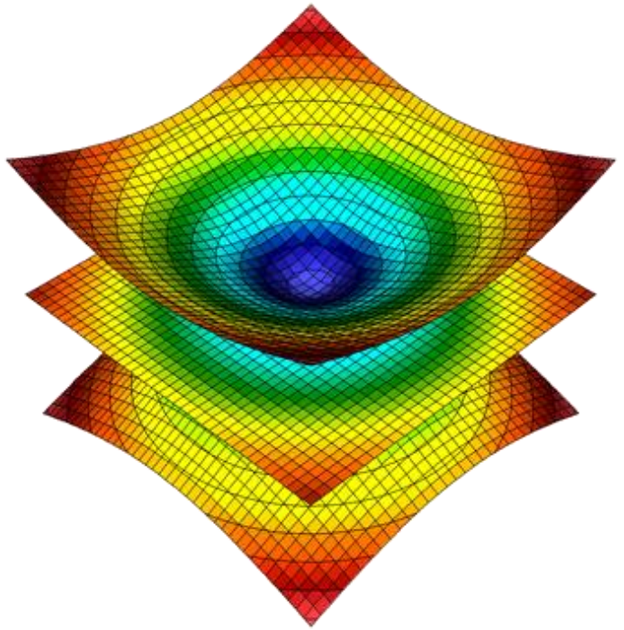
Scientific Visualization 101

VisIt: *An Introductory Hands-On Workshop*



KAUST Visualization Core Lab

James Kress

Workshop Site: *wiki.vis.kaust.edu.sa/training*
Install VisIt 3.3.0 or newer: https://visit-dav.github.io/visit-website/releases-as-tables/#latest

Resources

Presenter/KVL Contact Info:

- James Kress: james.kress@kaust.edu.sa
- KVL website: wiki.vis.kaust.edu.sa
- General Inquiries: help@vis.kaust.edu.sa

User Resources:

- Main website: http://www.llnl.gov/visit
- Discussions: https://github.com/visit-dav/visit/discussions
- User Guide: https://visit-sphinx-github-user-manual.readthedocs.io/en/develop/
- Wiki: http://www.visitusers.org

Developer Resources:

- Github: https://github.com/visit-dav/visit

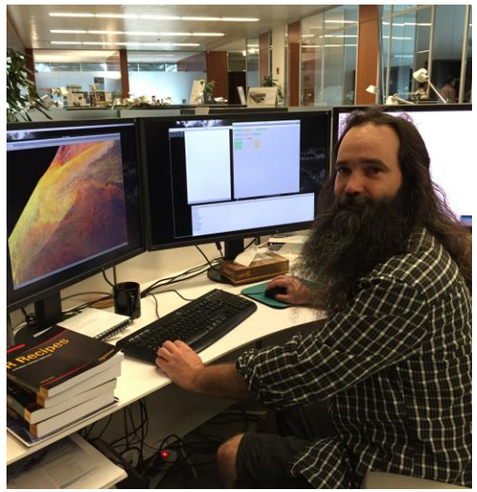
Visualization Core Lab

Overview of Facilities & Services

The Team

Dr. Sohaib Ghani
(LEAD STAFF SCIENTIST)

- VISUAL ANALYTICS
- INFORMATION VIS
- STATISTICAL ANALYSIS

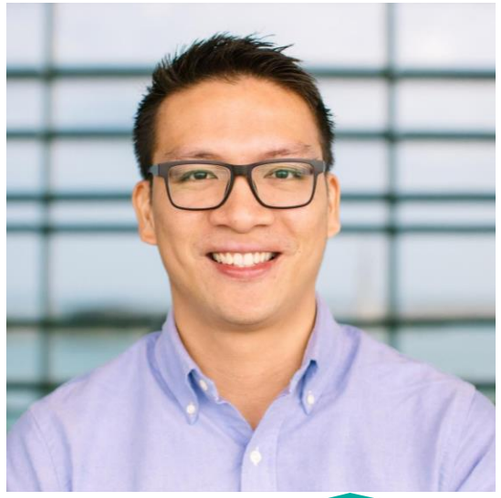


Thomas Theussl
SCIVIS

- SCIENTIFIC VISUALIZATION
- LARGE DATA ANALYSIS
- DISTRIBUTED VISUALIZATION

Dr. James Kress
HPC SCIVIS

- VISUALIZATION SOFTWARE
- HPC INSITU VISUALIZATION
- DISTRIBUTED VISUALIZATION



Dr. Ronell Sicat
VR/AR

- SCIENTIFIC VISUALIZATION
- VR DEVELOPMENT
- 3D RECONSTRUCTION

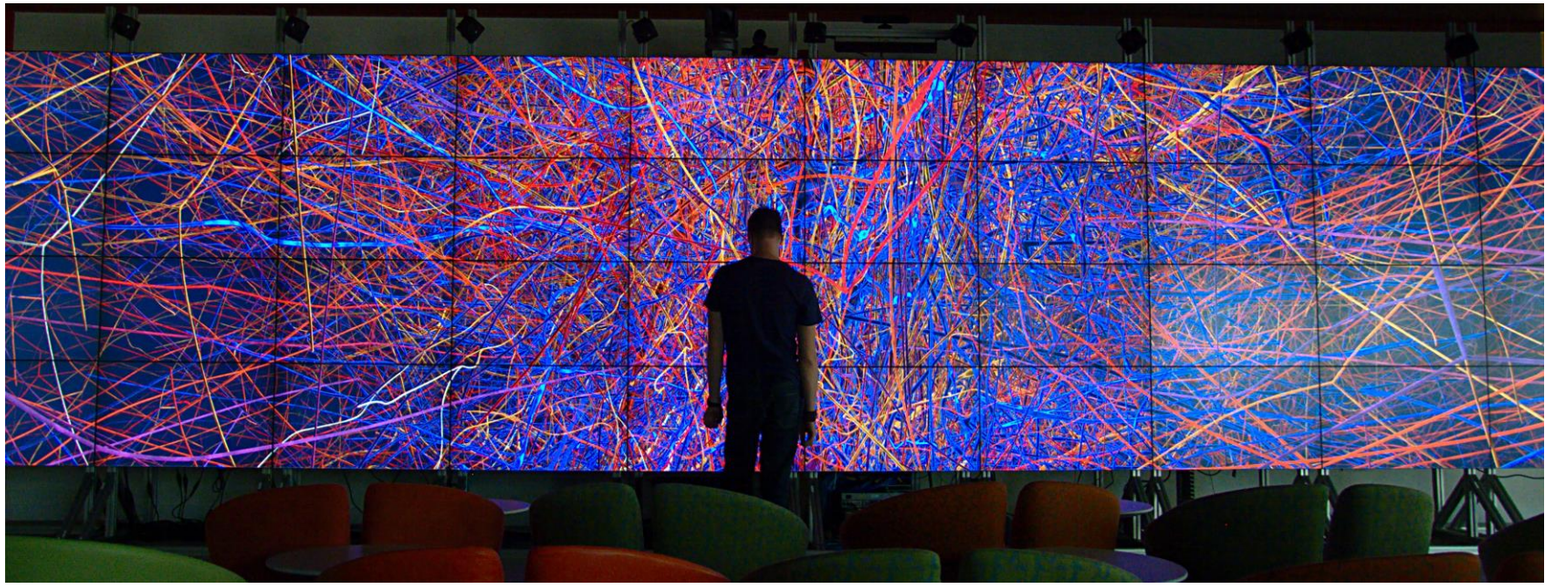
Dr. Didier Barradas
Data Scientist

- DATA SCIENCE
- MACHINE LEARNING
- DEEP LEARNING

Dr. Abdelghafour Halimi
Data Scientist

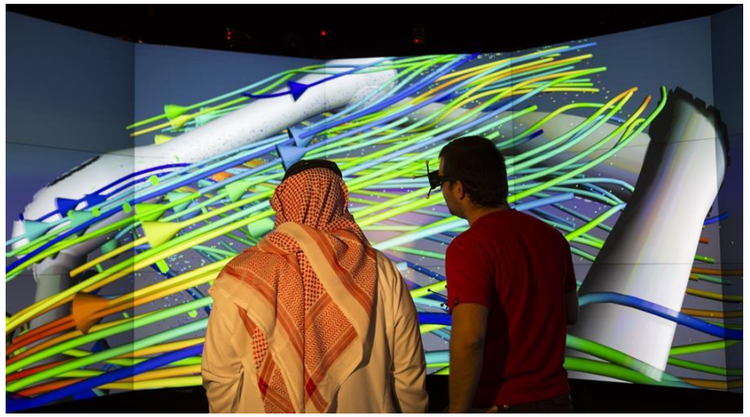
- Data Science
- Machine Learning
- Deep Learning

