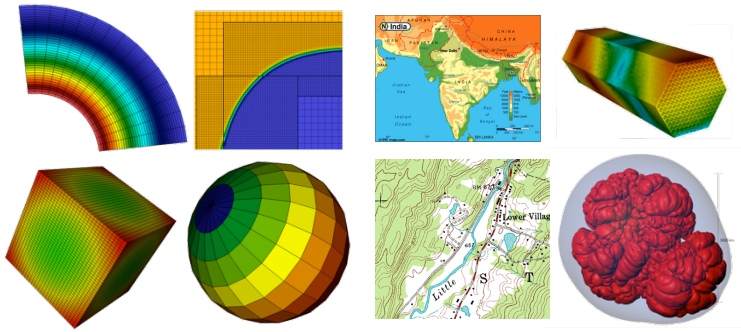
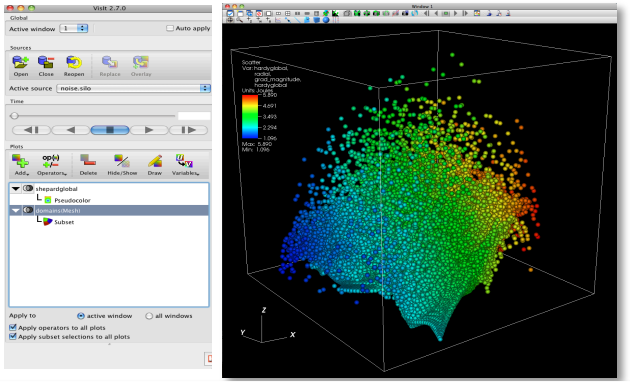
An introduction to Scientific Visualization using VisIt

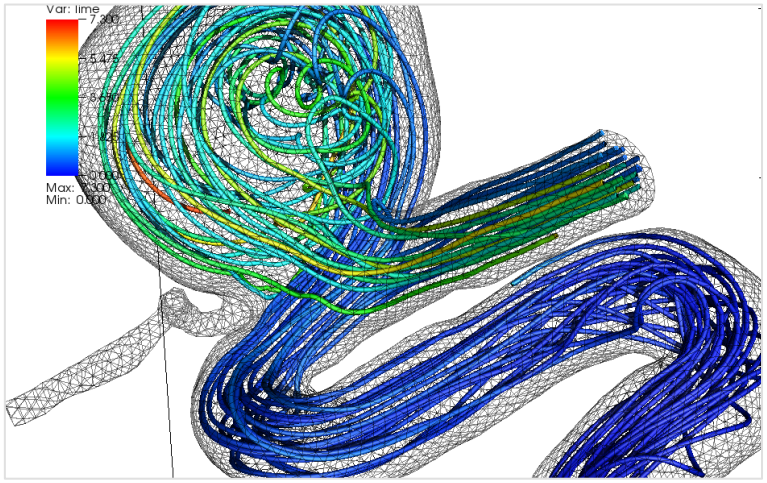


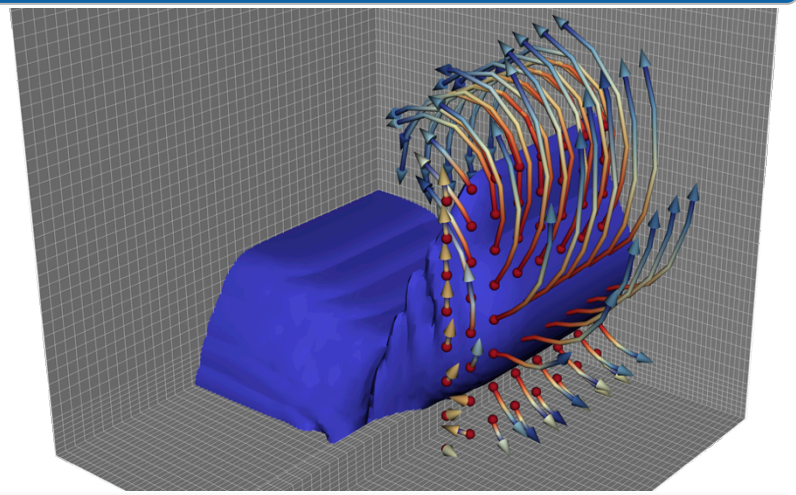


Scientific Visualization Concepts

Guided Tour of VisIt

With two in-depth / hands-on visualizations:





Aneurysm (Blood Flow) Simulation

Water Flow Simulation

Tutorial Outline

- Introductions:
  - Hari Krishnan (LBNL/NERSC, VisIt Developers)
- VisIt Overview
- Guided Tour & Showcase of VisIt
- Alternative ways of using VisIt.
- Demonstrations:
  - Visualization of an Aneurysm (Blood Flow) Simulation
  - Water Flow Simulation
- Closing Remarks and Questions

Tutorials:

http://visitusers.org/index.php?title=VisIt_Tutorial

Example Datasets, Blood Flow Tutorial, Water Flow Tutorial

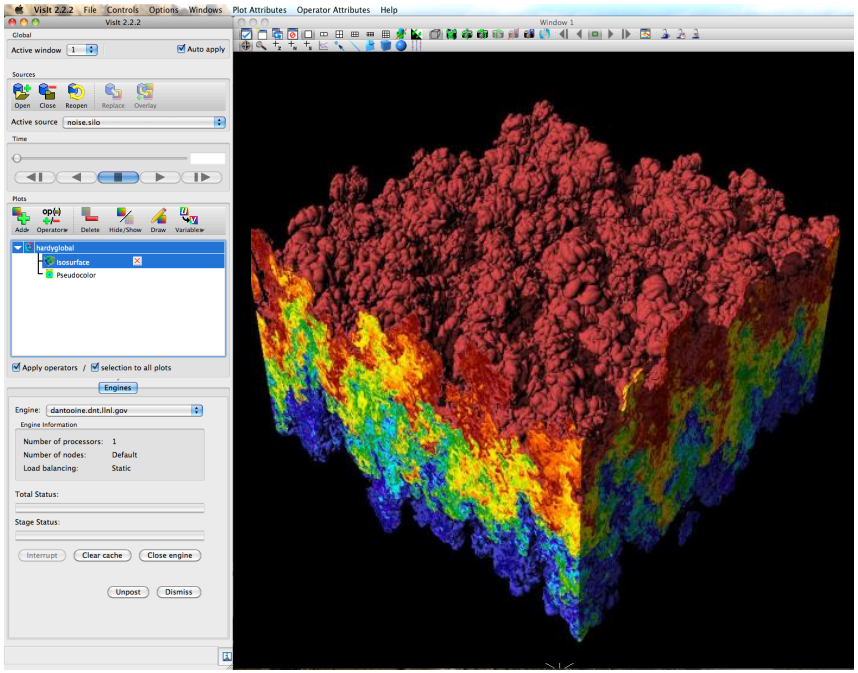
http://visitusers.org/index.php?title=Tutorial_Data

http://visitusers.org/index.php?title=Blood_Flow_Aneurysm_Tutorial

http://visitusers.org/index.php?title=Water_Flow_Tutorial

VisIt visualization tookit

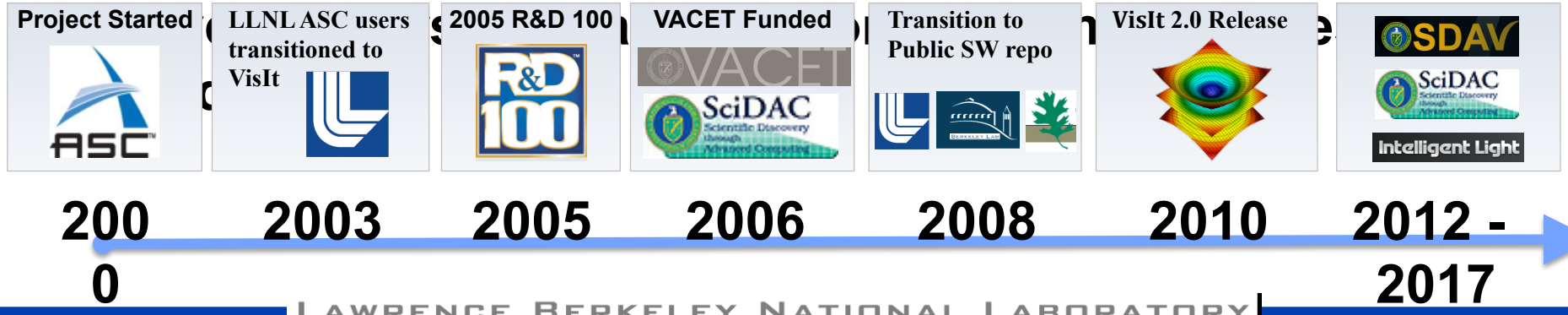
- open source, turnkey application for data analysis and visualization of mesh-based data
- Production end-user tool supporting scientific and engineering applications.
- Provides an infrastructure for parallel post-processing that scales from desktops to massive HPC clusters.
- Source released under a BSD style license.



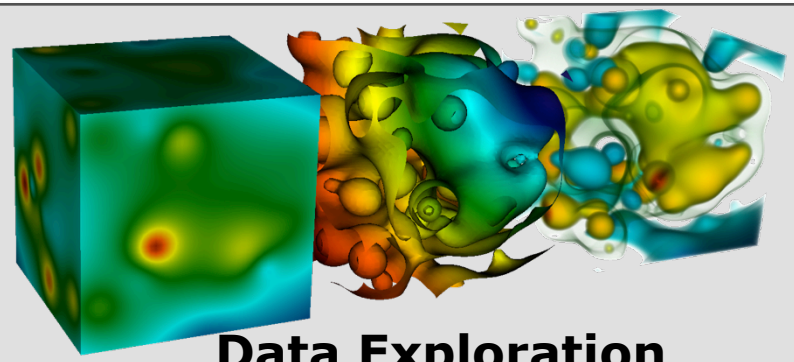
Density Isovolume of a 3K^3 (27 billion cell) dataset

VisIt is a vibrant project.

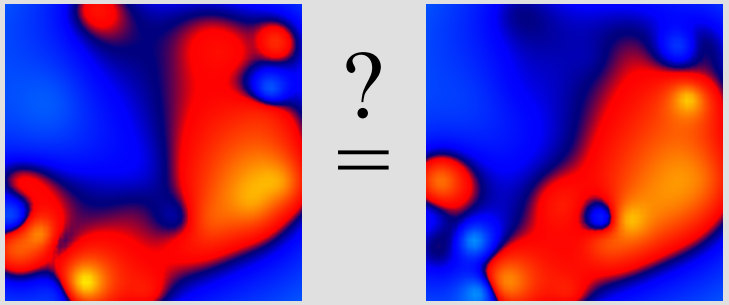
- The VisIt project started in 2000 to support LLNL's large scale ASC physics codes.
- The project grew beyond LLNL and ASC with research and development from DOE SciDAC and other efforts.
- VisIt is now supported by multiple organizations:
  - LLNL, LBNL, ORNL, UC Davis, Univ of Utah, Intelligent Light, …



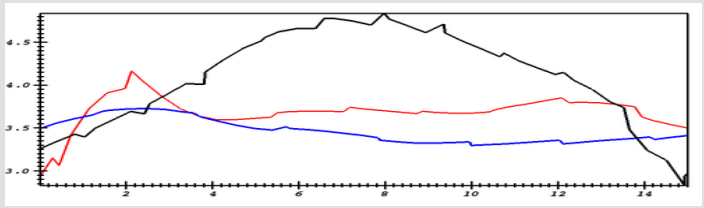
VisIt supports a wide range of use cases.



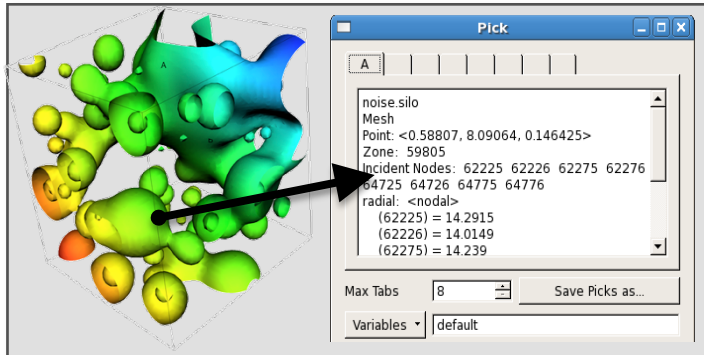
Data Exploration



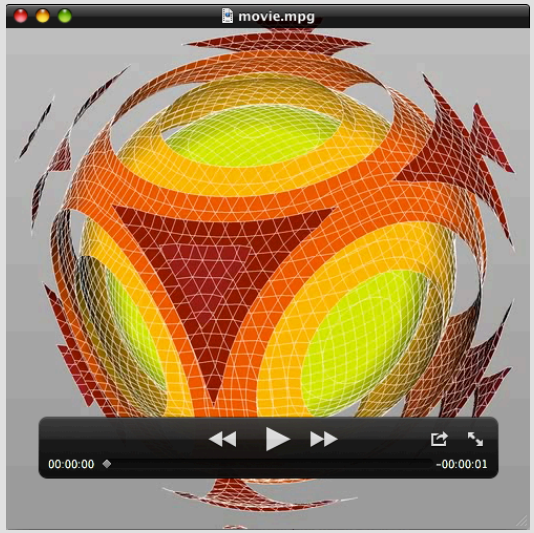
Comparative Analysis



Quantitative Analysis

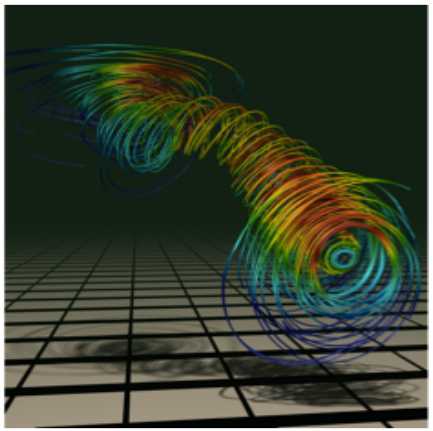


Visual Debugging

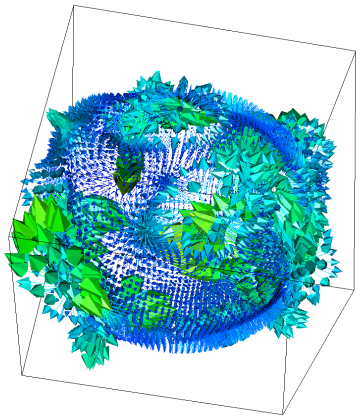


Presentation Graphics

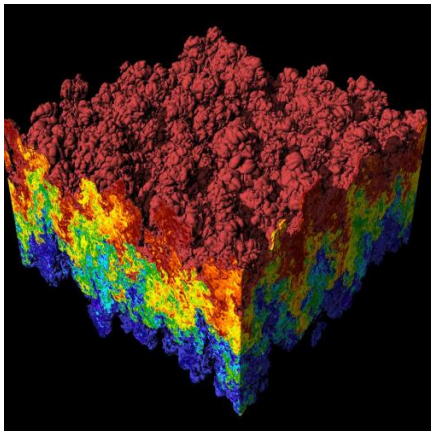
Examples of VisIt's visualization capabilities.



