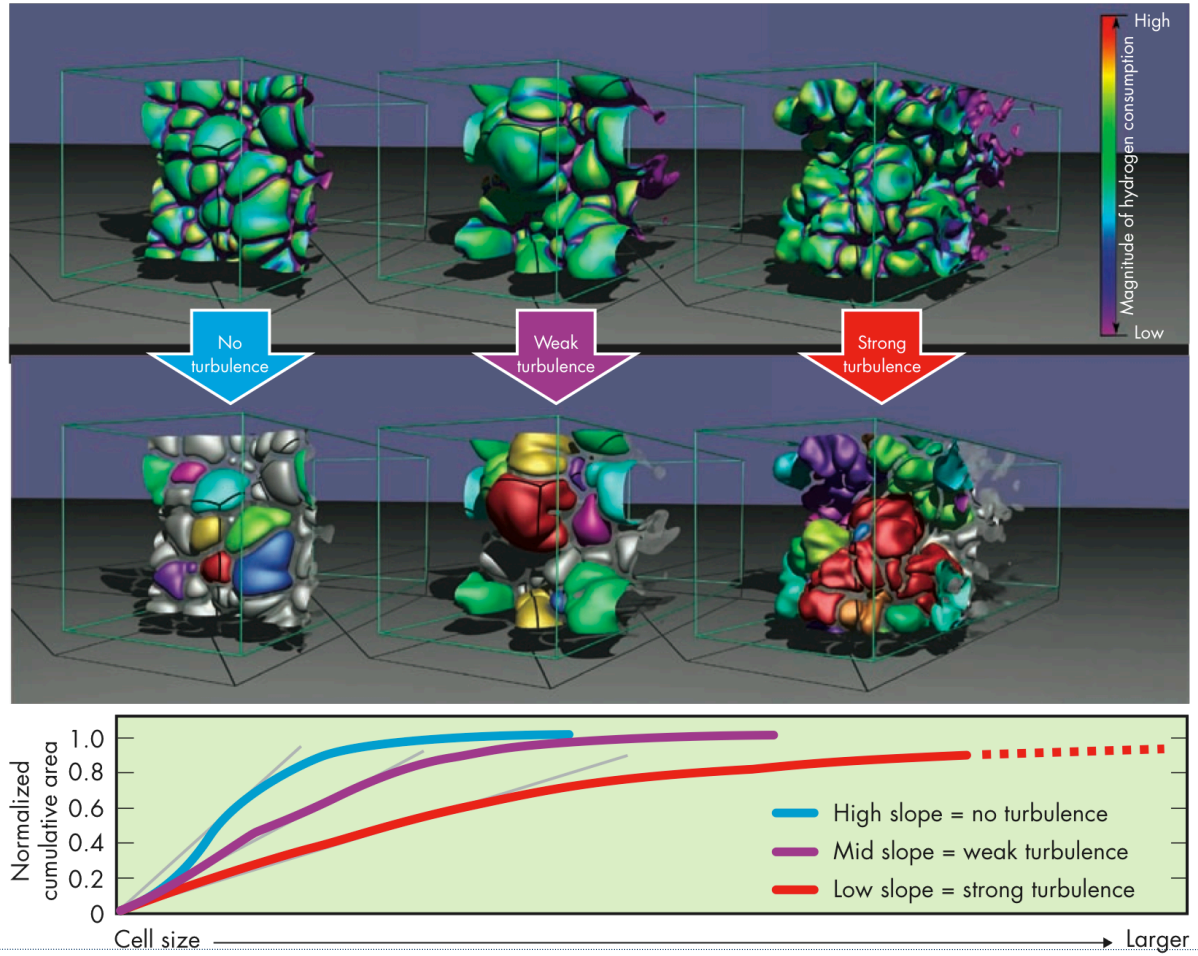
Why Use Topology to Define/Extract (Well-Known) Features ?