FACILITIES AND SPACES



ZONE 1/2 DISPLAY WALLS: 2D/3D Analytics

HMD’s



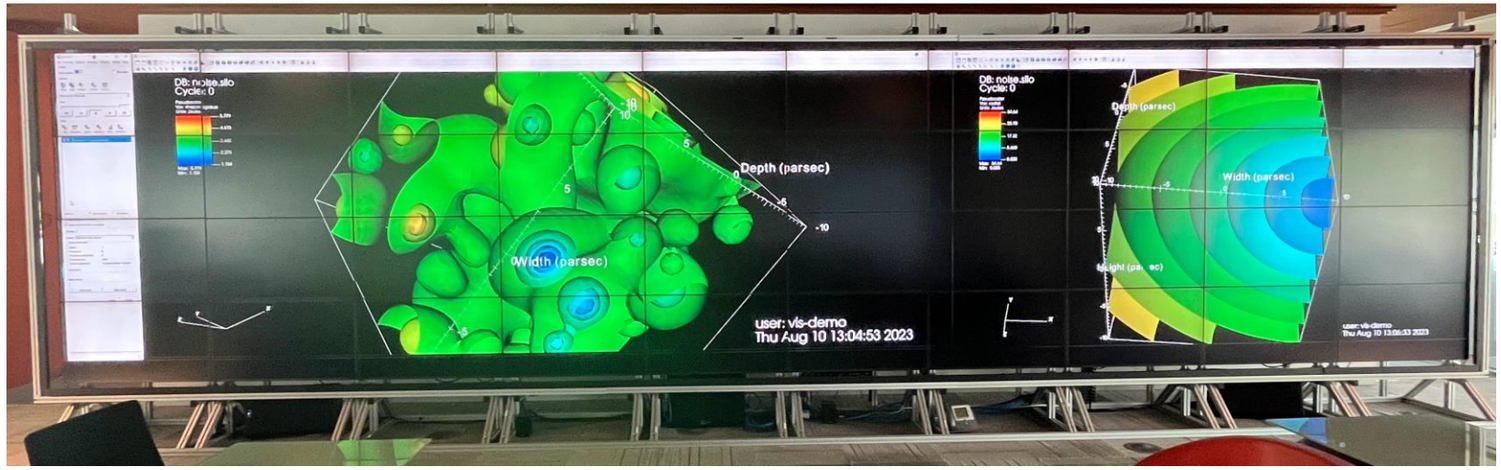
CUBES VR

ZONE 5 VR

MULTI-PURPOSE ROOM

Z2 Visualization and Collaboration

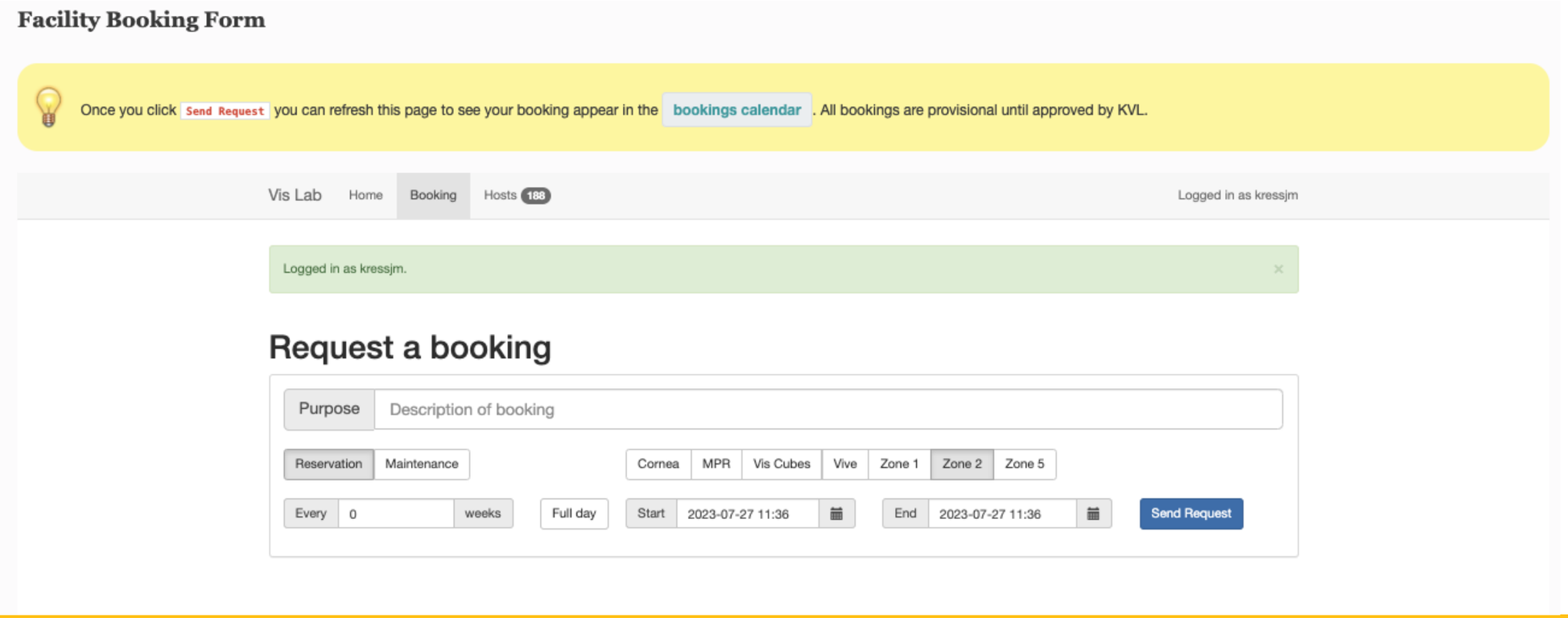
- **ParaView & VisIt on Z2**
  - Connect to Ibex for compute or other networked storage



- **Sage3 collaboration boards**
  - Software to enable teams of collaborators to work together with data in the form of data visualizations
  - https://sage3.sagecommons.org/

Accessing KVL Facilities

- Book here (requires Portal Credentials):
  - https://wiki.vis.kaust.edu.sa/booking



Collaborating with KVL

- Standard Request
  - Load data ‘X’ in program ‘P’ to produce a visualization ‘V’
- Advanced Support
  - Investigative visualization –
    - Asking “why?” & “what?” of your data
- Collaboration
  - Work with you through your research and discovery cycle

Upcoming Training Events

Scientific Visualization Workshop Series Fall 2023

Date	Training Event	Speaker	Registration
Sunday October 1, 2023	Scientific Visualization 101: VisIt ~ An Introductory Hands-On Workshop	James Kress	Register Now
Sunday October 8 2023	Scientific Visualization 210: ParaView & VisIt ~ Scripting and Supercomputing Workflows	James Kress & Thomas Theußl	Register Now
Sunday October 22, 2023	Scientific Visualization 101: Avizo/Amira ~ An Introductory Hands-On Workshop	Thomas Theußl	Register Now
Sunday October 29, 2023	Scientific Visualization 210: ParaView ~ A Plugin for Geometry Processing	Thomas Theußl	Register Now
Sunday November 5, 2023	Scientific Visualization 210: Avizo and Ilastik for Image Segmentation and 3D Analysis	Ronell Sicat	Register Now

Hands-on AI Tools and Techniques Workshop Series Fall 2023

Date	Training Event	Speaker	Registration
Tuesday October 3, 2023	Introduction to Machine Learning	Abdelghafour Halimi	Register Now
Tuesday October 10, 2023	Introduction to Deep Learning	Abdelghafour Halimi	Register Now
Tuesday October 17, 2023	Data Visualization for Data Science	Abdelghafour Halimi	Register Now

Edit

Data Science on Kaust HPC platforms Fall 2023

Date	Training Event	Speaker	Registration
Sunday November 20, 2023	Data Science on-boarding on KSL platforms	DB & MS	Register Now
Tuesday November 21, 2023	Distributed Deep Learning on KSL platforms	DB & MS	Register Now
Sunday November 22, 2023	High Throughput Hyperparameter Optimization of ML/DL models on KSL platforms	DB & MS	Register Now
Sunday November 23, 2023	Introduction to Containers on KSL platforms	DB & MS & AH	Register Now

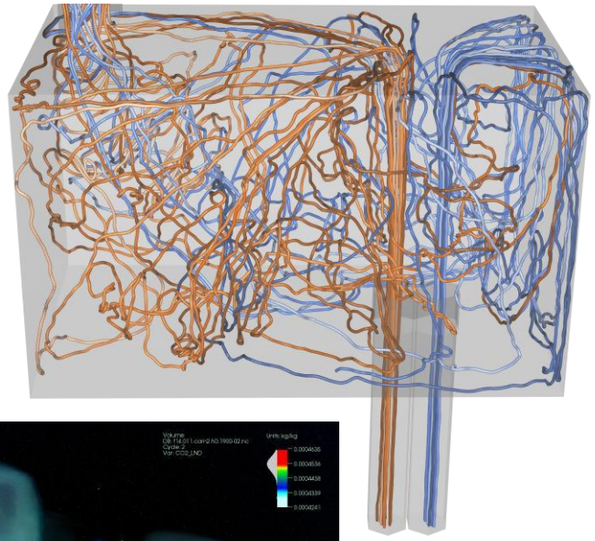
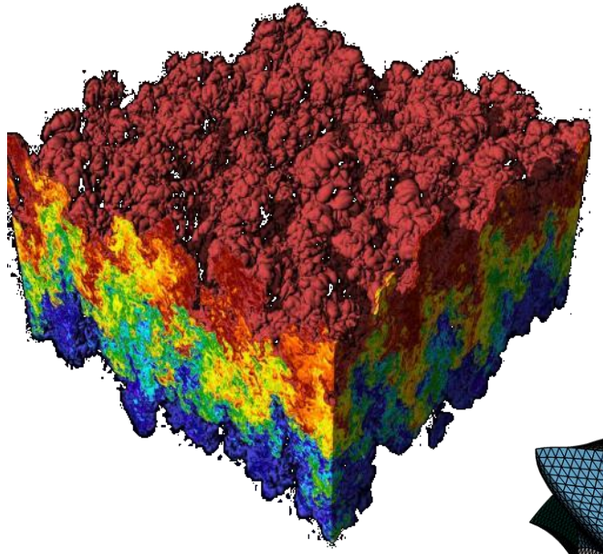
Workshop Goals

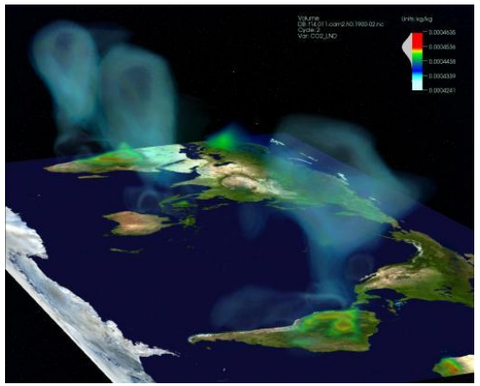
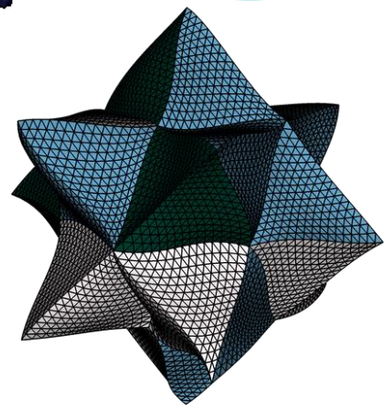
- Hands-on learning with VisIt
  - Introductory course
  - Slides / demonstrations
- Why VisIt @ KAUST
  - Open source, scalable, multi-platform visualization application with users worldwide
  - Available on all major computation resources at KAUST
    - VisIt on Ibex and Shaheen
      - Support for distributed computations to process very large data sets
        - VisIt has been proven on up to 27 billion element meshes
    - VisIt on IT Remote Workstations
    - VisIt @ KVL
      - Tiled-display walls

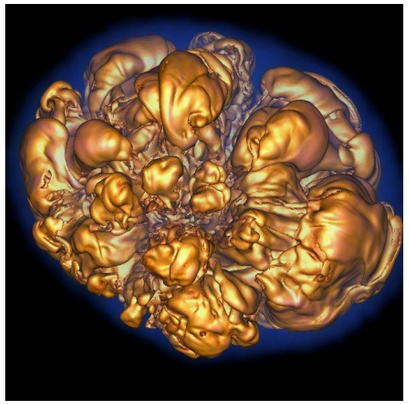
VisIt Basics

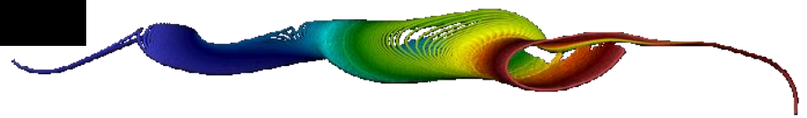
What is VisIt?