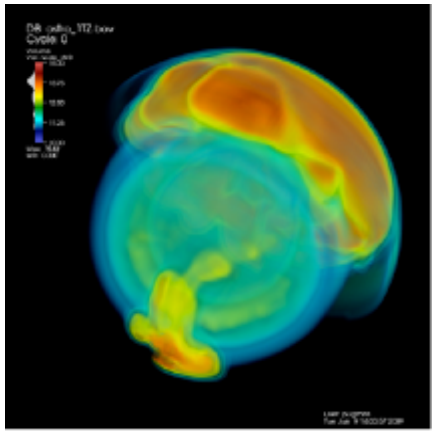
Streamlines



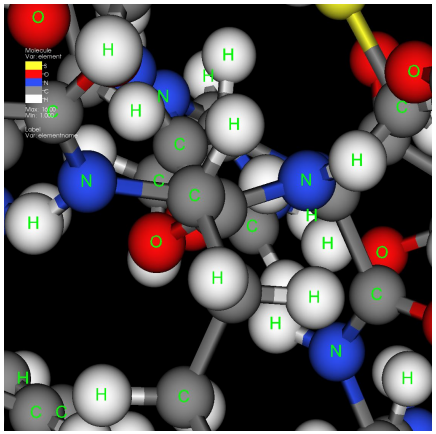
Vector / Tensor Glyphs



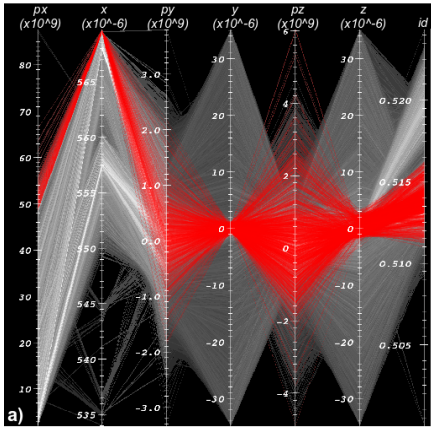
Pseudocolor



Volume Rendering

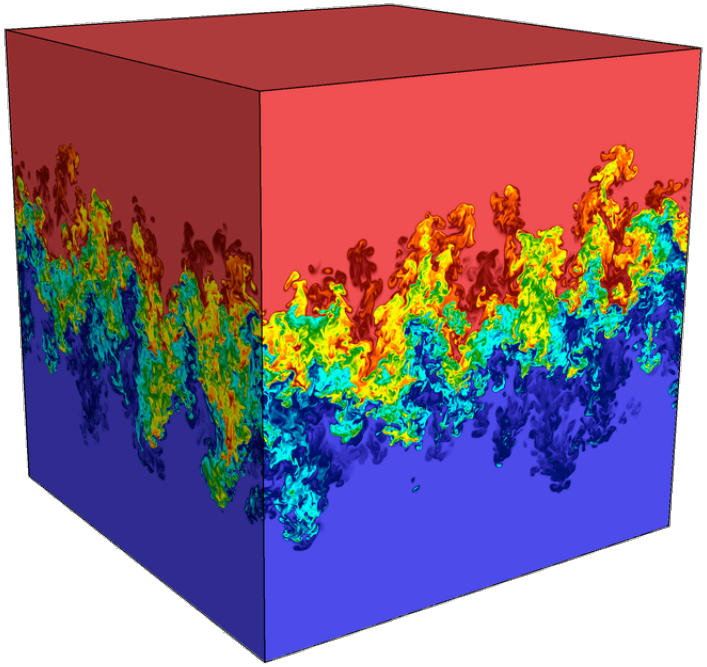


Molecular Visualization

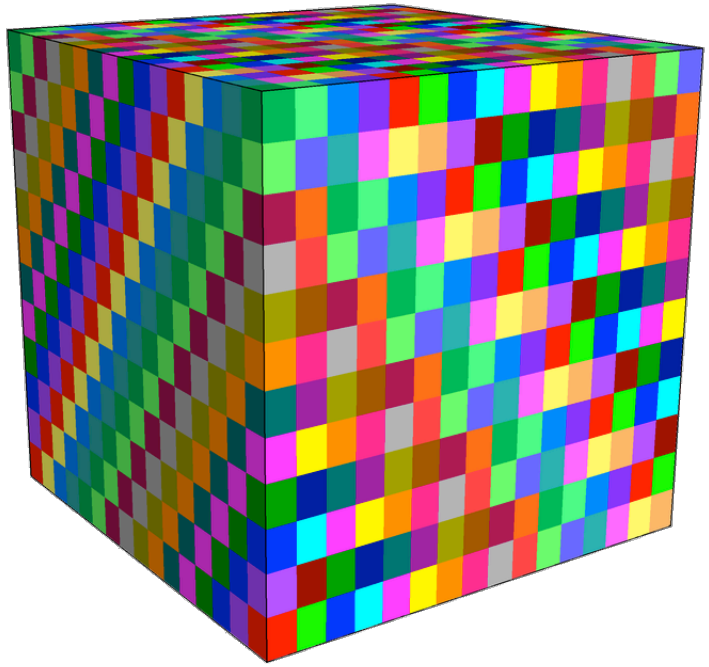


Parallel Coordinates

VisIt uses MPI for distributed-memory parallelism on HPC clusters.



Full Dataset
(27 billion total cells)

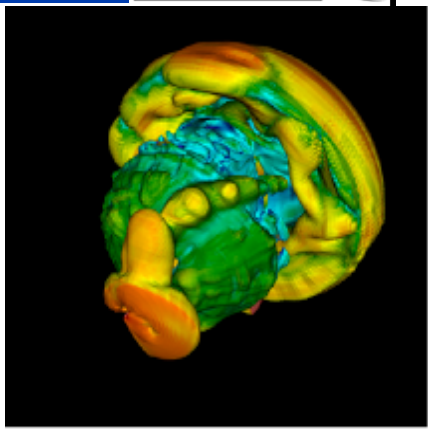


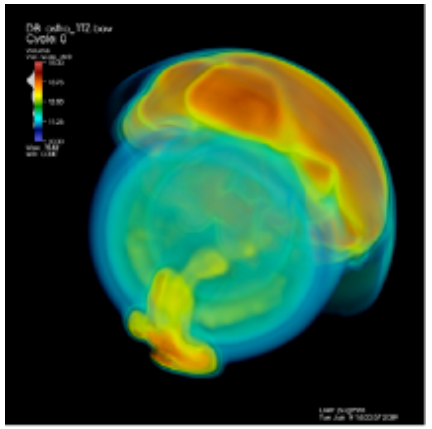
3072 sub-grids
(each 192x129x256 cells)

We are enhancing VisIt's pipeline infrastructure to also support threaded processing.

VisIt scales well on current HPC platforms.

Machine	Architecture	Problem Size	# of Cores
Graph	*X86_64*	***20,001³ (8 T cells)***	*12K*
Dawn	BG/P	15,871³ (4 T cells)	64K
Franklin	Cray XT4	12,596³ (2 T cells)	32K
JaguarPF	Cray XT5	12,596³ (2 T cells)	32K
Juno	X86_64	10,000³ (1 T cells)	16K
Franklin	Cray XT4	10,000³ (1 T cells)	16K
Ranger	Sun	10,000³ (1 T cells)	16K
Purple	IBM P5	8,000³ (0.5 T cells)	8K



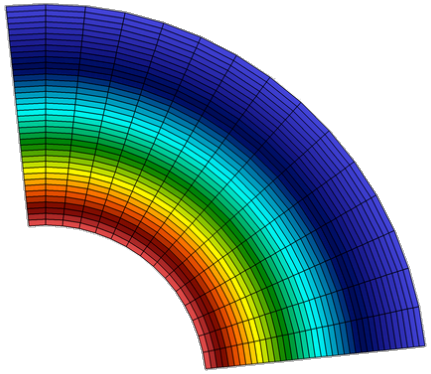
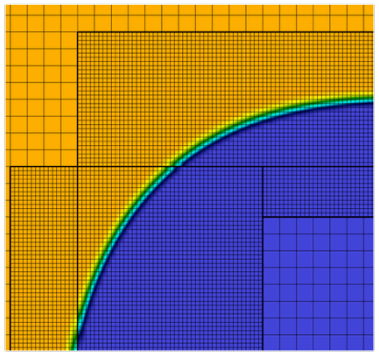
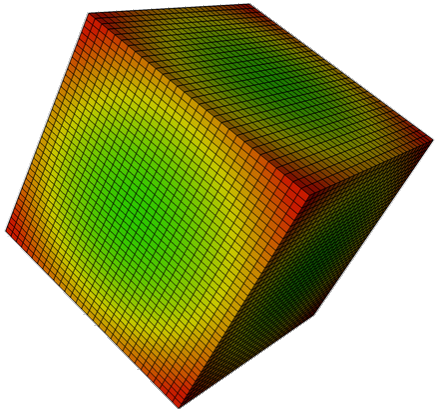
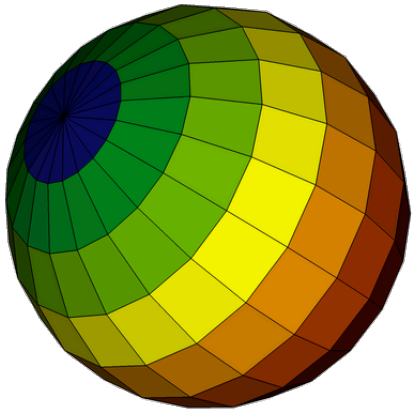


Scaling Studies of Isosurface Extraction and Volume Rendering (2009)

VisIt is also used daily by domain scientists.

VisIt provides a flexible data model, suitable for many application domains.

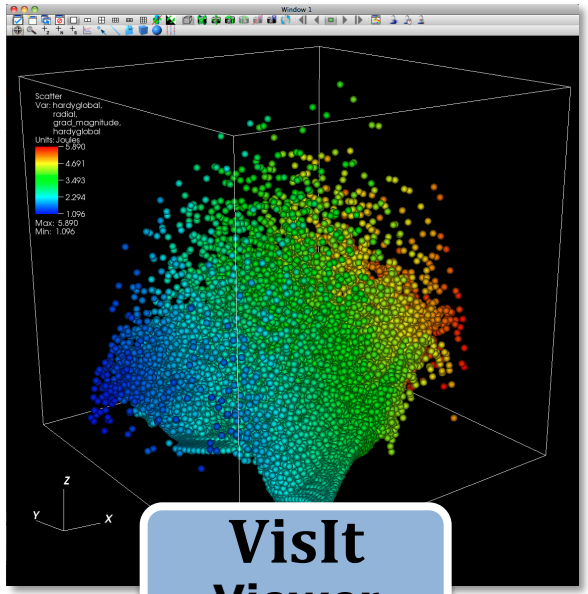
- **Mesh Types:**
  - —Point, Curve, 2D/3D Rectilinear, Curvilinear, Unstructured
  - —Domain Decomposed, AMR
  - —Time Varying
- **Fields:**
  - —Scalar, Vector, Tensor, Material volume fractions, Species

VisIt currently supports over 110 file formats.

VisIt employs a parallelized client-server architecture.

Local Components



VisIt Viewer

VisIt GUI

VisIt CLI

Python Clients

Java Clients

network connection

Parallel Cluster

MPI

VisIt Engine

Data Plugin

Data

VisIt Engine

Data Plugin

Data

VisIt Engine

Data Plugin

Data

(Files or Simulation)

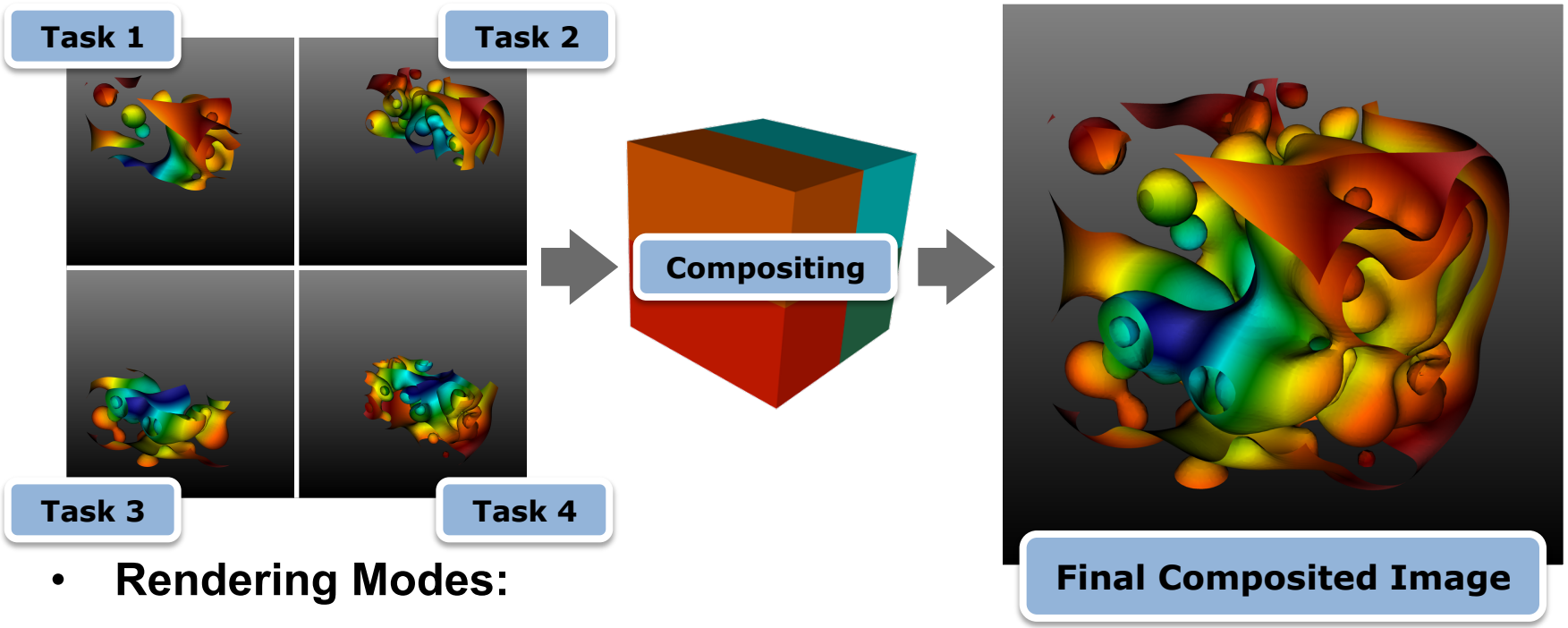
Data Flow Network

Filter

Filter

Filter

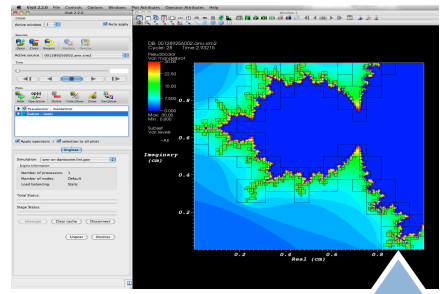
VisIt automatically switches to a scalable rendering…

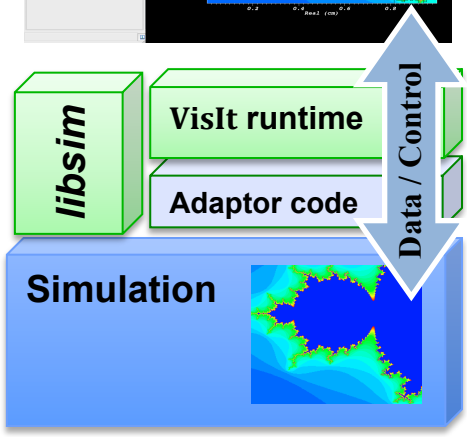


- Rendering Modes:
  - Local (hardware)
  - Remote (software or hardware)
- Beyond surfaces:
  - VisIt also provides scalable volume rendering.