- Structural consistency:
  - Features are guaranteed to be of the expected dimension
- Natural feature-based hierarchy
- Parameter independent representations

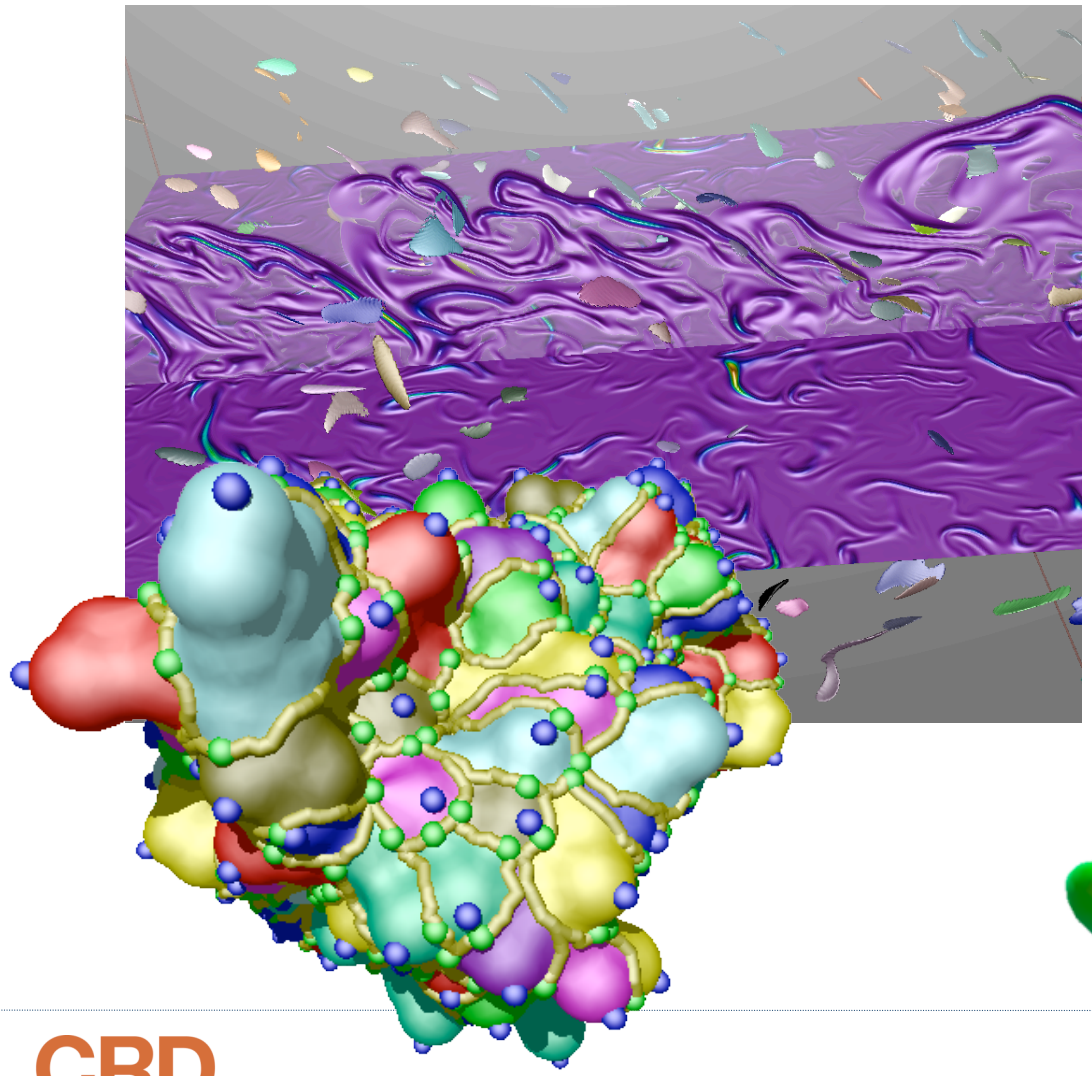


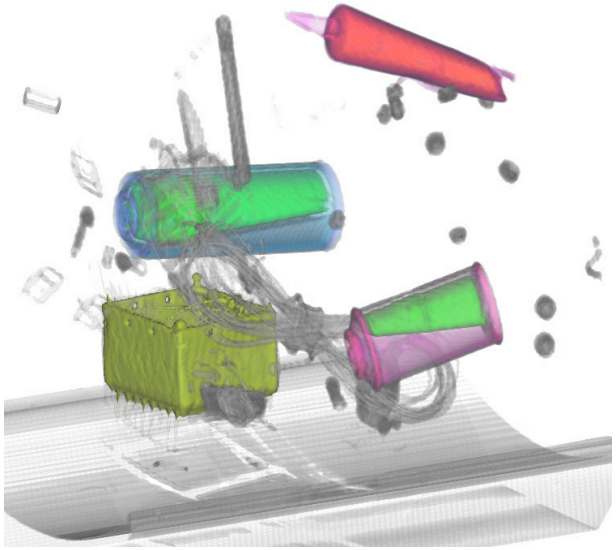
Topological Analysis Provides New Scientific Insights

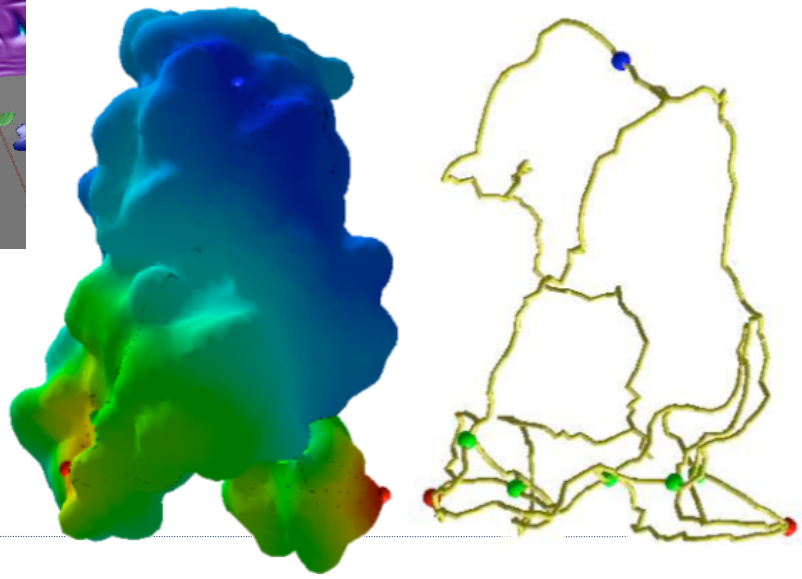
Dependency between level of turbulence



Other Examples of Topological Analysis







Summary

- VACET has been deploying VisIt to CAC participants for (1) bread-and-butter visualization needs and (2) advanced analysis needs (upcoming work not discussed here)
- VACET has been investing in technologies that we believe will be very relevant to the astrophysics community
  - Particle advection
  - Topology-based data analysis
  - Support for massive scale data

- Contact info:
  - Gunther Weber, ghweber@lbl.gov
  - VisIt help: visit-help-scidac@ornl.gov

Questions?

Additional Contributors

- VisIt Development Team
  - Too many to list
- VisIt Scaling Study
  - Sean Ahern, Wes Bethel, Mark Howison, Prabhat, Dave Pugmire, Brad Whitlock
- Parallel Streamlines
  - Dave Pugmire, Christoph Garth, Sean Ahern
- AMR Streamlines
  - Sergey Borovikov, Eduard Deines, Christoph Garth, Ken Joy, Dan Martin, Brian Van Straalen
- Topology-based Data Analysis
  - Peer-Timo Bremer, Valerio Pascucci