- Open source turnkey application for data analysis and visualization of mesh-based data
- Infrastructure for parallel post-processing that scales from laptops to HPC clusters
- Built-in in situ capabilities

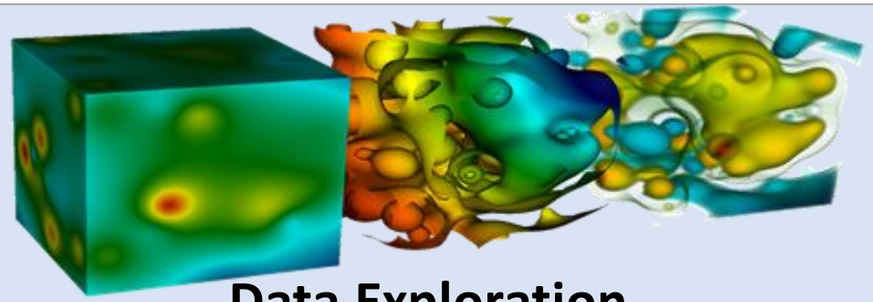



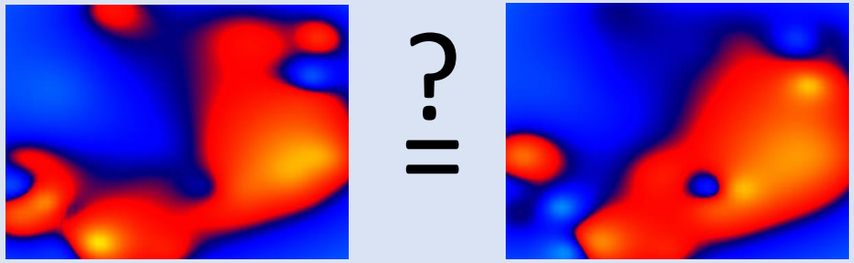




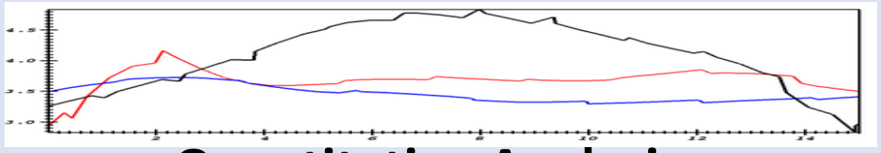
VisIt Supports a Wide Range of Use Cases



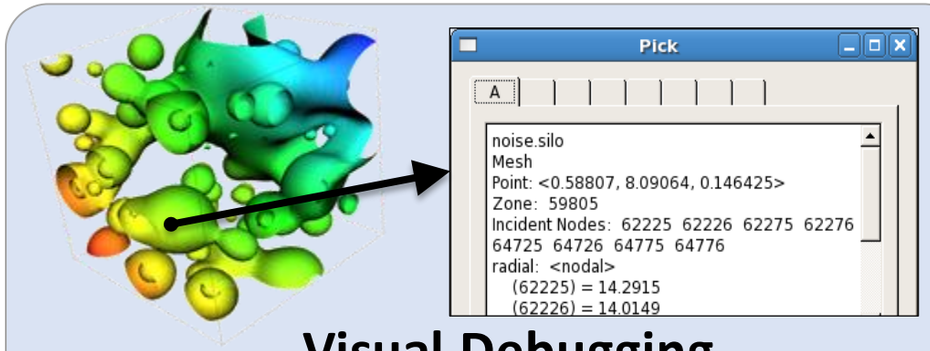
Data Exploration



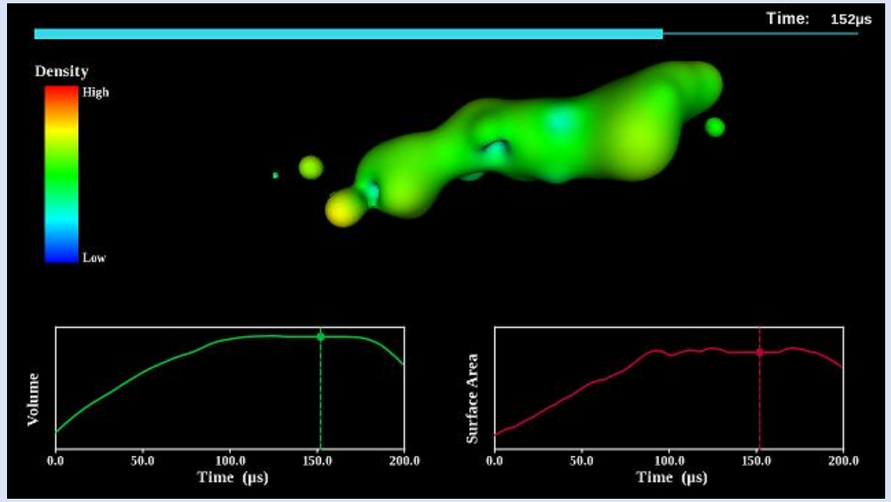
Comparative Analysis



Quantitative Analysis

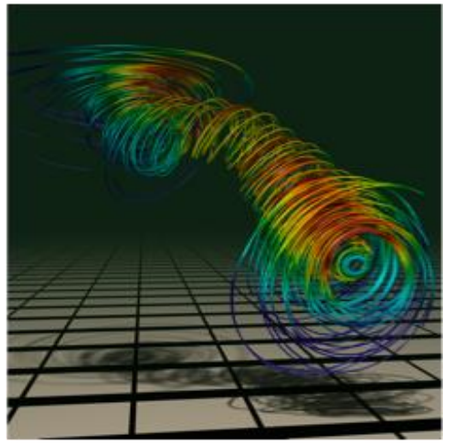


Visual Debugging

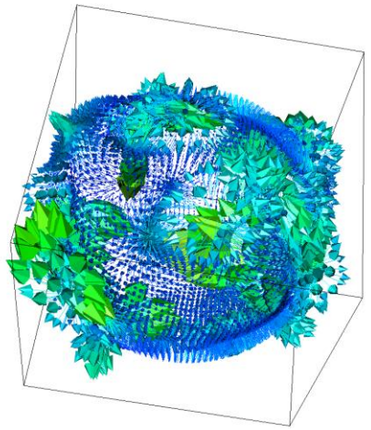


Presentation Graphics

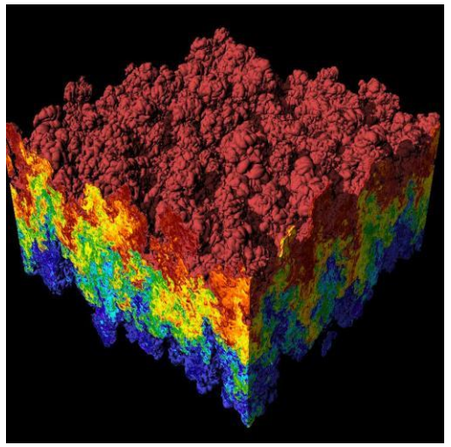
VisIt Supports a Wide Range of Plotting Types



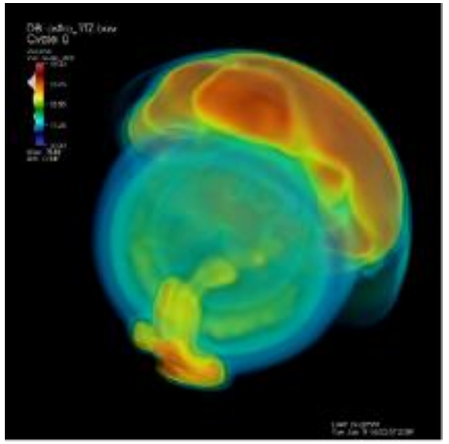
Streamlines / Pathlines



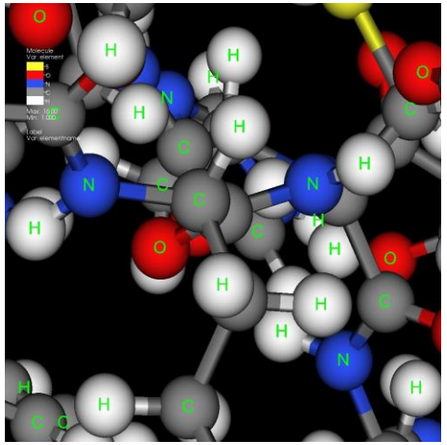
Vector / Tensor Glyphs



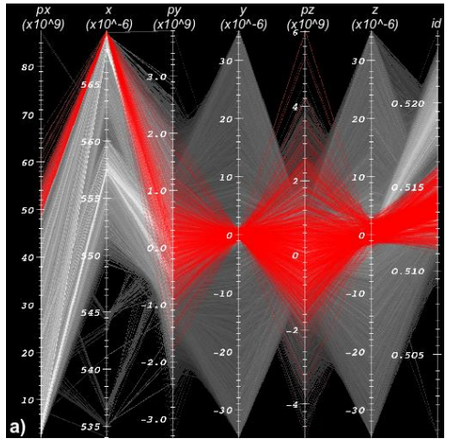
Pseudocolor Rendering



Volume Rendering



Molecular Visualization



Parallel Coordinates

How Do I Obtain VisIt?

- Use an existing build:
  - For your Laptop or Workstation:
    - Binaries for Windows, OSX, and Linux (RHEL + Ubuntu): (https://visit-dav.github.io/visit-website/releases-as-tables/#latest)
  - KVL team manages builds on Ibex and Shaheen
  - IT Remote Workstations
- Build VisIt yourself:
  - "build_visit" is a script that automates the process of building VisIt and its third-party dependencies. (docs: https://visit-sphinx-github-user-manual.readthedocs.io/en/develop/building_visit/index.html)

How Do I Get My Data Into VisIt?

VisIt supports more than 110 file formats

- *VTK, Silo, Xdmf, PVTK*
- The *PlainText* database reader can read simple text files (CSV, etc)
  - https://visit-sphinx-github-user-manual.readthedocs.io/en/develop/data_into_visit/PlainTextFormat.html
- *visit_writer* utility: code to write VTK files from your sim code
  - https://visit-sphinx-github-user-manual.readthedocs.io/en/develop/data_into_visit/VTKFormat.html
- Support for Mesh-based data in Conduit Blueprint:
  - http://llnl-conduit.readthedocs.io/en/latest/blueprint_mesh.html

Read the docs: https://visit-sphinx-github-user-manual.readthedocs.io/en/develop/data_into_visit/index.html