VisIt's infrastructure provides a flexible platform for custom workflows.

- C++ Plugin Architecture
  - —Custom File formats, Plots, Operators
  - —Interface for custom GUIs in Python, C++ and Java
- Python Interfaces
  - —Python scripting and batch processing
  - —Data analysis via Python Expressions and Queries.
- *Libsim* library
  - —Enables coupling of simulation codes to VisIt for in situ visualization.





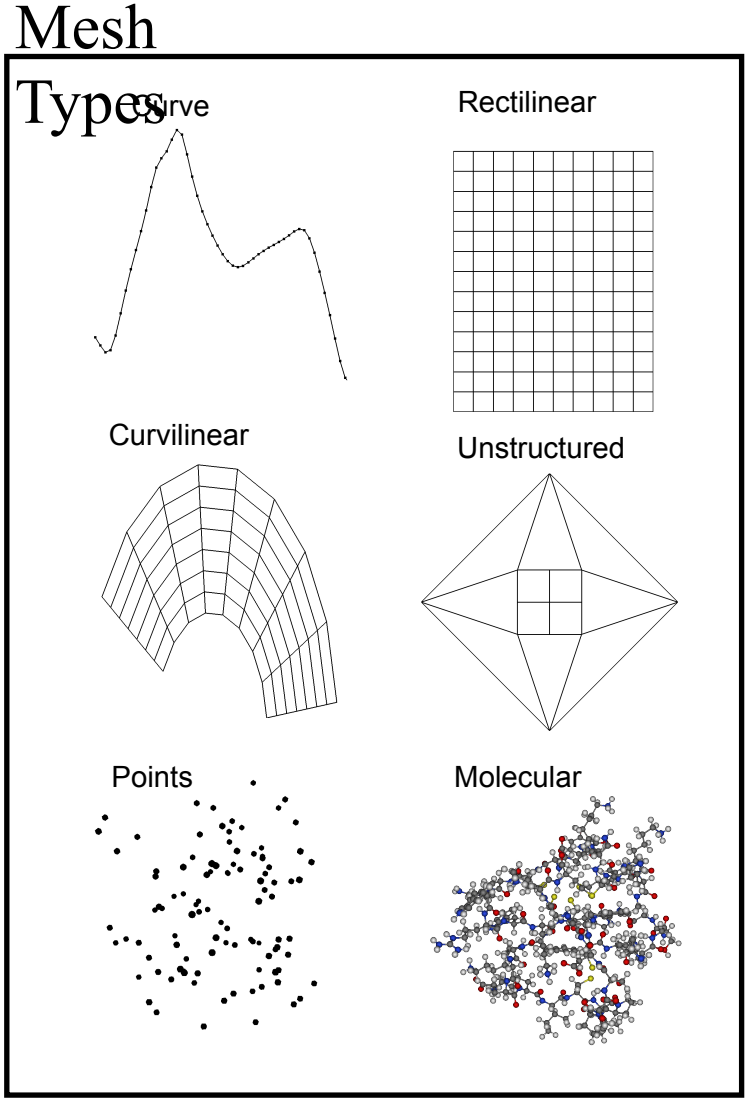
VisIt's Core

Terminology

- **Meshes: discretization of physical space**
  - **—Contains “zones” / “cells” / “elements”**
  - **—Contains “nodes” / “points” / “vertices”**
    - **VisIt speak: zone & node**
- **Fields: variables stored on a mesh**
  - **—Scalar: 1 value per zone/node**
    - **Example: pressure, density, temperature**
  - **—Vector: 3 values per zone/node (direction)**
    - **Example: velocity**
      - Note: 2 values for 2D, 3 values for 3D
  - **—More fields discussed later…**

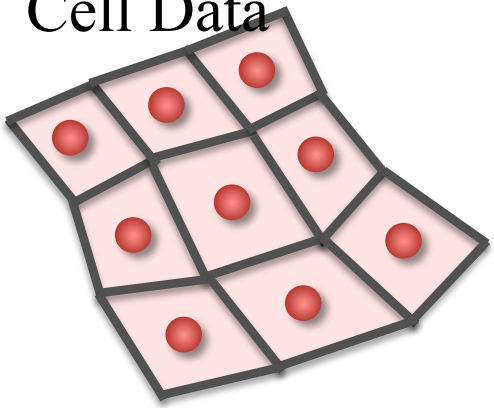
Meshes

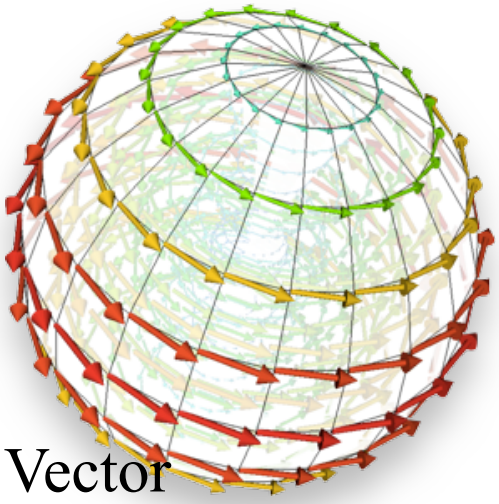
- All data in VisIt lives on a mesh
- Discretizes space into points and cells
  - —(1D, 2D, 3D) + time
  - —Mesh dimension need not match spatial dimension *(e.g. 2D surface in 3D space)*
- Provides a place for data to be located
- Defines how data is interpolated



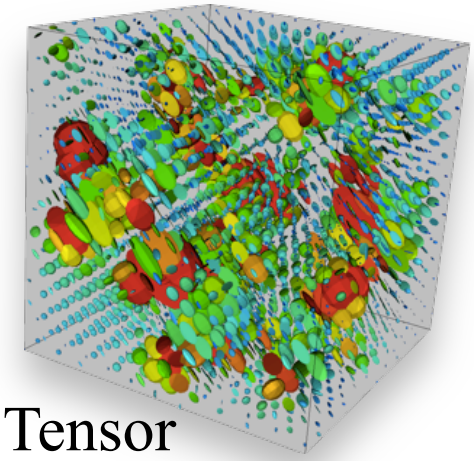
Variables

- **Scalars, Vectors, Tensors**
- **Associated with points or cells of a mesh**
  - **Points: linear interpolation**
  - **Cells: piecewise constant**
- **Can have different dimensionality than the mesh (e.g. 3D vector data on a 2D mesh)**

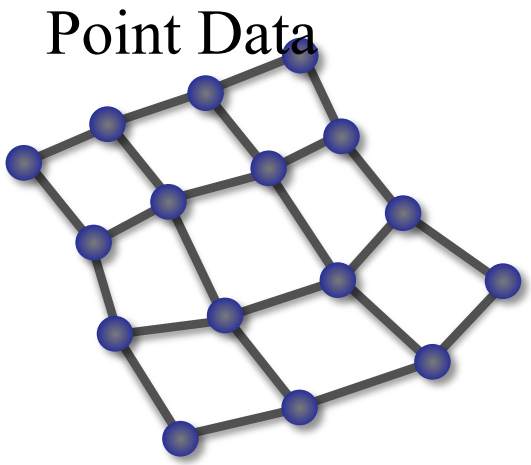




Vector Data

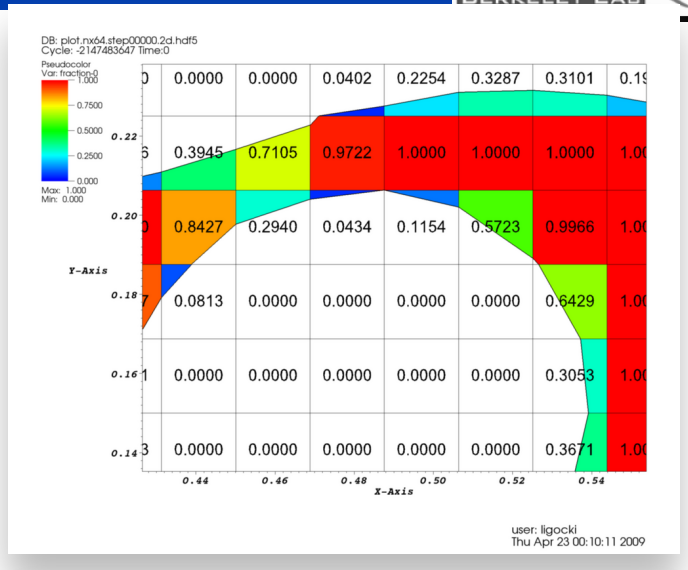


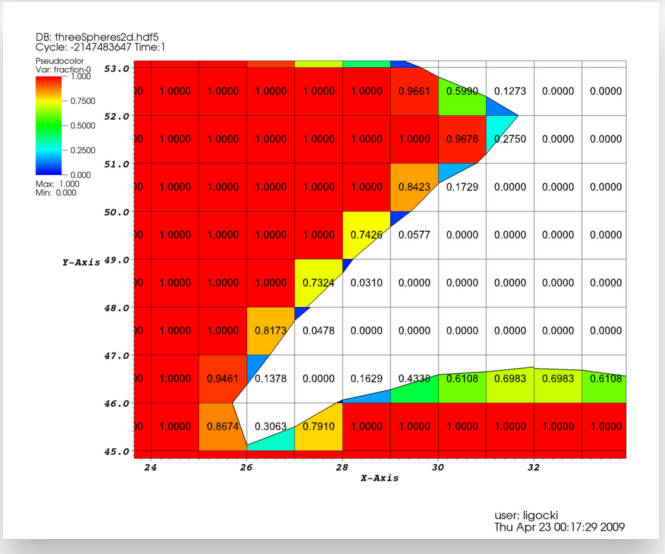
Tensor Data



Materials

- Describes disjoint spatial regions at a sub-grid level
- Volume/area fractions
- VisIt will do high-quality sub-grid material interface reconstruction



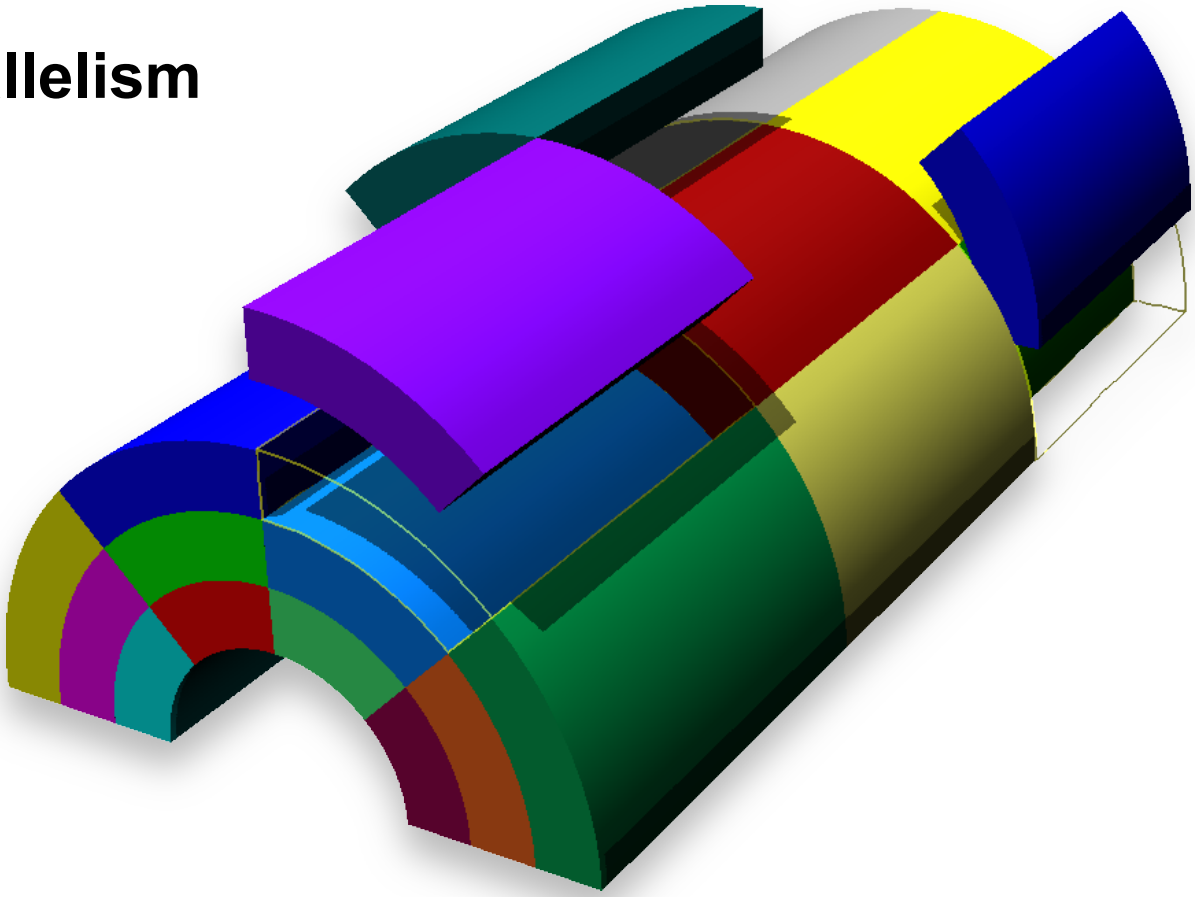


Species

- **Similar to materials, describes sub-grid variable composition**
  - **Example: *Material "Air" is made of species "N_2" ,"O_2", "Ar", "CO_2", etc.***
- **Used for mass fractions**
- **Generally used to weight other scalars (e.g. partial pressure)**

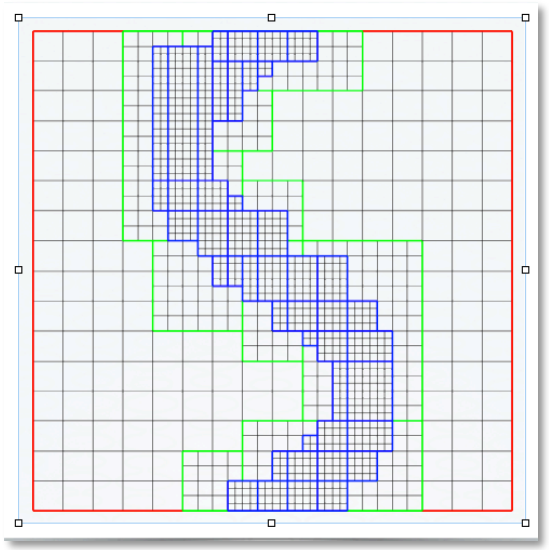
Parallel Meshes

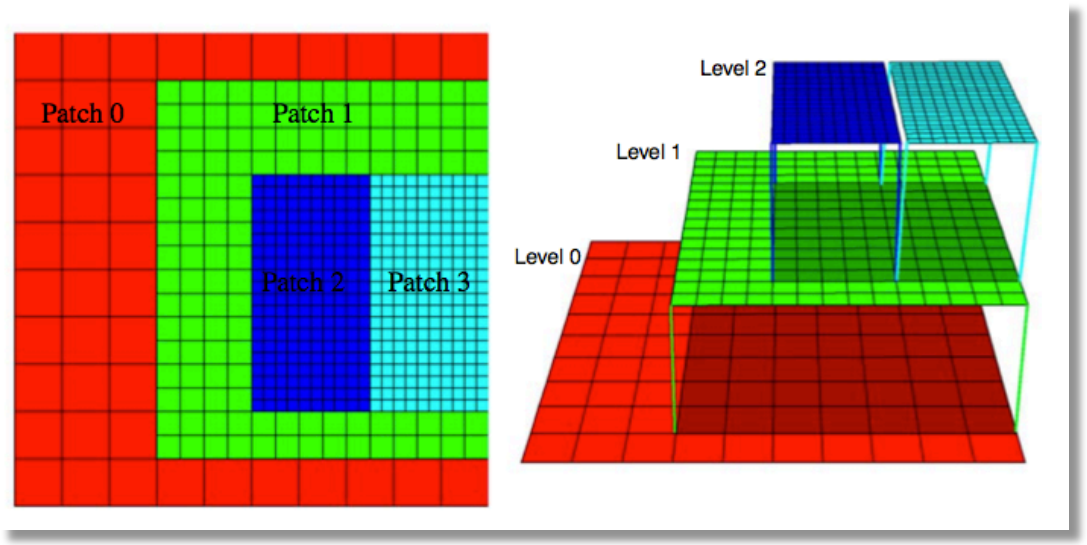
- Provides aggregation for meshes
- A mesh may be composed of large numbers of mesh "blocks"
- Allows data parallelism



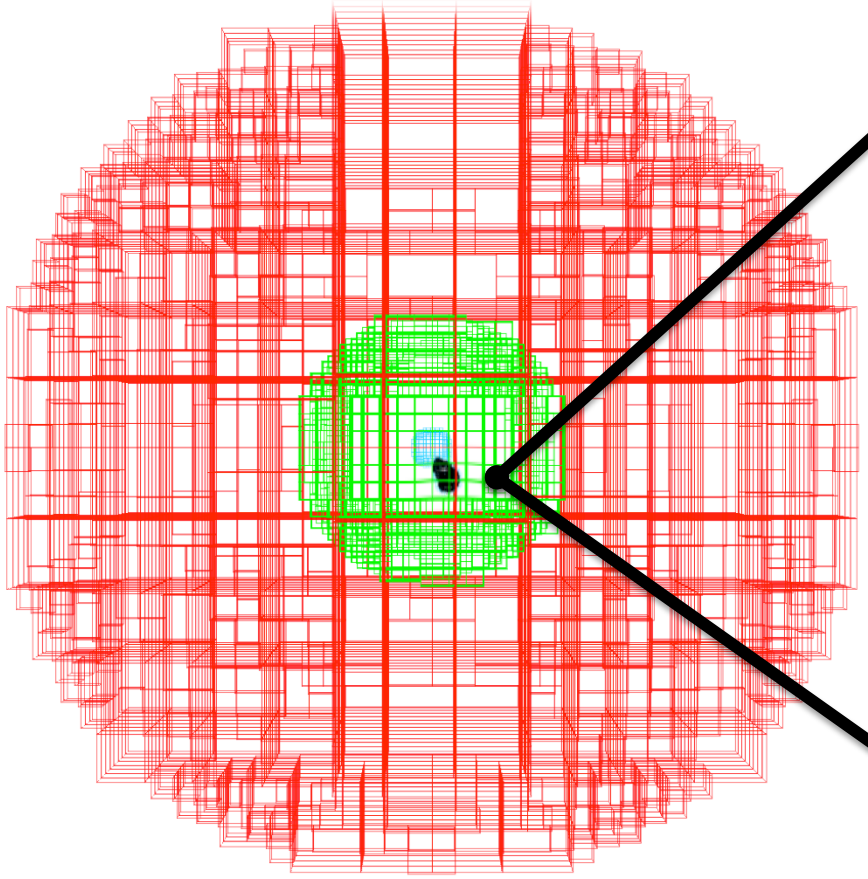
AMR meshes

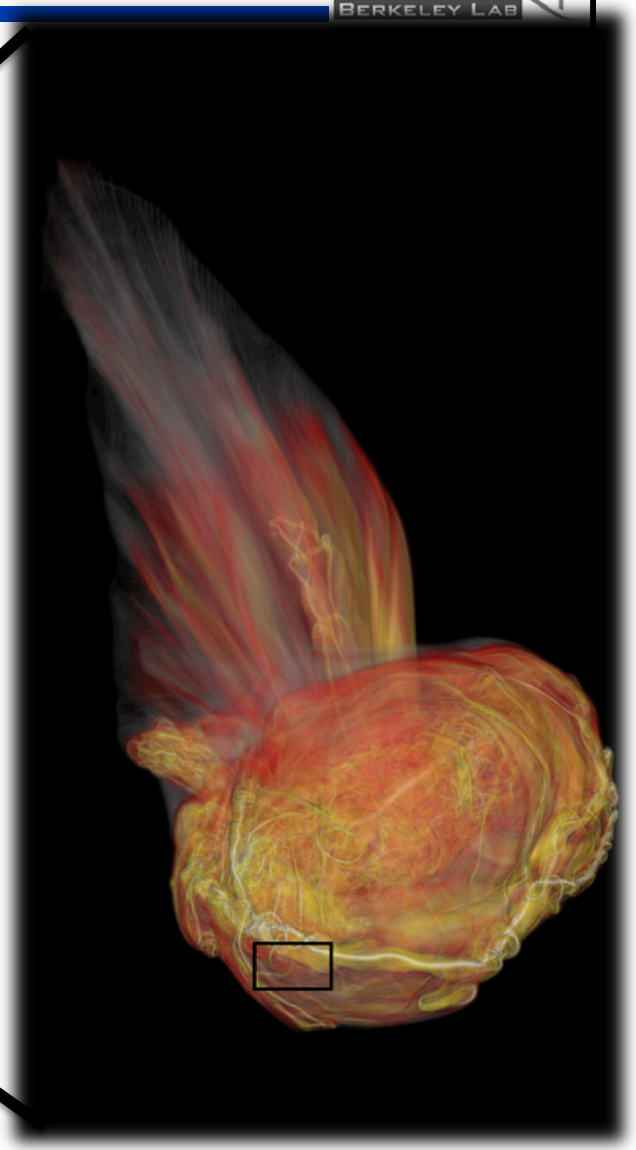
- **Mesh blocks can be associated with patches and levels**
- **Allows for aggregation of meshes into AMR hierarchy levels**

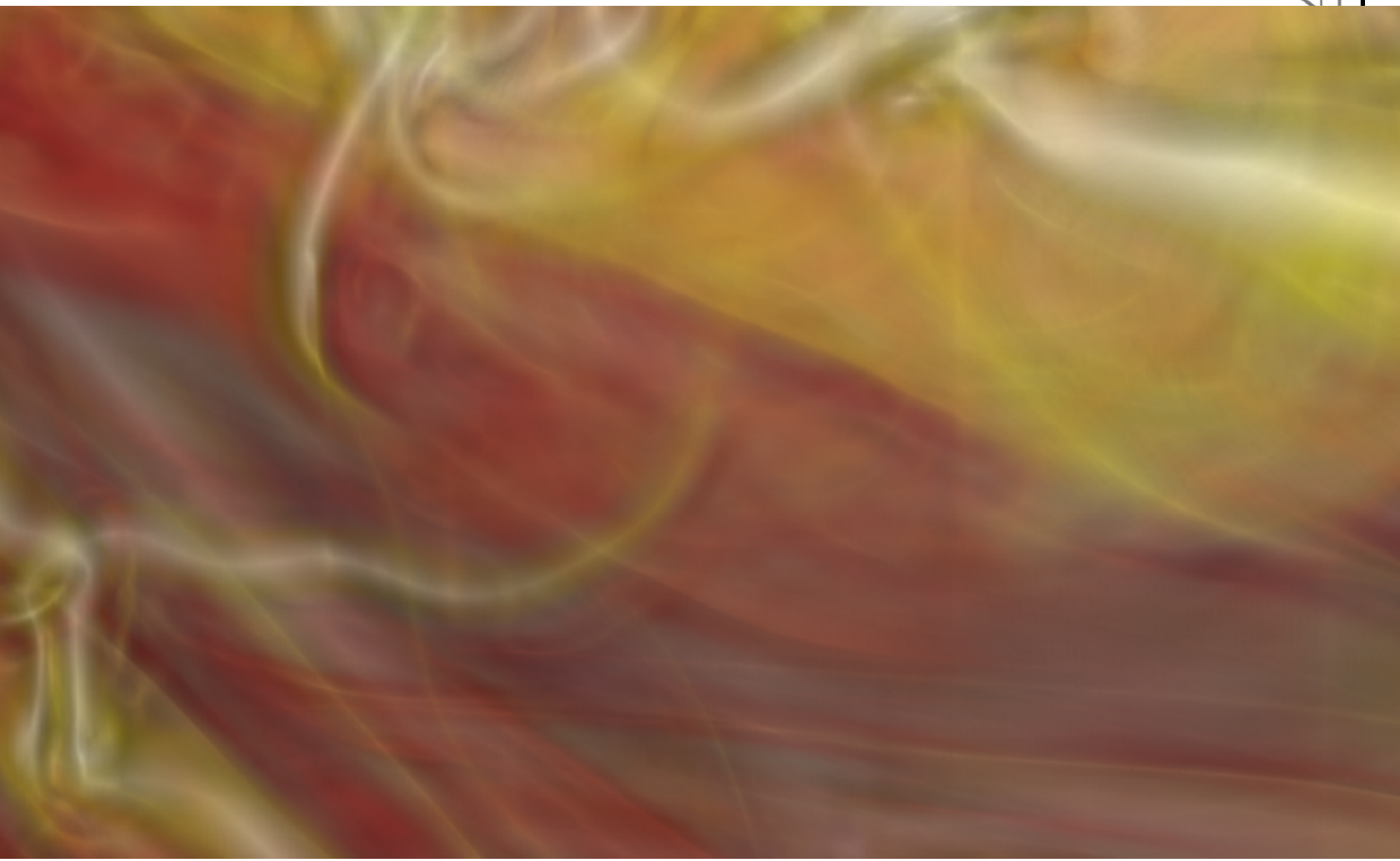




AMR Example: Image vs. Data Resolution







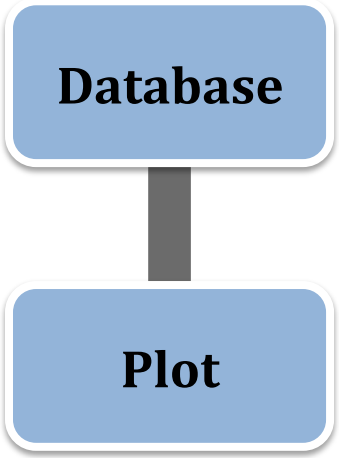
VisIt's Core Abstractions

VisIt's core abstractions

- **Databases**: How datasets are read
- **Plots:** How you render data
- **Operators:** How you manipulate data
- **Expressions:** Mechanism for generating derived quantities
- **Queries:** How to access quantitative information

Examples of VisIt Pipelines

- **Databases: how you read data**
- **Plots: how you render data**
- **Operators: how you transform/manipulate data**
- **Expressions: how you create new fields**
- **Queries: how you pull out quantitative information**

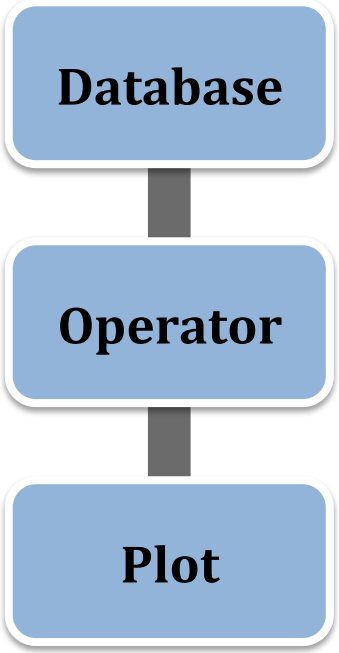


Open a database,
which reads from a file
(example: open file1.hdf5)

Make a plot of a variable in the database
(example: Volume plot)

Examples of VisIt Pipelines

- **Databases: how you read data**
- **Plots: how you render data**
- **Operators: how you transform/manipulate data**
- **Expressions: how you create new fields**
- **Queries: how you pull out quantitative information**



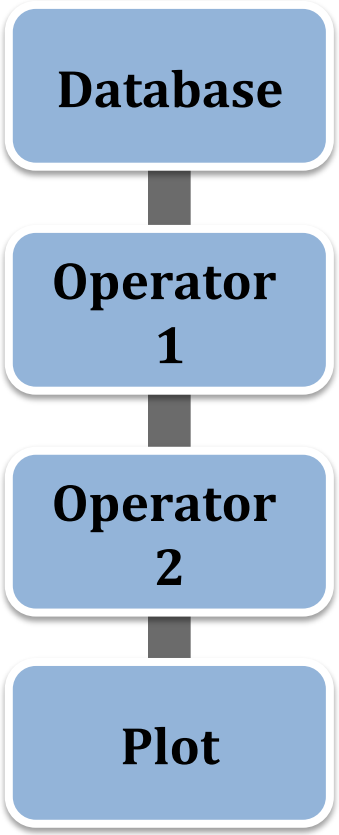
Open a database, which reads from a file (example: open file1.hdf5)

Apply an operator to transform the data (example: Slice operator)

Plot a variable in the database (example: Pseudocolor plot)

Examples of VisIt Pipelines

- **Databases: how you read data**
- **Plots: how you render data**
- **Operators: how you transform/manipulate data**
- **Expressions: how you create new fields**
- **Queries: how you pull out quantitative information**



Open a database, which reads from a file (example: open file1.hdf5)

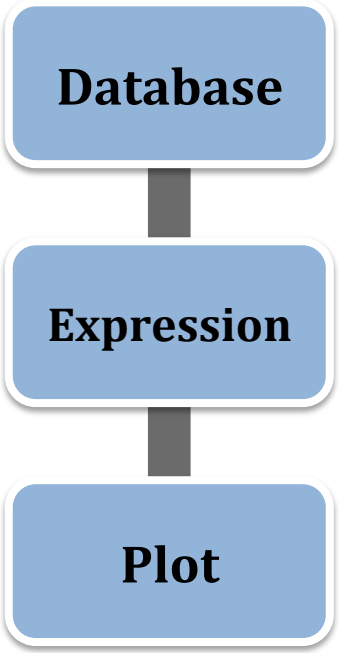
Apply an operator to transform the data (example: Slice operator)