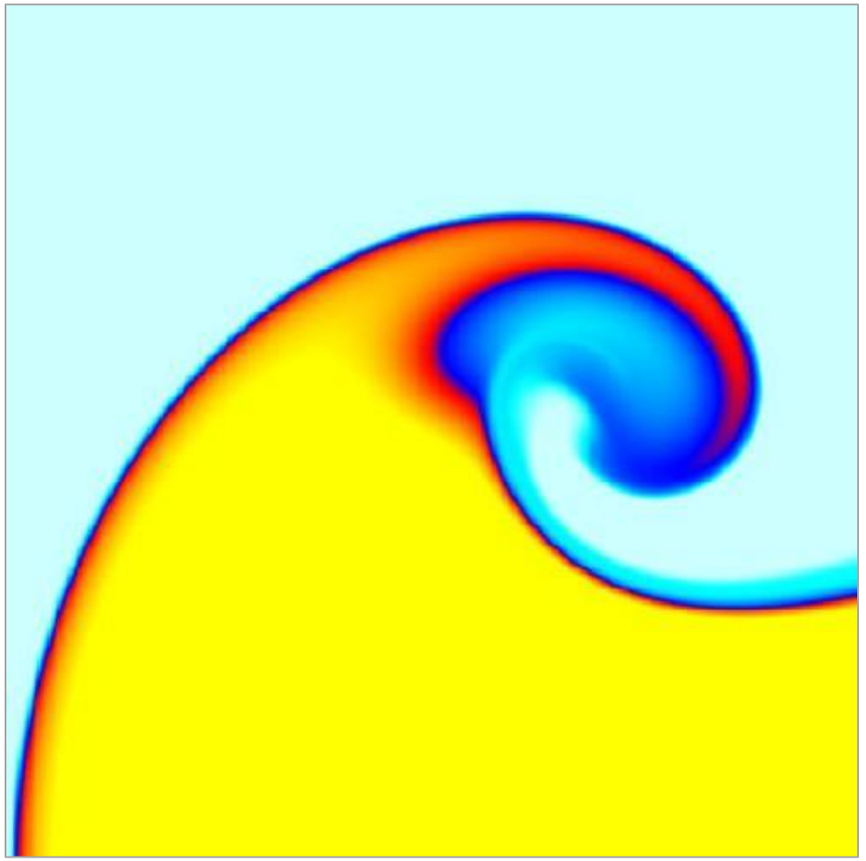
Visualization Techniques

For Mesh Based Simulations

Pseudocolor Rendering

maps scalar fields to a range of colors

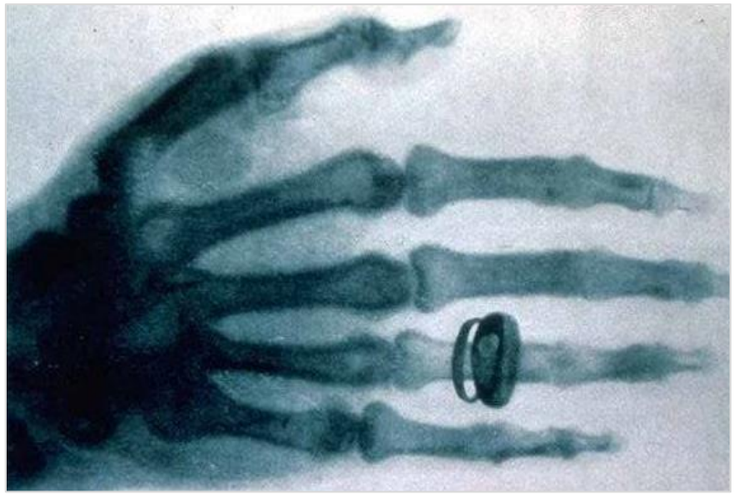
Pseudocolor rendering of Elevation

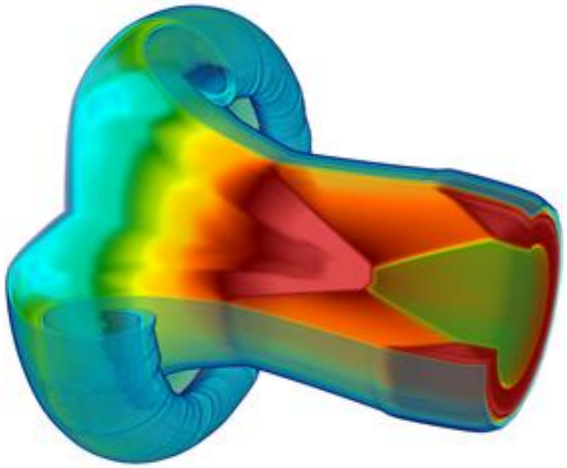


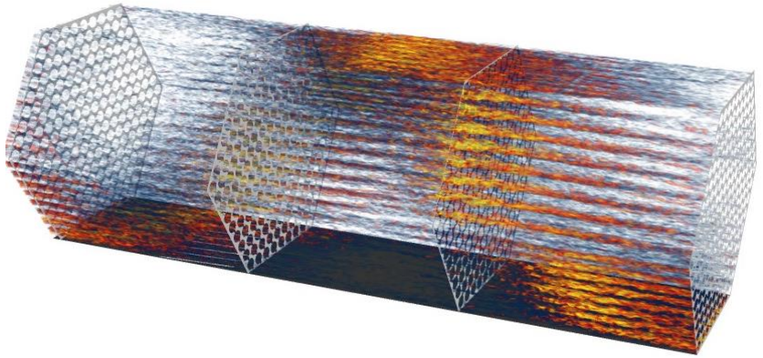
Pseudocolor rendering of Density

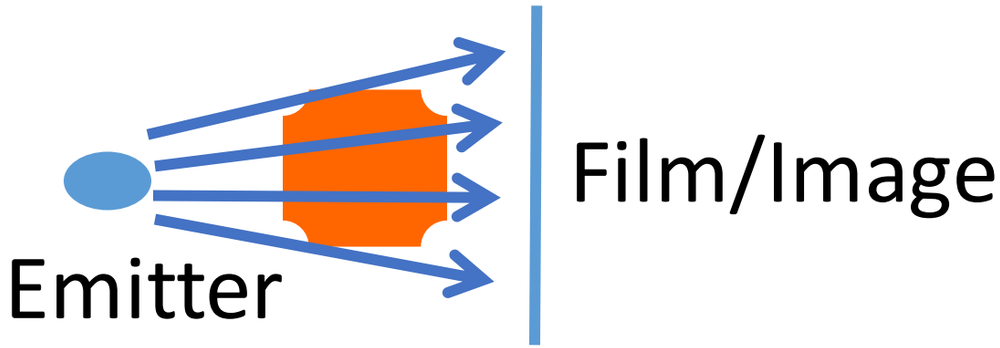
Volume Rendering

cast rays though data and applies transfer functions to produce an image



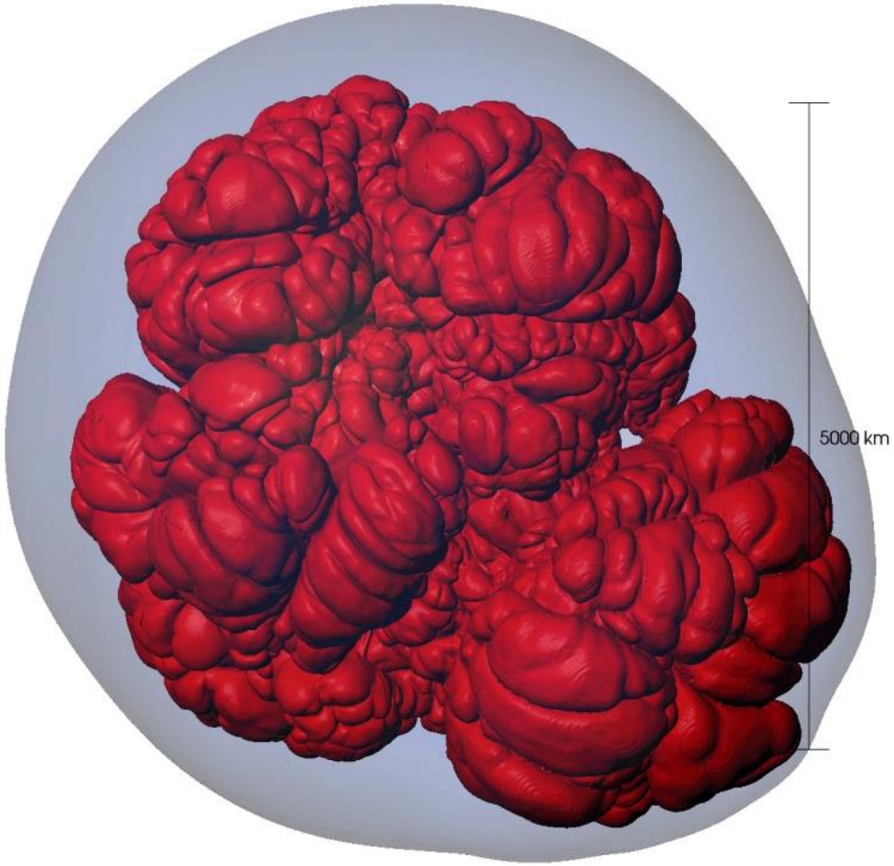






Isosurfacing (Contouring)

extracts surfaces of that represent level sets of field values

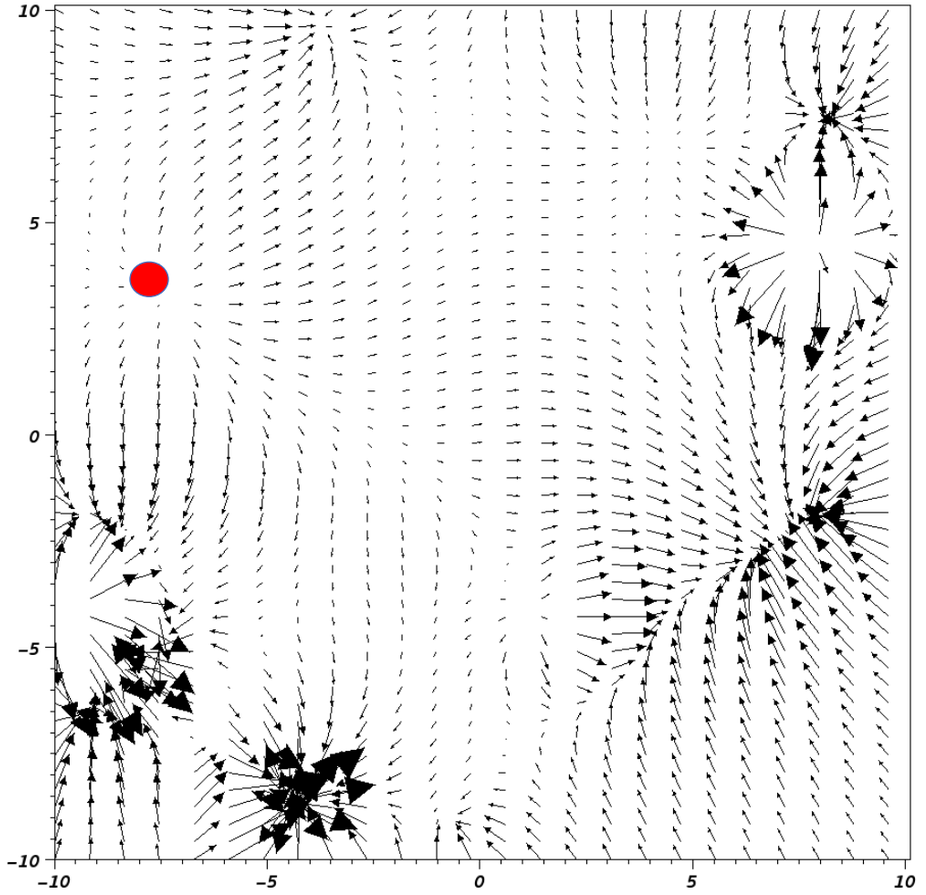


Particle advection

the foundation of several flow visualization techniques

- S(t) = position of particle 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