Apply a second operator to transform the data (example: Elevate operator)

Plot a variable in the database (example: Pseudocolor plot)

Examples of VisIt Pipelines

- **Databases: how you read data**
- **Plots: how you render data**
- **Operators: how you transform/manipulate data**
- **Expressions: how you create new fields**
- **Queries: how you pull out quantitative information**



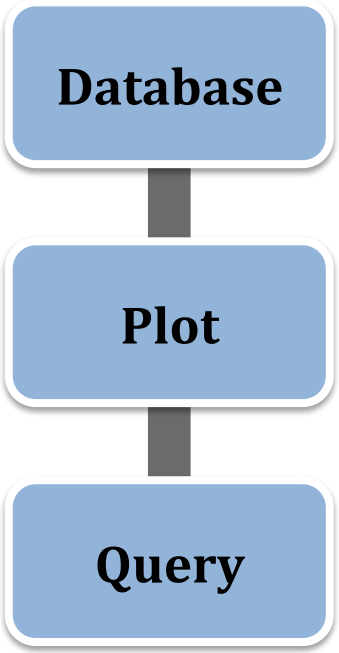
Open a database, which reads from a file (example: open file1.hdf5)

Create derived quantities from fields in the file (ex: magnitude(velocity))

Plot the expression variable (example: Pseudocolor plot)

Examples of VisIt Pipelines

- **Databases: how you read data**
- **Plots: how you render data**
- **Operators: how you transform/manipulate data**
- **Expressions: how you create new fields**
- **Queries: how you pull out quantitative information**



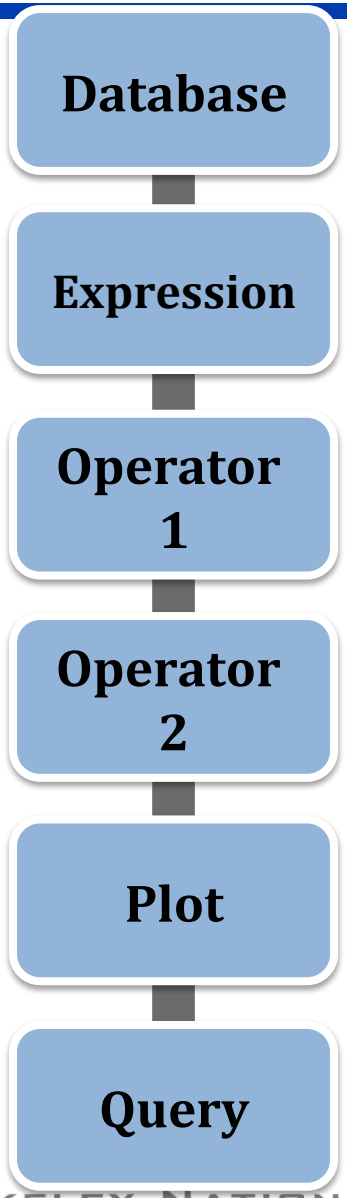
Open a database, which reads from a file (example: open file1.hdf5)

Plot a field from the file (example: density + Pseudocolor plot)

Extract quantitative information (example: integrate density to find mass)

Examples of VisIt Pipelines

- **Databases: how you read data**
- **Plots: how you render data**
- **Operators: how you transform/manipulate data**
- **Expressions: how you create new fields**
- **Queries: how you pull out quantitative information**



Open a database, which reads from a file (ex: open file1.hdf5)

Create derived quantities from fields in the file (ex: magnitude(velocity))

Apply an operator to transform the data (ex: Slice operator)

Apply a second operator to transform the data (ex: Elevate operator)

Plot a field (ex: speed + pseudocolor plot)

Extract quantitative information (ex: max speed over cross-section)

VisIt's core building blocks

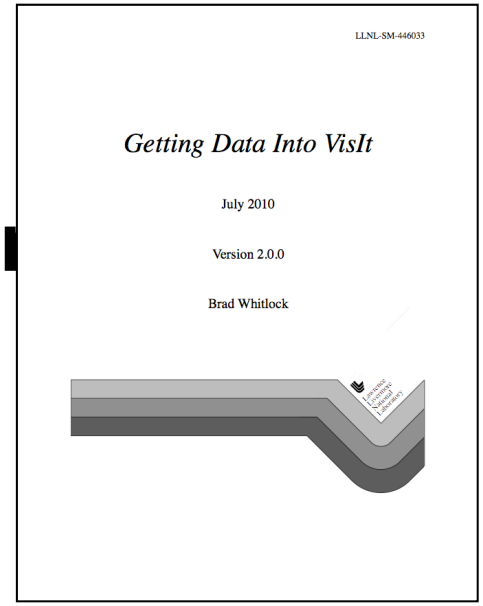
- Databases: How datasets are read
- Plots: How you render data
- Operators: How you manipulate data
- Expressions: Mechanism for generating derived quantities
- Queries: How to access quantitative information

How to get VisIt to read your data.

- There is an extensive manual on this topic: "Getting Data Into VisIt"

https://wci.llnl.gov/simulation/computer-codes/visit/manuals

- Three ways:
  - —Use a known format
  - —Write a file format reader
  - —In situ processing


LLNL-SM-446033

Getting Data Into VisIt

July 2010

Version 2.0.0

Brad Whitlock

File formats that VisIt supports

- 110+ Total Readers: ADIOS, BOV, Boxlib, CCM, CGNS, Chombo, CLAW, EnSight, ENZO, Exodus, FLASH, Fluent, GDAL, Gadget, Images (TIFF, PNG, etc), ITAPS/MOAB, LAMMPS, NASTRAN, NETCDF, Nek5000, OpenFOAM, PLOT3D, PlainText, Pixie, Shapefile, Silo, Tecplot, VTK, Xdmf, Vs, and many more

http://www.visitusers.org/index.php?title=Detailed_list_of_file_formats_VisIt_supports

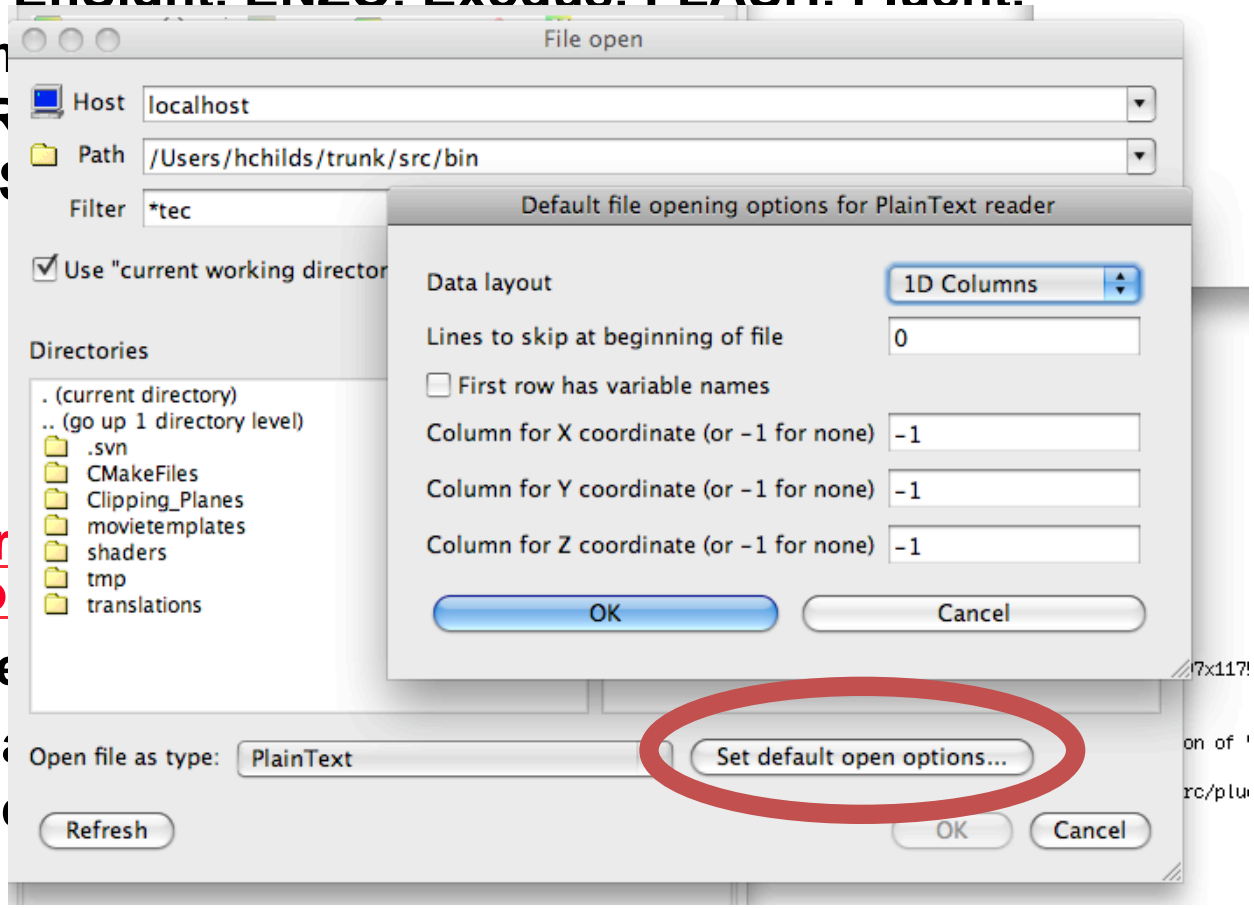
- Some readers are more robust than others.
  - For some formats, support is limited to flavors of a file a VisIt developer has encountered previously (e.g. Tecplot).

File formats that VisIt supports

- 110+ Total Readers: ADIOS, BOV, Boxlib, CCM, CGNS, Chombo, CLAW, EnSight, ENZO, Exodus, FLASH, Fluent, GDAL, Gadget, In
  LAMMPS, NASTR
  PlainText, Pixie, S
  many more

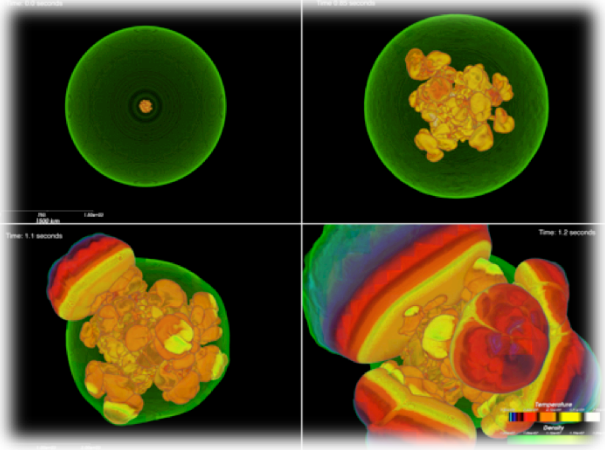
http://www.visituser
title=Detailed_list_o

- Some readers are
  - For some forma
    developer has

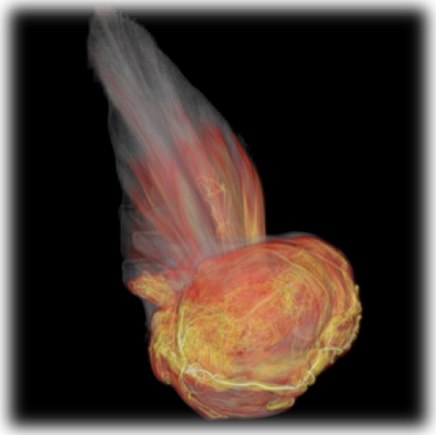


Application Code Formats

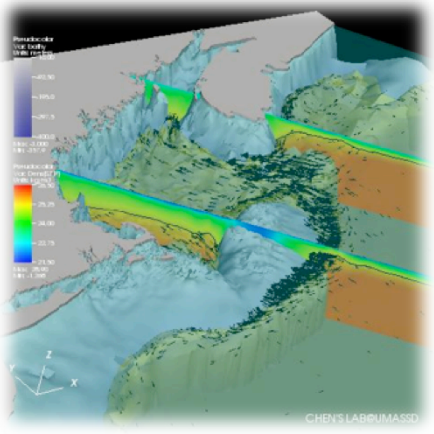
- ANSYS
- Cale
- CASTRO
- CCM
- DDCMD
- Dyna3D
- Enzo
- FLASH
- FVCOM
- Gadget
- LAMMPS
- NASTRAN
- Nek5000
- OVERFLOW
- PATRAN
- Pixie
- S3D
- ZeusMP