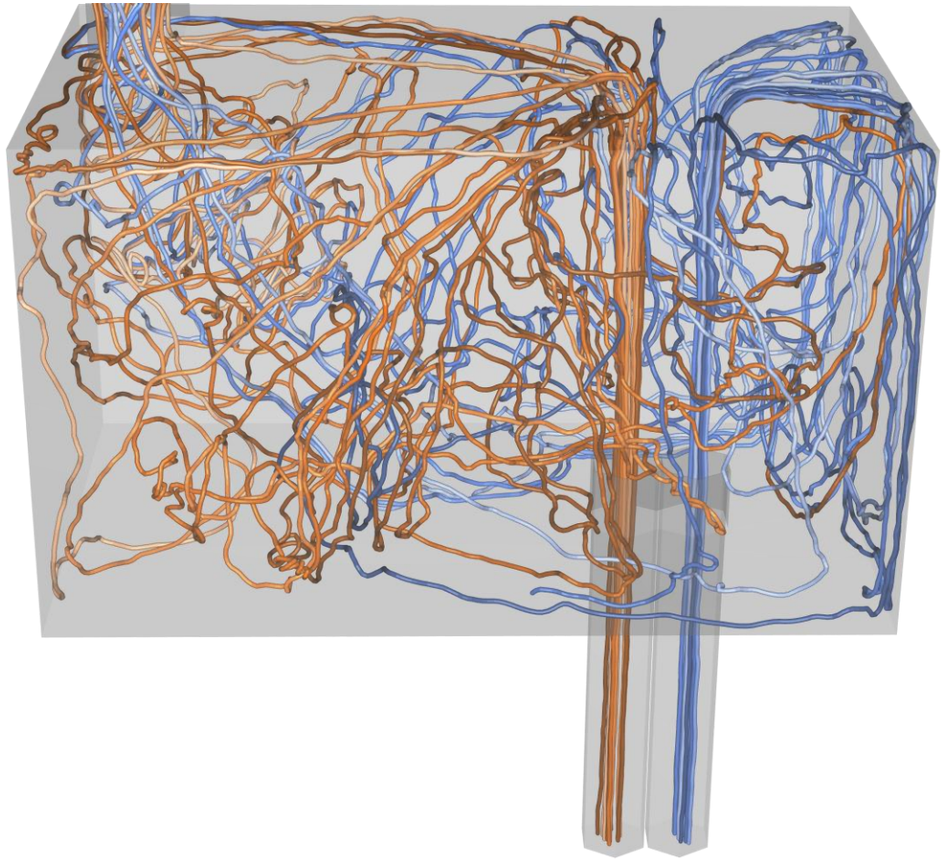
This is an ordinary differential equation.



Streamline and Pathline

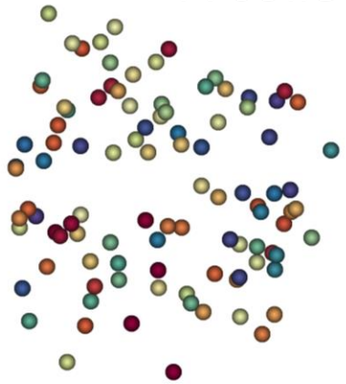
built on particle advection

- **Streamlines** – Instantaneous paths
- **Pathlines** – Time dependent paths

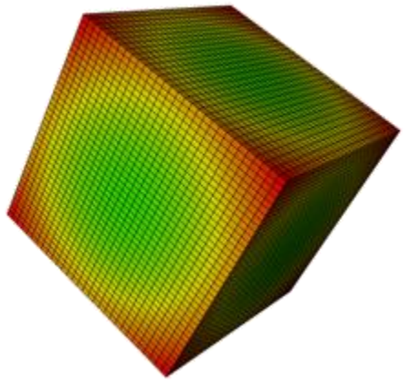


Meshes discretize continuous space

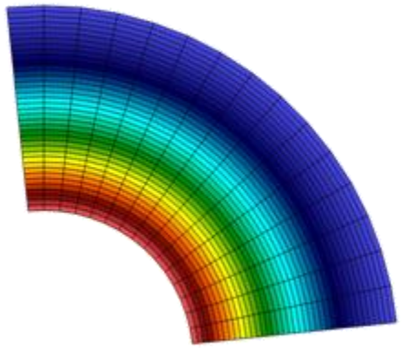
- **Simulations use a wide range of mesh types, defined in terms of:**
  - A set of coordinates (“nodes” / “points” / “vertices”)
  - A collection of “zones” / “cells” / “elements” on the coordinate set



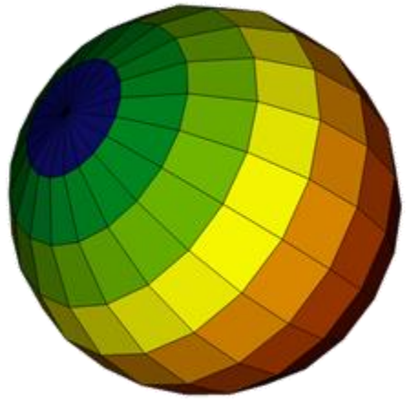
Points



Uniform



Curvilinear



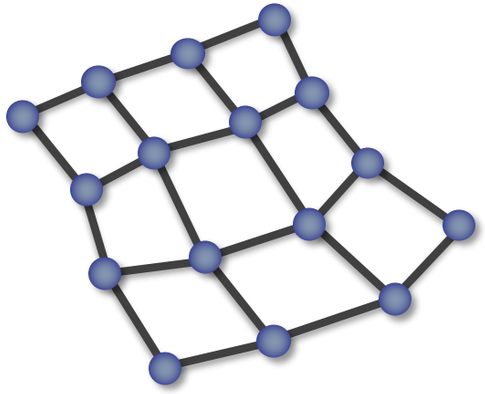
Unstructured

VisIt uses the “Zone” and “Node” nomenclature throughout its interface.

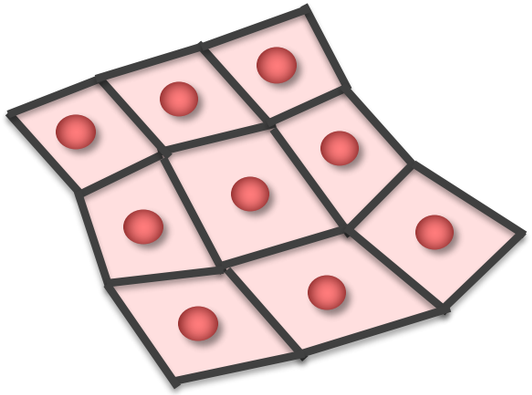
Mesh fields

variables associated with the mesh that hold simulation state

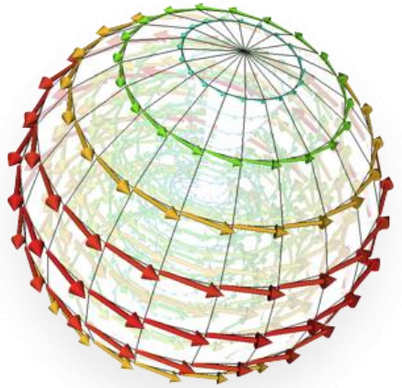
- Field values are associated with the zones or nodes of a mesh
  - Nodal: Linearly interpolated between the nodes of a zone
  - Zonal: Piecewise Constant across a zone
- Field values for each zone or node can be scalar, or multi-valued (vectors, tensors, etc.)



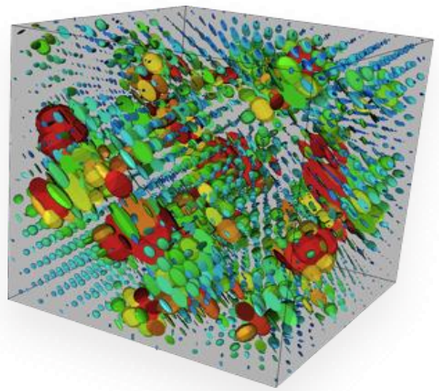
Nodal Association



Zonal Association



Vector Field

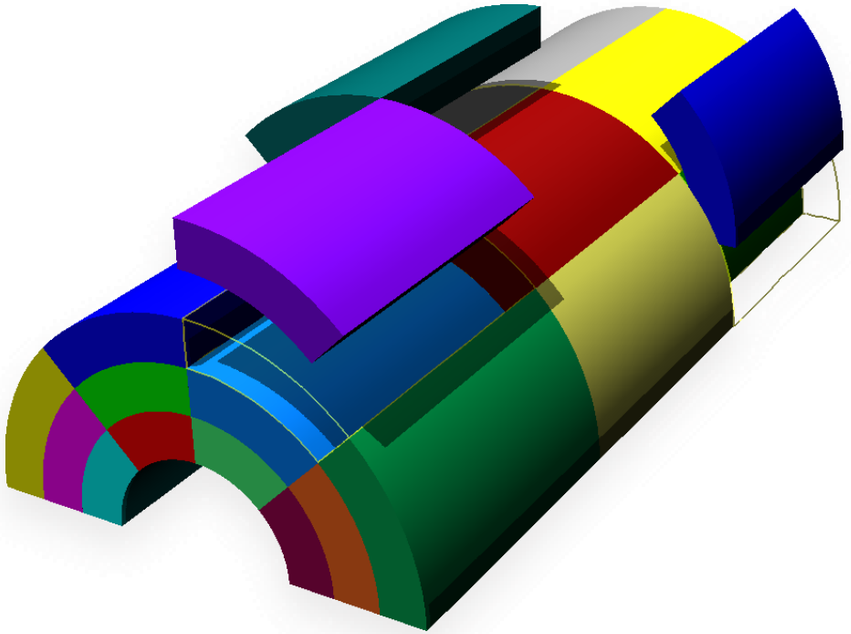


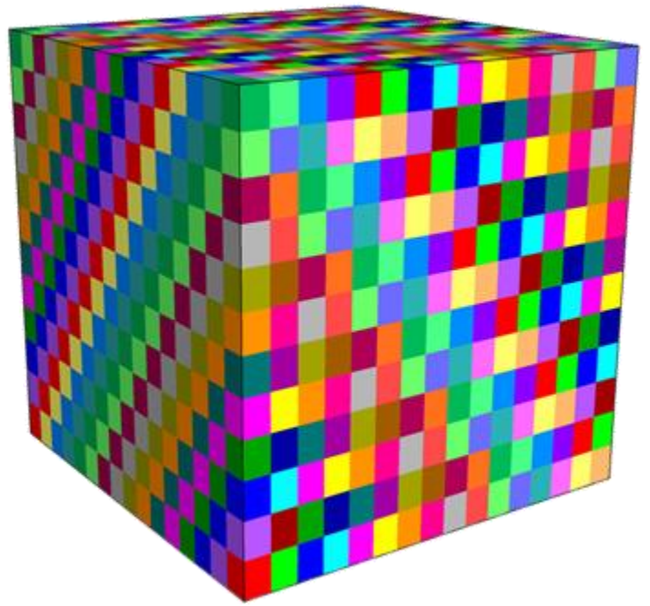
Tensor Field

Domain decomposed meshes

enable scalable parallel visualization and analysis algorithms

- Simulation meshes may be composed of smaller mesh “blocks” or “domains”
- Domains are partitioned across MPI tasks for processing