FLASH



CASTRO

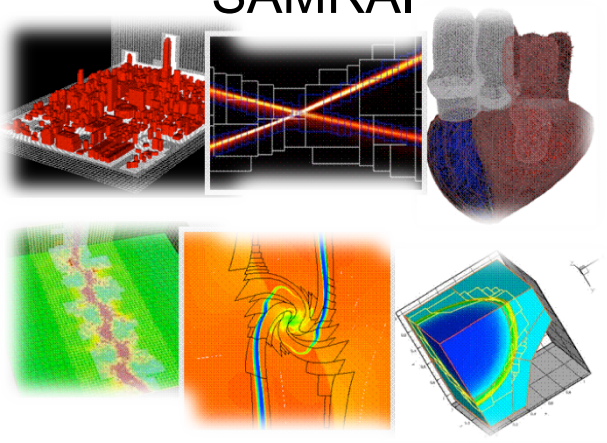


FVCOM

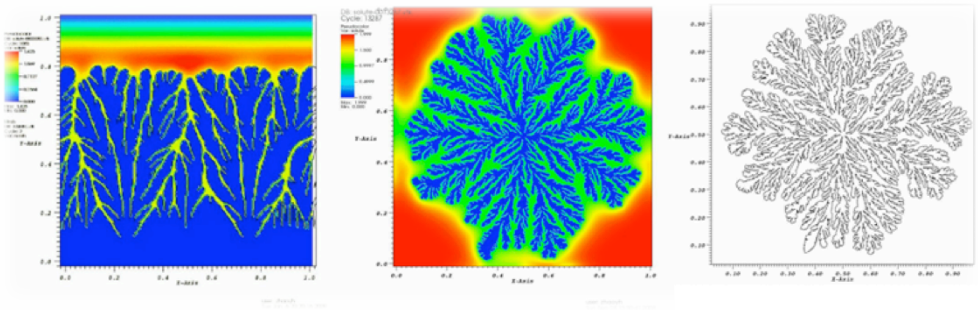
Application Toolkit Formats

- Adventure I/O
- BoxLib
- Chombo
- ITAPS
- OpenFOAM
- SAMRAI
- Spheral

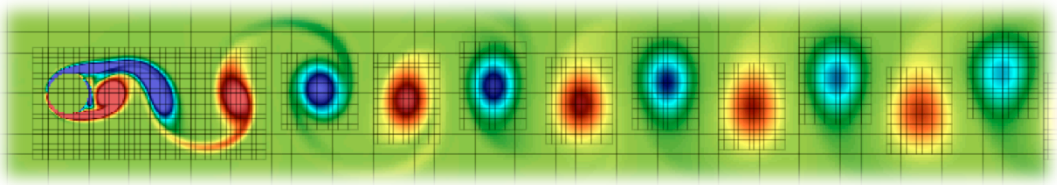
SAMRAI



ITAPS



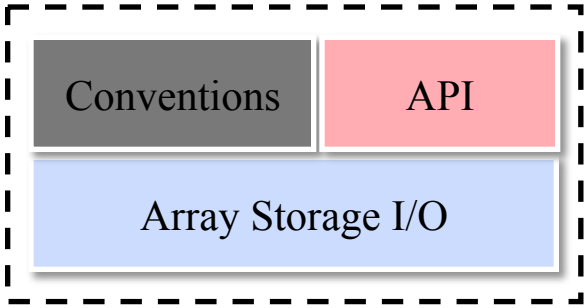
Chombo



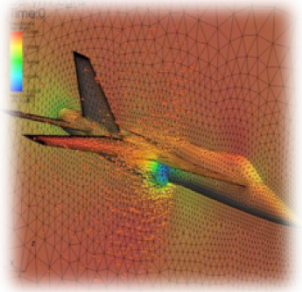
General Scientific Data Formats

- ADIOS
- CGNS
- Exodus
- HDF5
- H5Part
- NETCDF
- PDB
- Silo
- XDMF

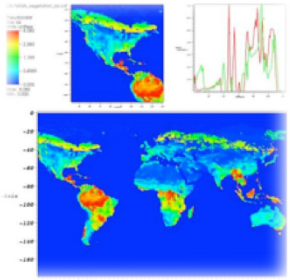
Common Structure



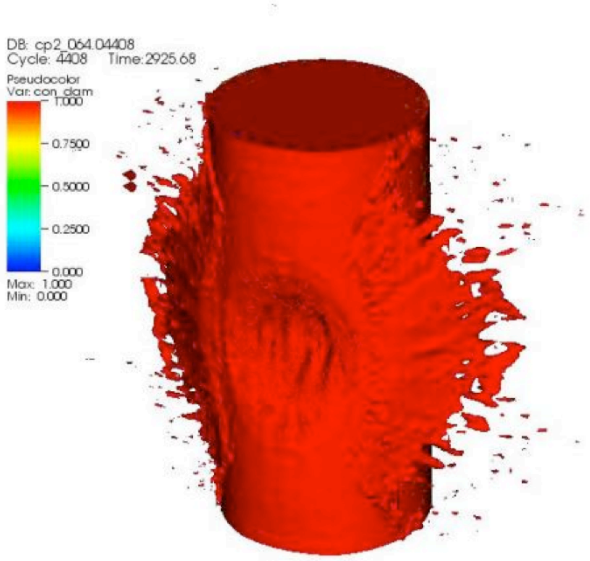
CGNS



NETCDF



Silo / Ale3d



File formats that VisIt supports

- Common array writing libraries:
  - NETCDF
    - VisIt reader understands many (but not all) conventions
  - HDF5
    - Pixie is most general HDF5 reader
    - Many other HDF5 readers
- Xdmf: specify an XML file that describes semantics of arrays in HDF5 file
- VizSchema (Vs): add attributes to your HDF5 file that describes semantics of the arrays.

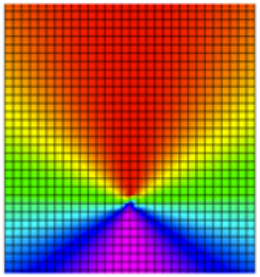
Silo

WCI | B Codes – SILO

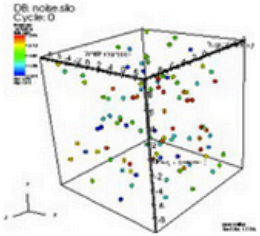
https://wci.llnl.gov/codes/silo/

Welcome to Silo

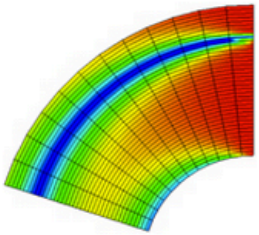
A mesh and field I/O library and scientific database



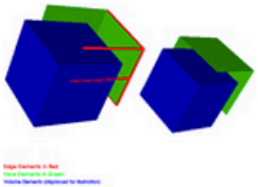
Structured Rectilinear Mesh



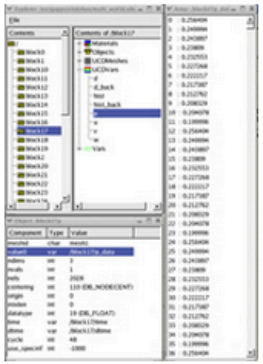
Gridless Point Mesh



Structured (Curvilinear) Mesh

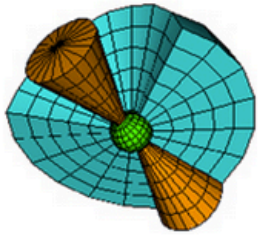


Arbitrary Subsets

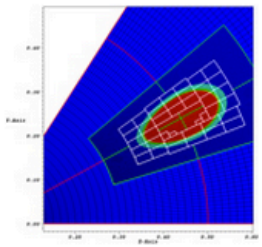


Silex browser for Silo files

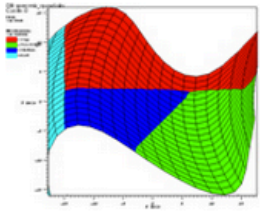
Constructive Solid Geometry (CSG) Mesh



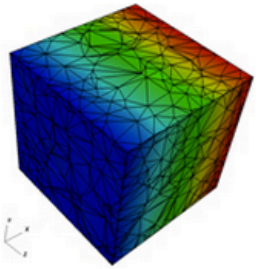
Unstructured Zoo (UCD) Mesh



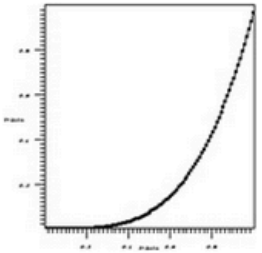
Adaptive Mesh Refinement (AMR) Mesh



Mixing Materials



Arbitrary Polyhedral Mesh

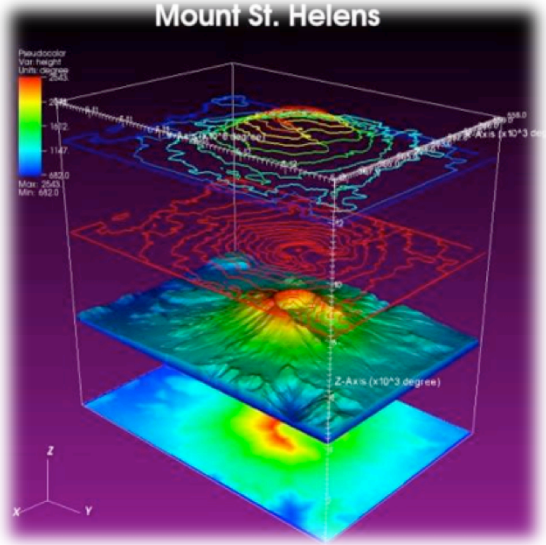


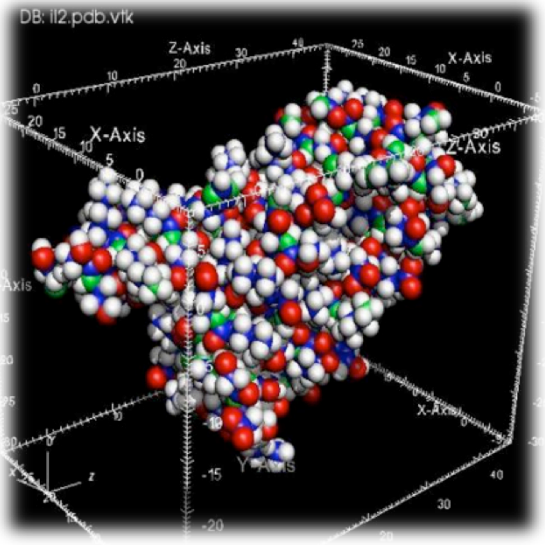
XY Curve

Specialized Scientific Data Formats

- BOW
- FITS
- GDAL
- MatrixMarket
- ProteinDataBank
- ESRI Shapefile
- XYZ

DEM from GDAL

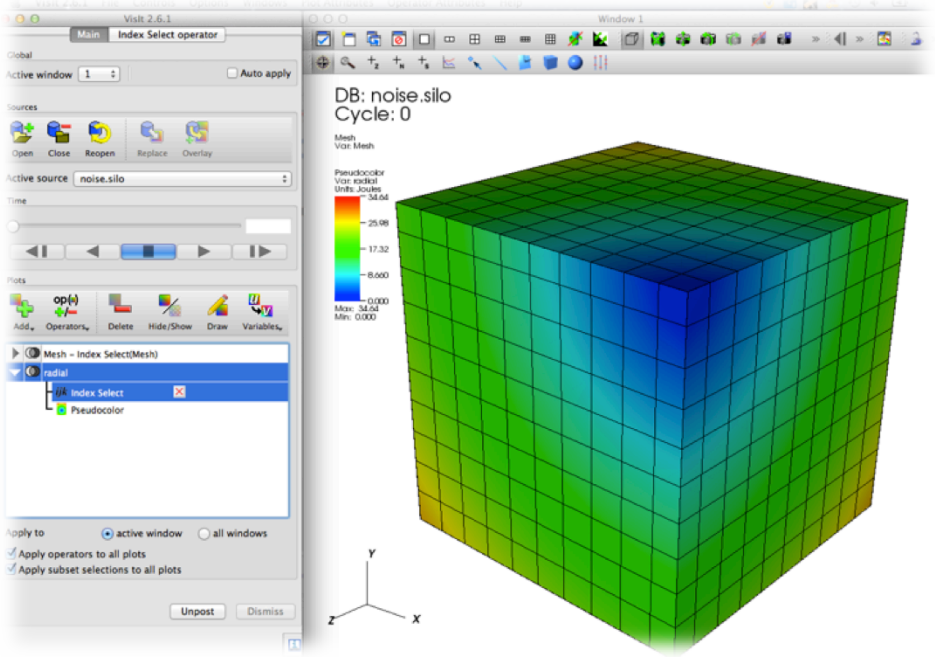




Protein Data Bank

Visualization Formats

- VTK
- EnSight
- GMV
- Plot3D
- Tecplot
- Vis5D
- Xmdv



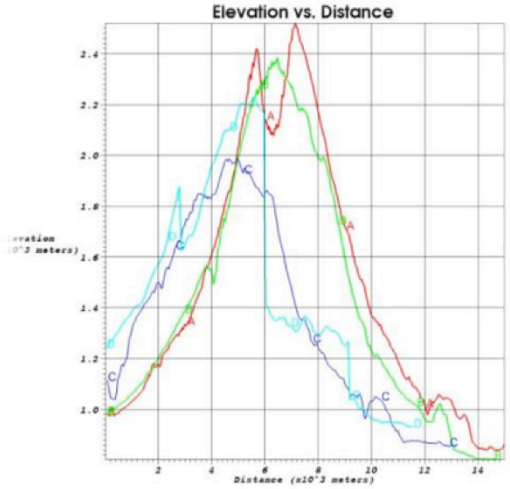
Graphics Formats

- Image
  - (PNG, JPEG, TIFF, BMP, etc.)
- RAW
- STL
- Wavefront OBJ

Carina Nebula

General ASCII Data Formats

- Curve2D
- Lines
- PlainText
- Point3D



	i=0	i=1	i=2	i=3	i=4	i=5
j=7	2.517243	2.550414	2.581495	2.609803	2.634335	2.653569
j=6	2.472034	2.503052	2.531701	2.557125	2.578064	2.592629
j=5	2.427398	2.456259	2.482482	2.505081	2.522616	2.532976
j=4	2.383583	2.410415	2.434426	2.454567	2.469347	2.476640
j=3	2.340819	2.365857	2.388012	2.406262	2.419193	2.424868
j=2	2.299279	2.322814	2.343538	2.360524	2.372542	2.377986
j=1	2.259063	2.281395	2.301101	2.317398	2.329294	2.335572
j=0	2.220195	2.241595	2.260633	2.276686	2.289012	2.296766

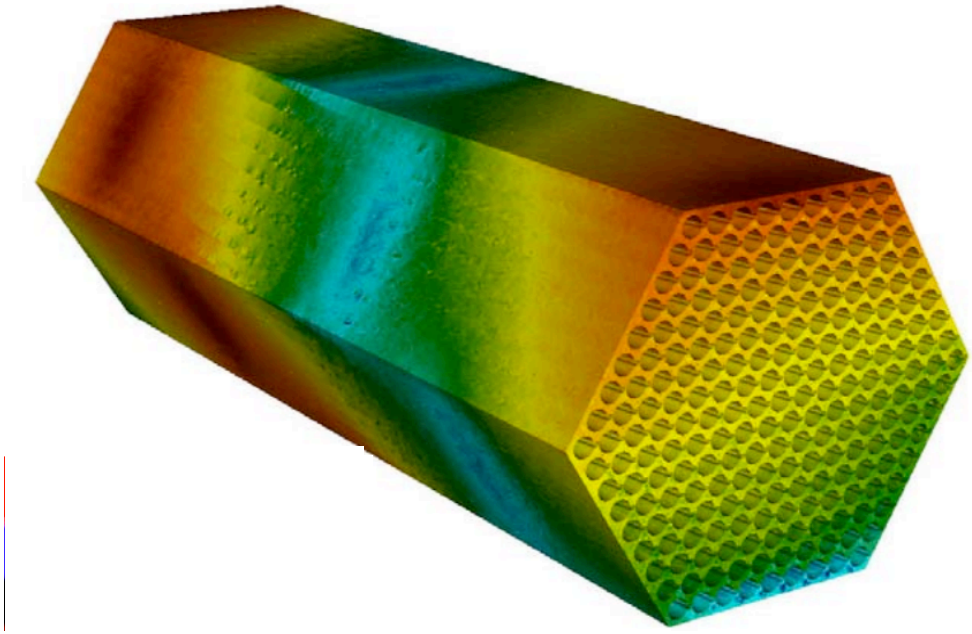
```
2.72727272727 7.57851239669
2.77777777778 7.78703703704
2.82828282828 7.99938781757
2.87878787879 8.21556473829
2.92929292929 8.4355677992
2.9797979798 8.65939700031
3.0303030303 8.8870523416
3.08080808081 9.11853382308
3.13131313131 9.35384144475
3.18181818182 9.59297520661
3.23232323232 9.83593510866
3.28282828283 10.0827211509
3.33333333333 10.3333333333
3.38383838384 10.587771656
3.43434343434 10.8460361188
3.48484848485 11.1081267218
3.53535353535 11.374043465
3.58585858586 11.6437863483
3.63636363636 11.9173553719
3.68686868687 12.1947505357
3.73737373737 12.4759718396
```

VisIt's core building blocks

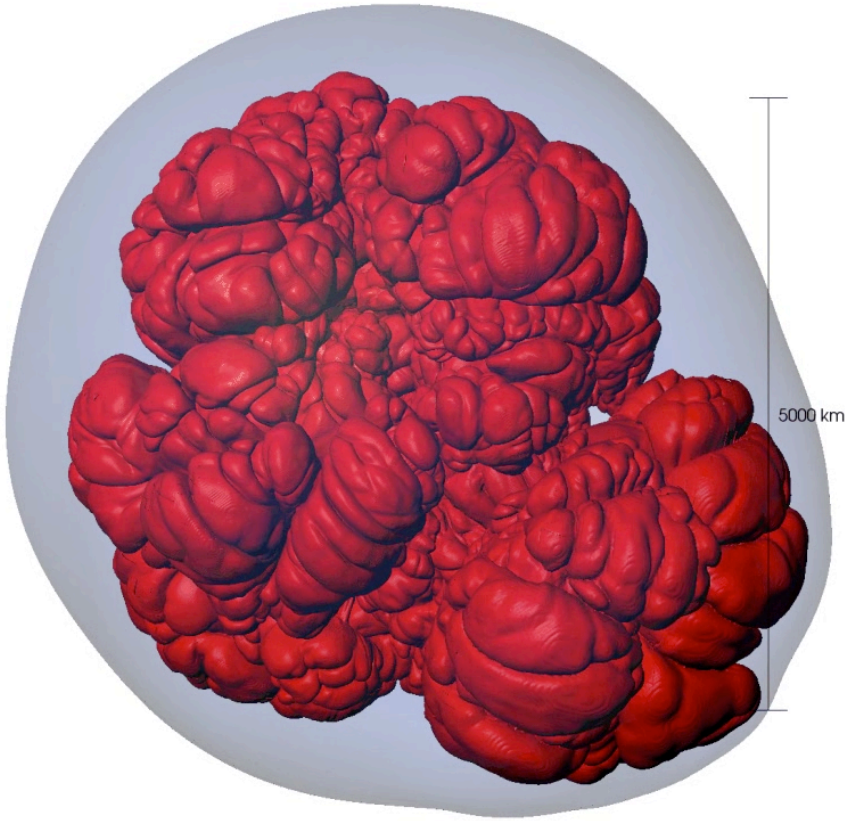
- Databases: How datasets are read
- Plots: How you render data
- Operators: How you manipulate data
- Expressions: Mechanism for generating derived quantities
- Queries: How to access quantitative information

Pseudocolor Rendering

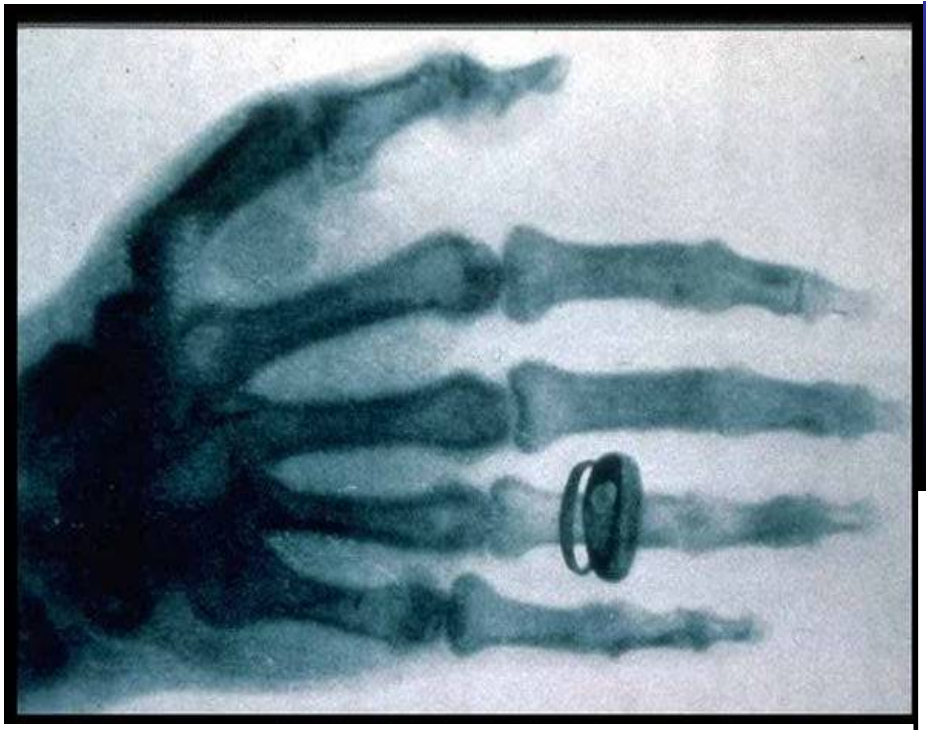
- **Maps scalar fields (e.g., density, pressure, temperature) to colors.**

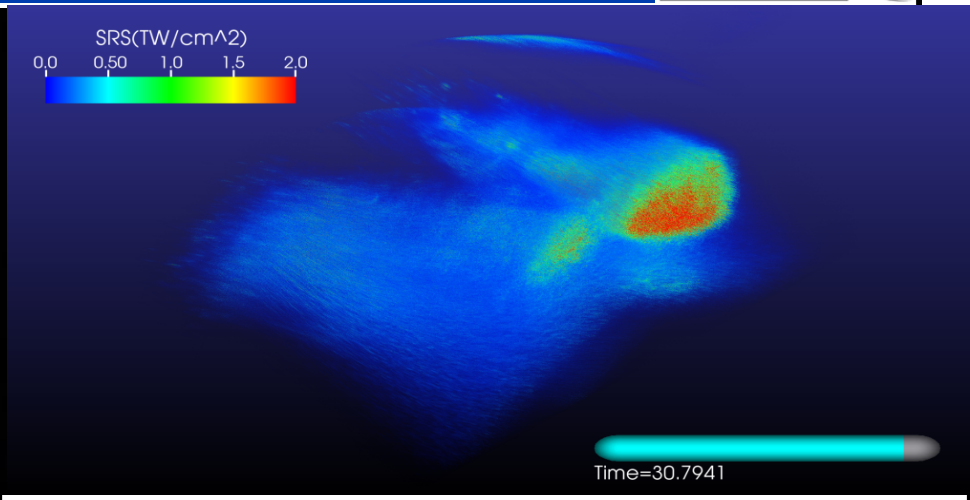


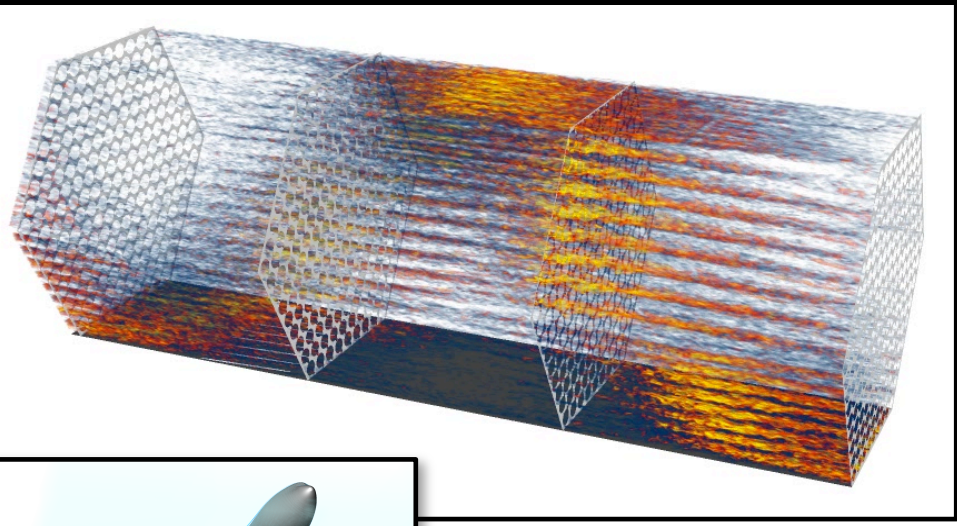
Contour / Isosurface Rendering

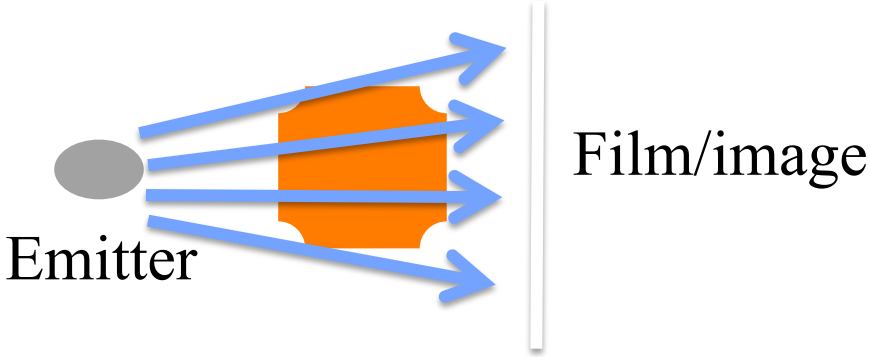


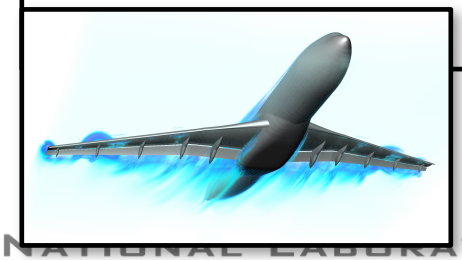
Volume rendering







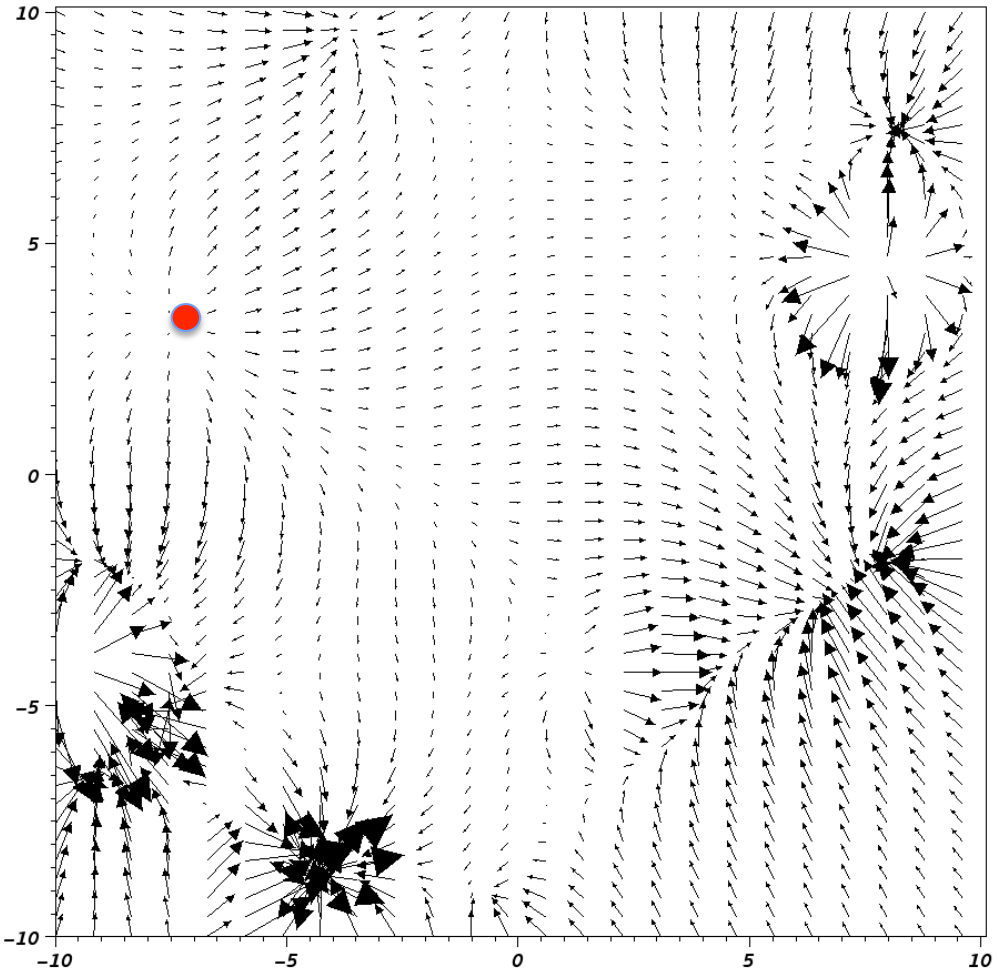




VisIt can combine volume rendering and opaque geometry

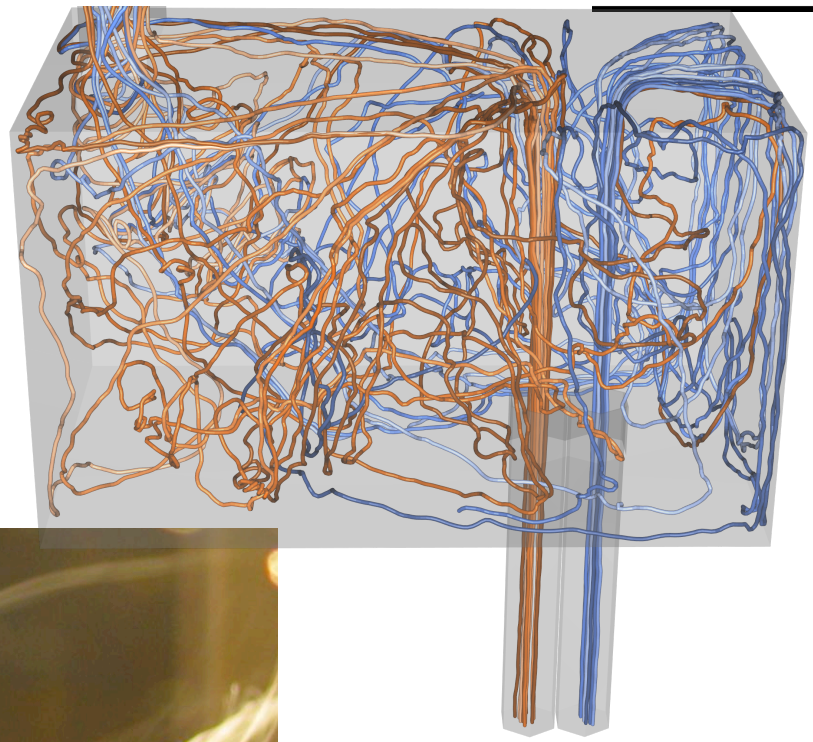
Particle advection: the foundation of flow visualization