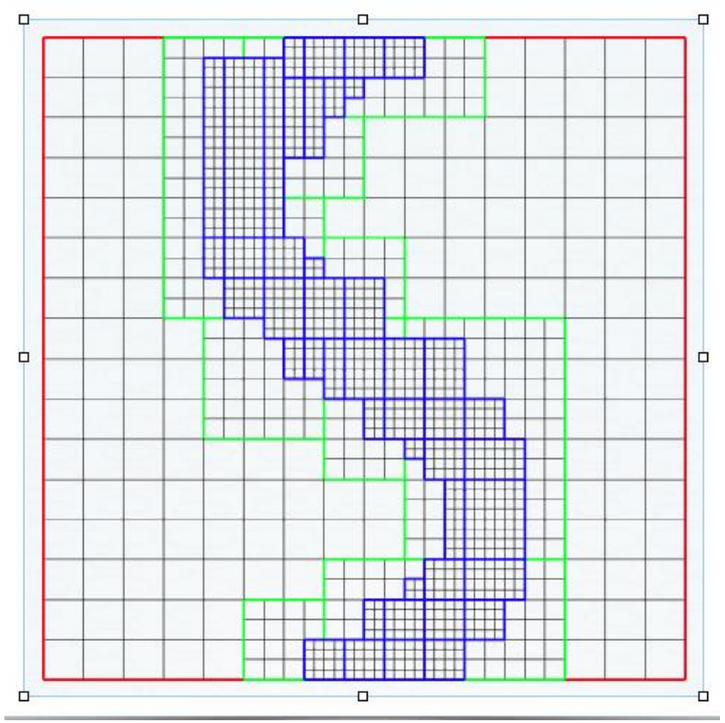


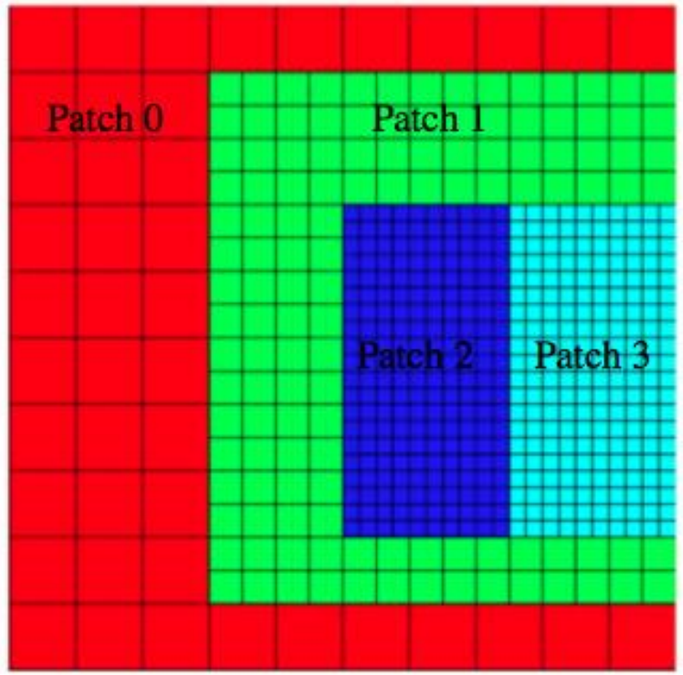


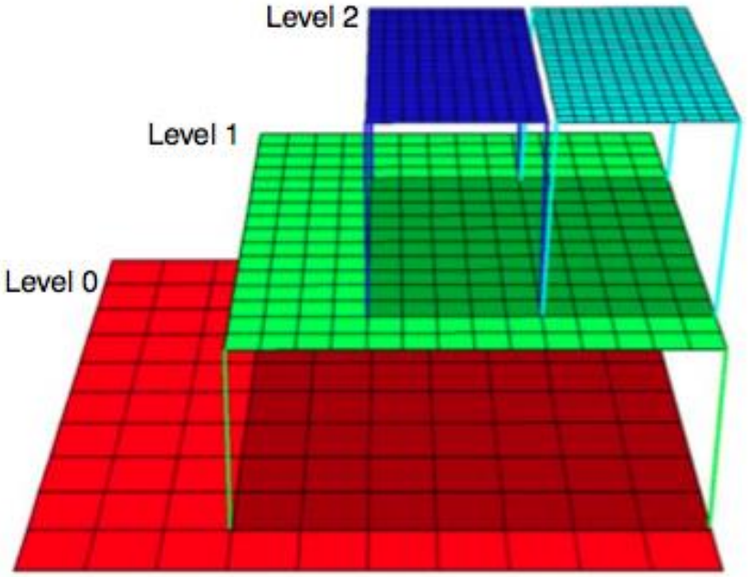
Adaptive Mesh Refinement (AMR)

refines meshes into patches that capture details across length scales

- Mesh domains are associated with patches and levels
- Patches are nested to form a AMR hierarchy