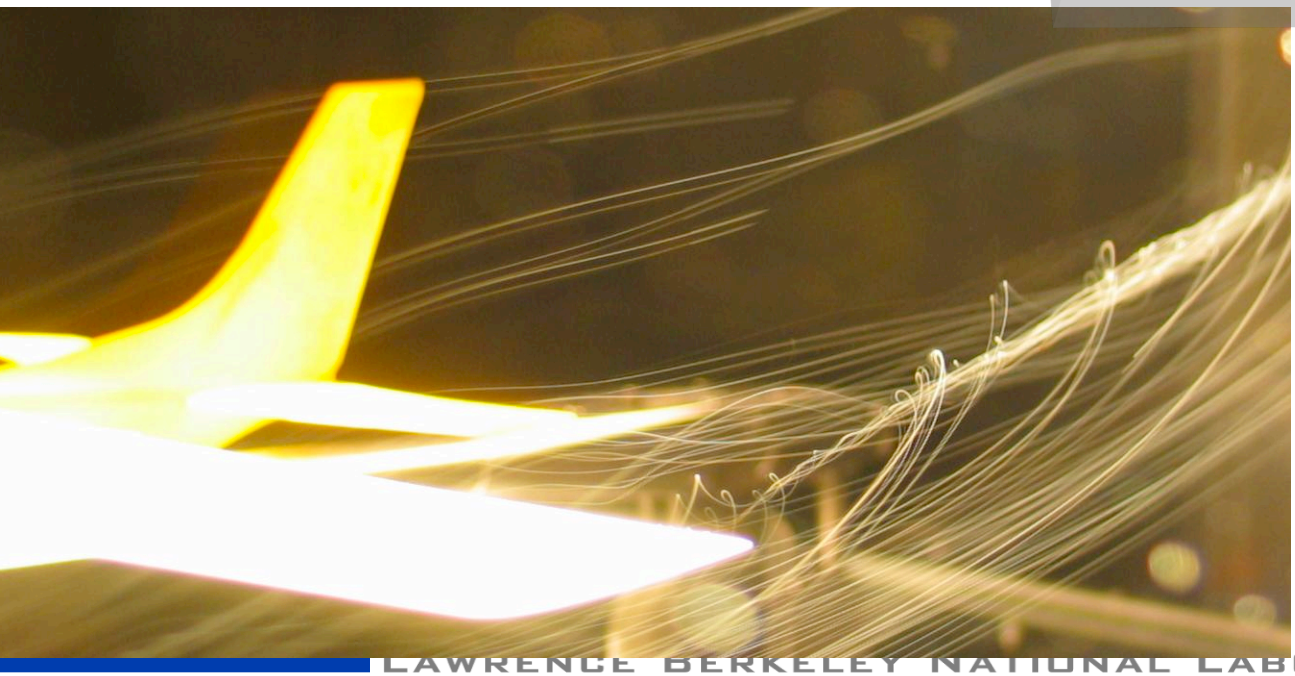
- Displace massless particle based on velocity field
- S(t) = position of curve at time t
  - —$S(t_0) = p_0$
    - t_0: initial time
    - p_0: initial position
  - —$S'(t) = v(t, S(t))$
    - v(t, p): velocity at time t and position p
    - S'(t): derivative of the integral curve at time t



This is an ordinary differential equation

Streamlines

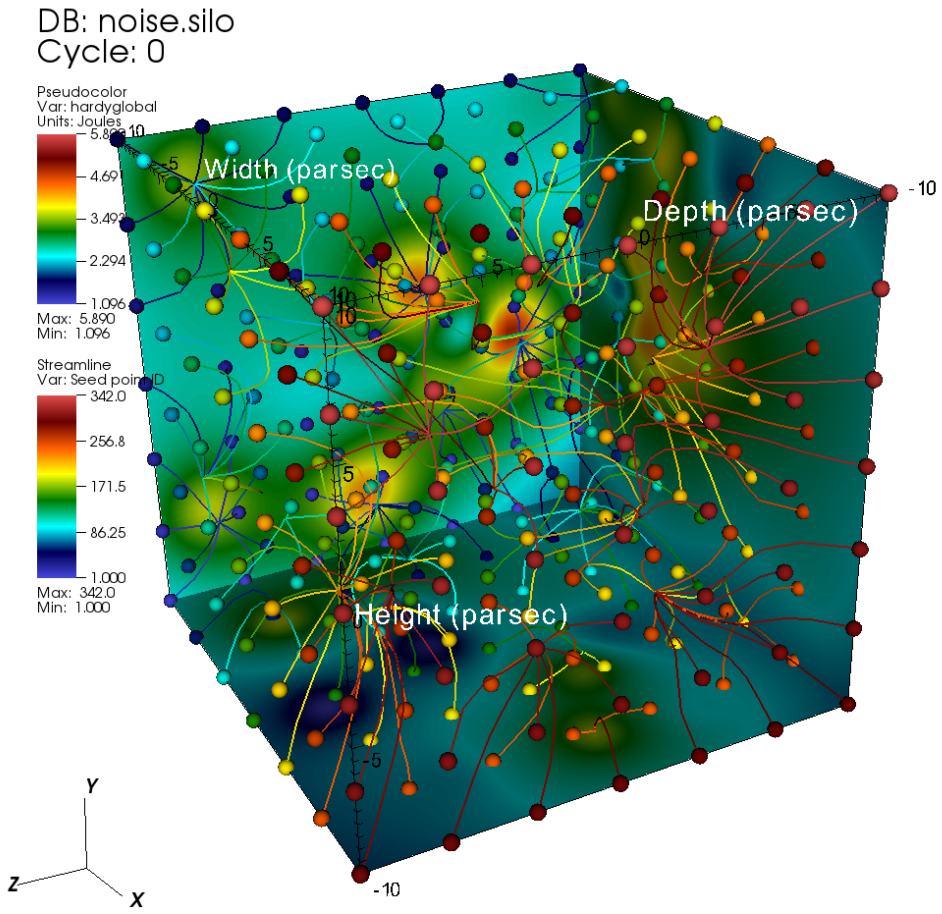




There are several ways to access VisIt's Python Client Interface.

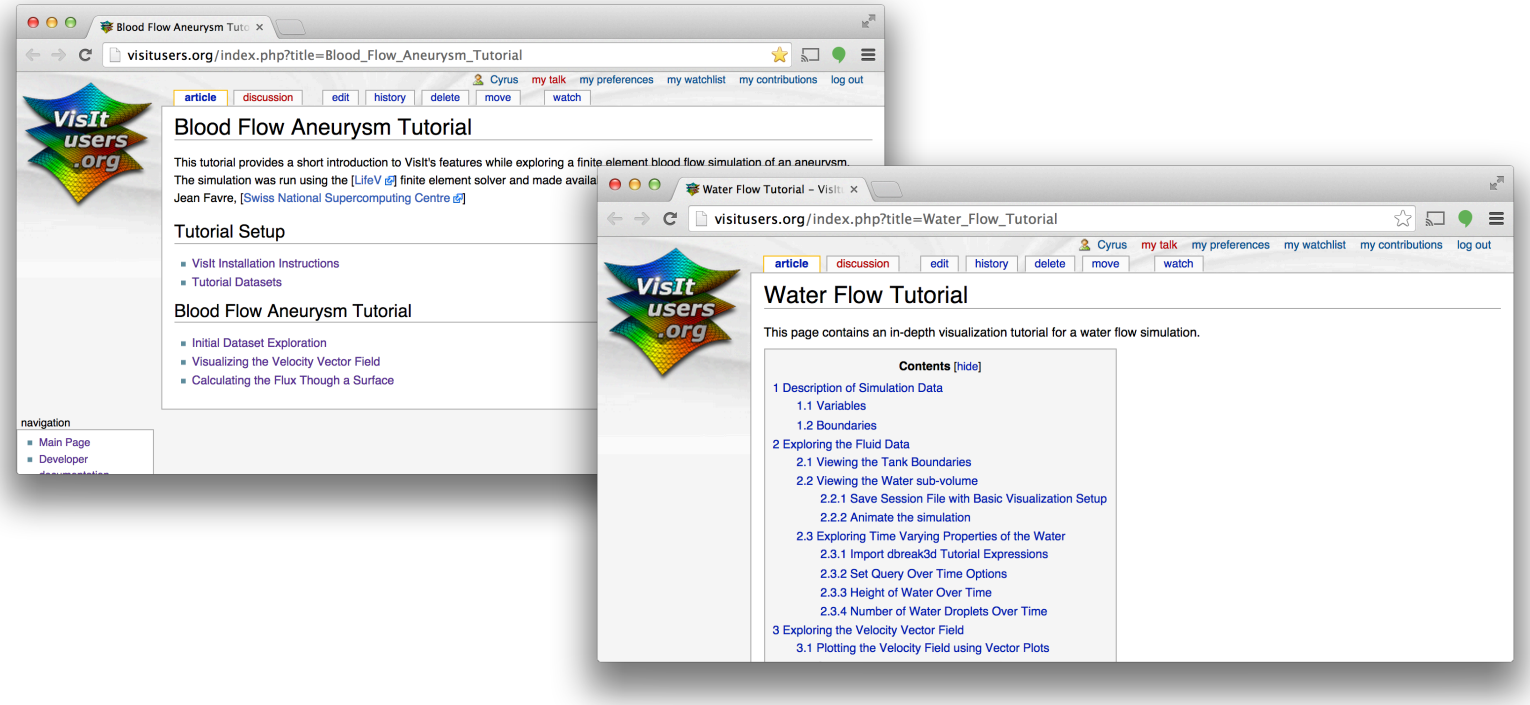
- Launch VisIt's CLI binary:
  - `visit -cli`
- Launch for windowless batch processing:
  - `visit -nowin -cli -s <script_file.py>`
- Control VisIt from a Python interpreter:
  - `import visit'
  - http://visitusers.org/index.php?title=Python_Module_Support
- Record GUI actions in to Python snippets:
  - Macro Recording provides a quick path to learn VisIt's Python Client API.

Python Client Interface Example Script



Demos/Visualizations.

- http://visitusers.org/index.php?title=Blood_Flow_Aneurysm_Tutorial
- http://visitusers.org/index.php?title=Water_Flow_Tutorial

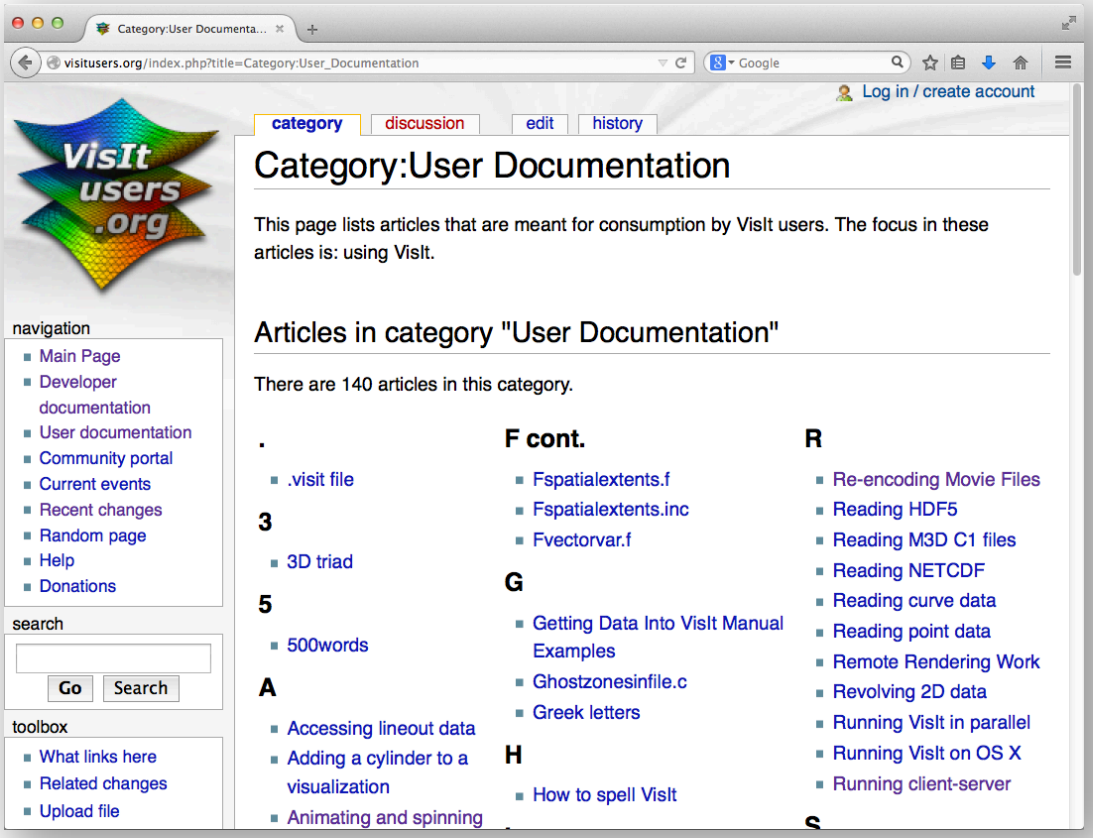


How to get help when you run into trouble

- FAQ
  - —https://wci.llnl.gov/simulation/computer-codes/visit/faq
- VisIt Users Mailing List
  - —Address: visit-users@elist.ornl.gov
  - —Info: https://elist.ornl.gov/mailman/listinfo/visit-users
  - —Archive: https://elist.ornl.gov/pipermail/visit-users/
- VisIt Users Wiki
  - —http://www.visitusers.org
- VisIt Users Forum
  - —http://visitusers.org/forum/YaBB.pl
- Priority support for specific user groups:
  - —VisIt-help-{XYZ} Mailing Lists
- Reference Manuals
  - —https://wci.llnl.gov/simulation/computer-codes/visit/manuals

VisItusers.org

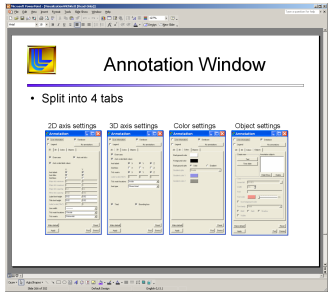
- **Great source for VisIt tips and recipes.**
- **Users section has lots of practical advice:**
  - **—"I solved this problem using this technique"**
  - **—"Here's my script to do this analysis"**

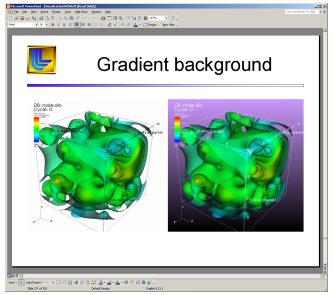


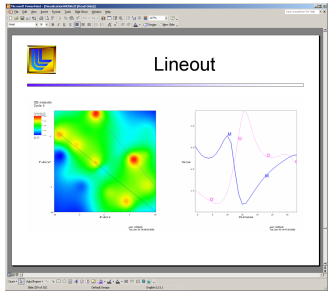
VisItusers.org is the VisIt project's staging area for usage recipes and future formal documentation.

Conclusion: The VisIt team focuses on making a robust, usable product for end users.

- Regular releases (~ 6 / year)
  - Executables for all major platforms
  - End-to-end build process script ``build_visit''
- Customer Support and Training
  - visitusers.org, wiki for users and developers
  - Email lists: visit-users, visit-developers
  - Beginner and advanced tutorials
  - VisIt class with detailed exercises
- Documentation
  - "Getting data into VisIt" manual
  - Python interface manual
  - Users reference manual







Slides from the VisIt class