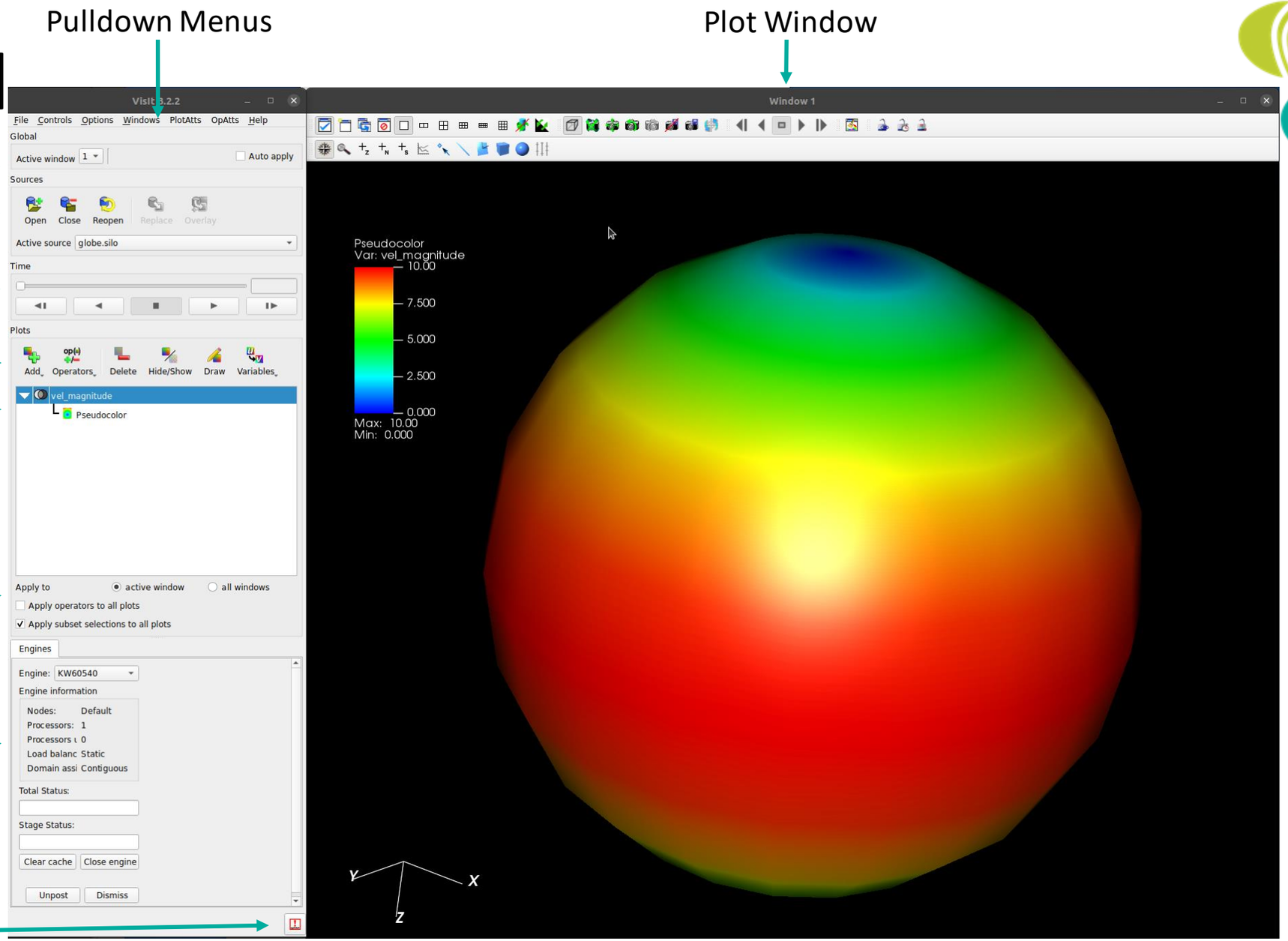




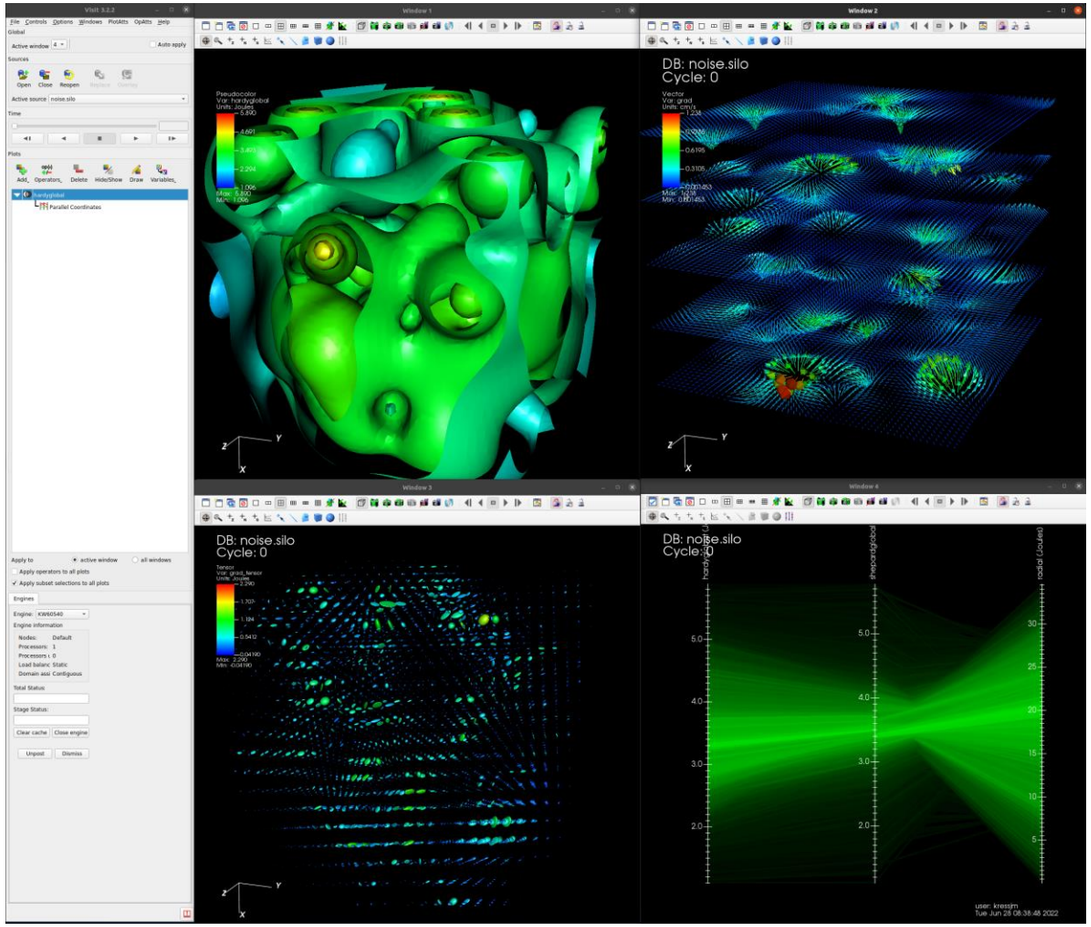


VisIt Interface Tour

VisIt GUI


Pulldown Menus
Plot Window
Active Window Selector
Data Sources
Time Slider
Plots & Operators
Pipeline Browser
Apply To
Notepad
Output Indicator
VisIt 3.2.2
File Controls Options Windows PlotAtts OpAtts Help
Global
Active window 1
Auto apply
Sources
Open Close Reopen Replace Overlay
Active source globe.silo
Time
Plots
Add Operators Delete Hide/Show Draw Variables
vel_magnitude
Pseudocolor
Apply to active window all windows
Apply operators to all plots
Apply subset selections to all plots
Engines
Engine: KW60540
Engine information
Nodes: Default
Processors: 1
Load balanc Static
Domain assi Contiguous
Total Status:
Stage Status:
Clear cache Close engine
Unpost Dismiss
Window 1
Pseudocolor
Var: vel_magnitude
10.00
7.500
5.000
2.500
0.000
Max: 10.00
Min: 0.000
Y
X
Z

VisIt GUI Tour



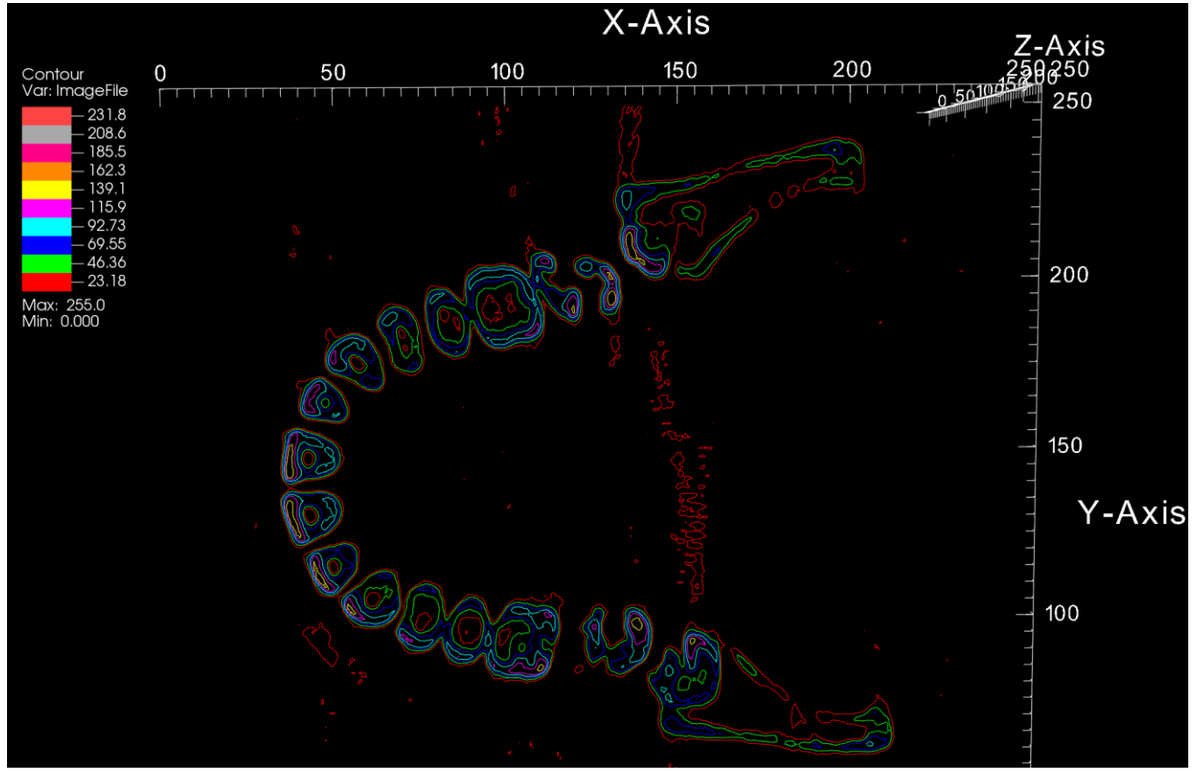
- Opening files / file types
- View file info
- Navigating views
- Multiple views
- Window tools
- Add plot / add operator
- Change plot / operator attributes
- Selectively applying operators
- Link views

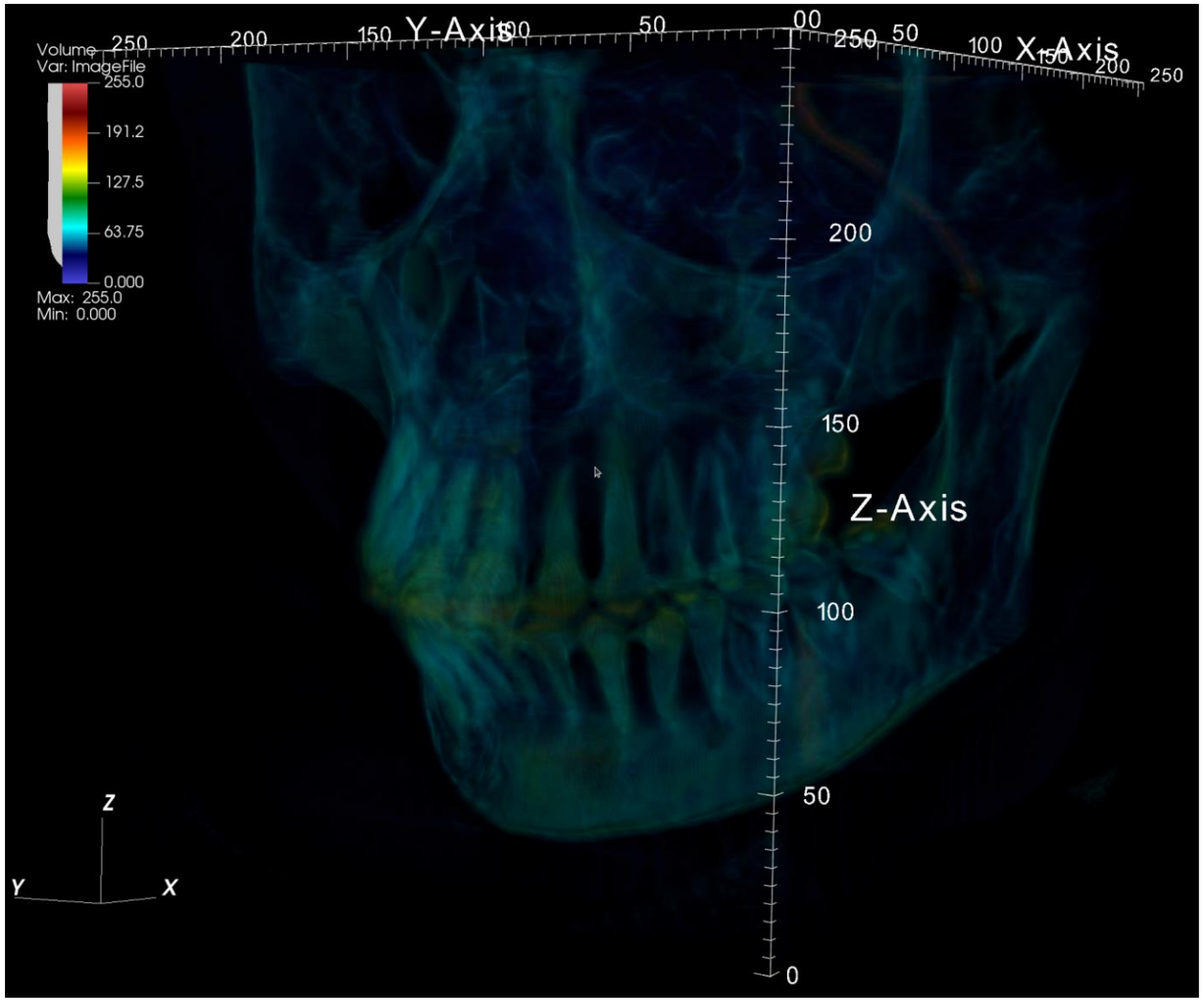
Hands–On Session 1

Basic Plots / Slices / Volume Rendering

Data Set

- Skull data set
  - skull.vti
    - VTK image data format



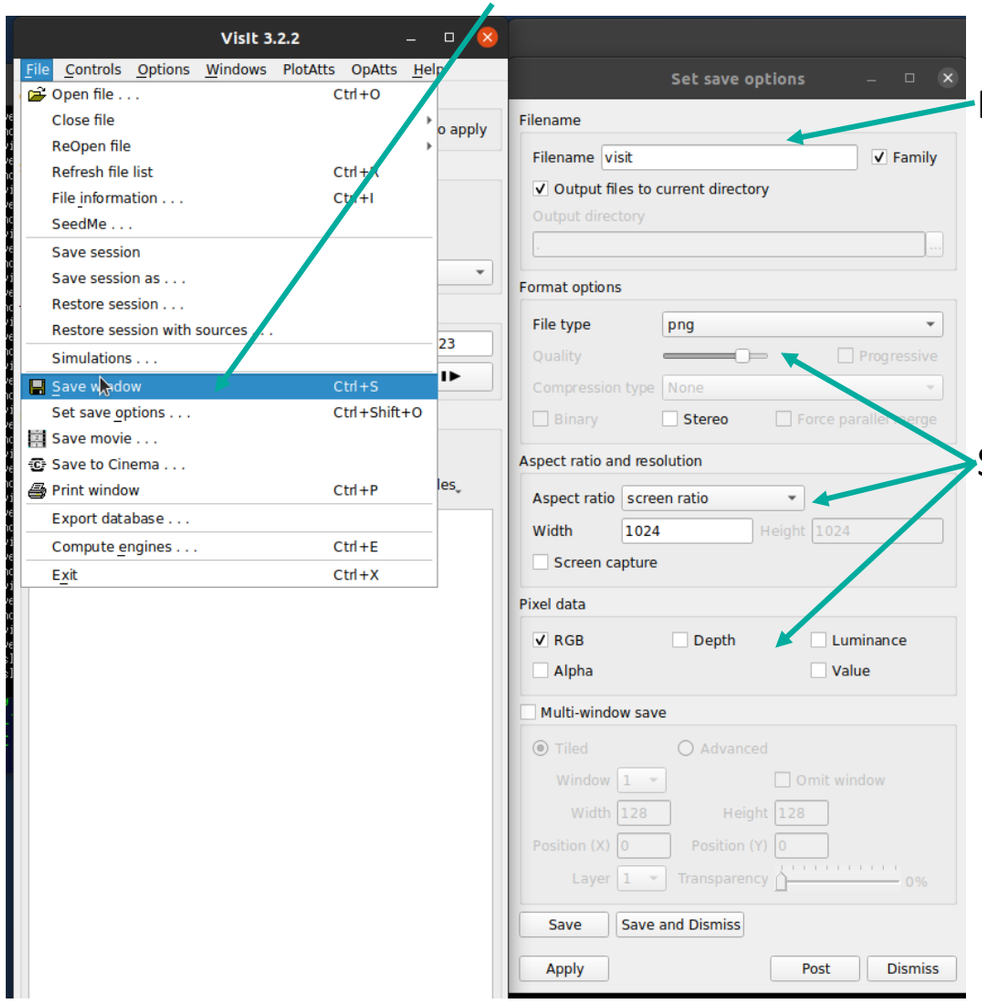


Hands–On Session 2

Screenshots / Movies / Animations / Custom Expressions / Queries / Saving VisIt State

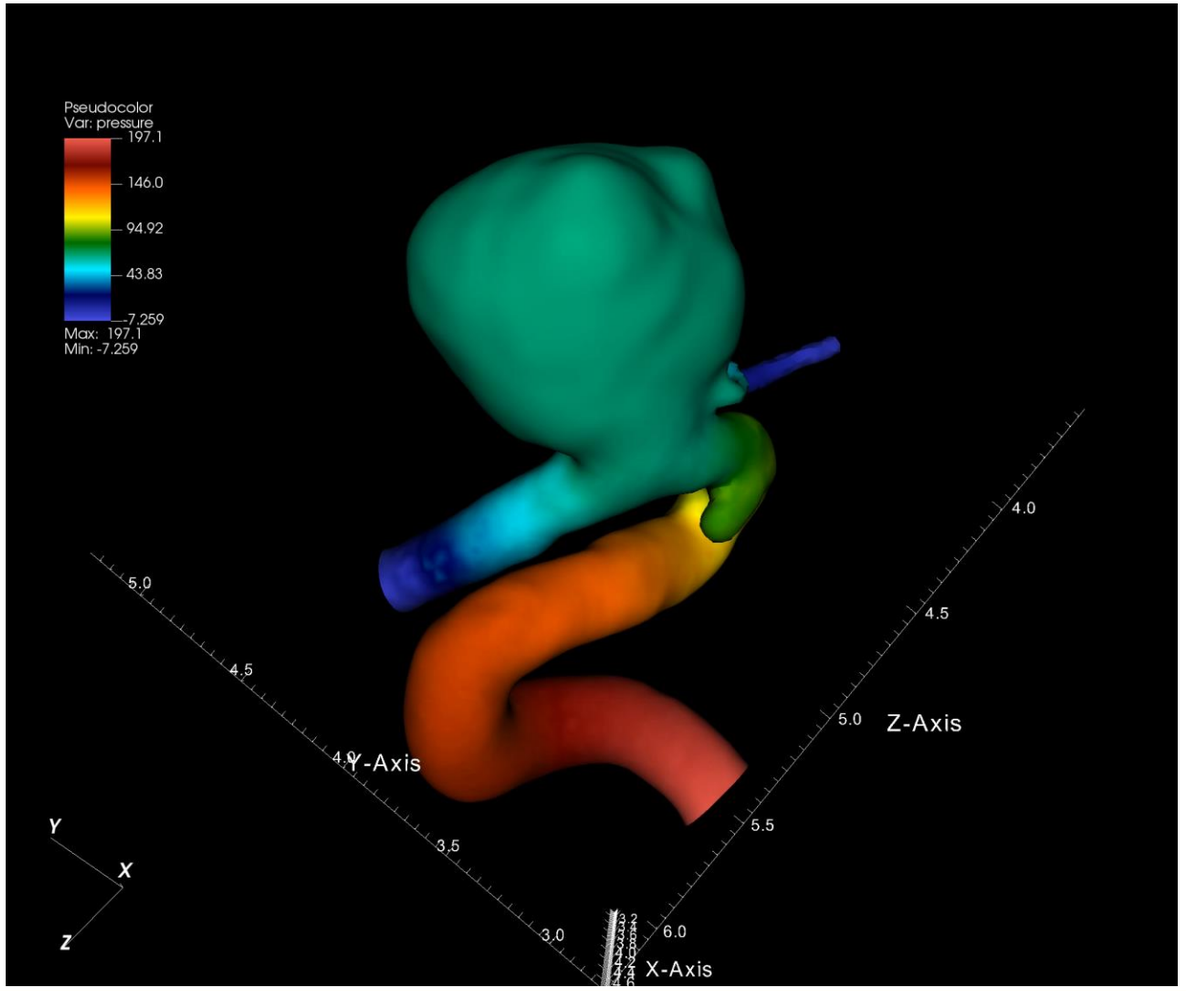
Screenshots / Movies

Save Window / Save Options



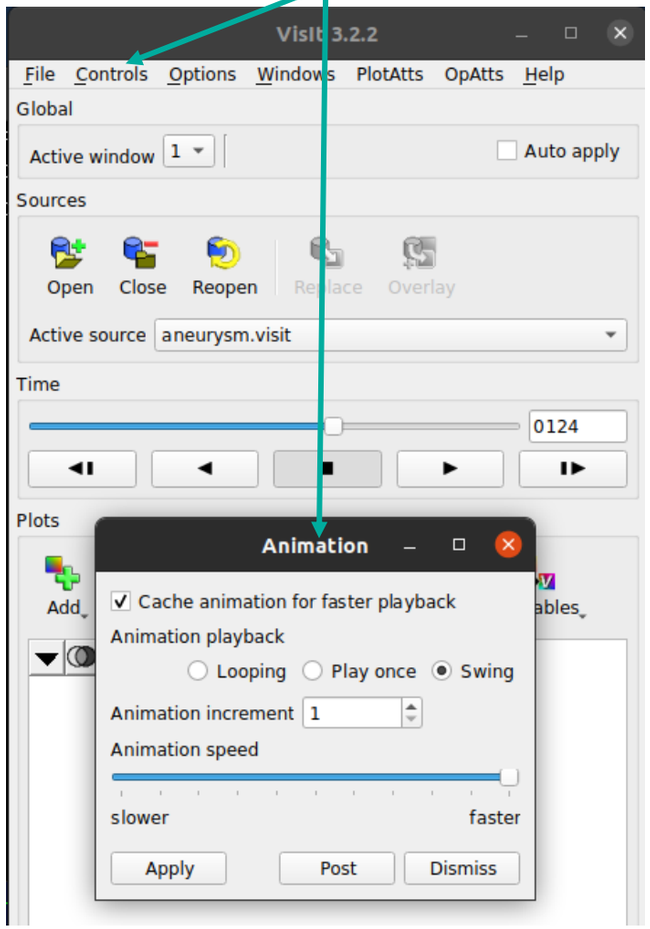
Filename & Directory

Save Options

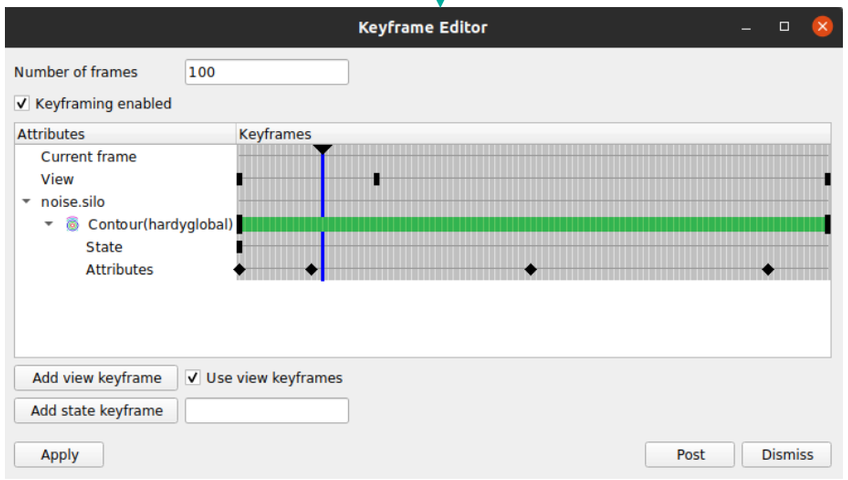


Animation / Keyframing

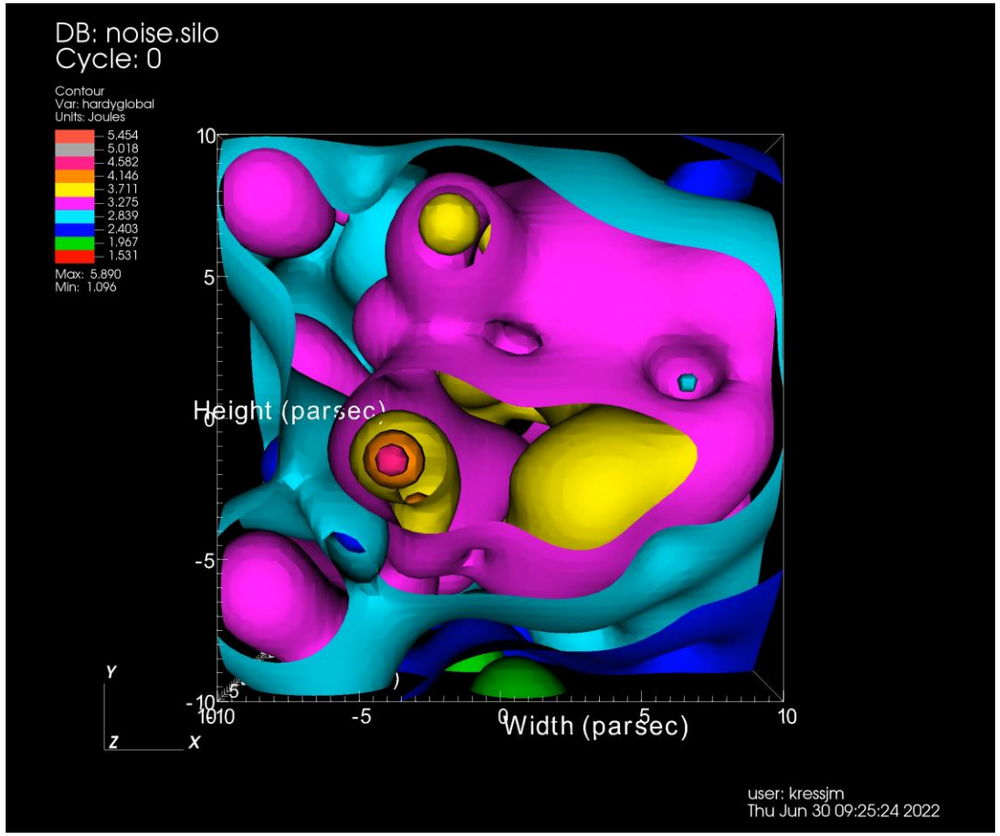
Animation Time Slider



Keyframe editor



Resulting Movie

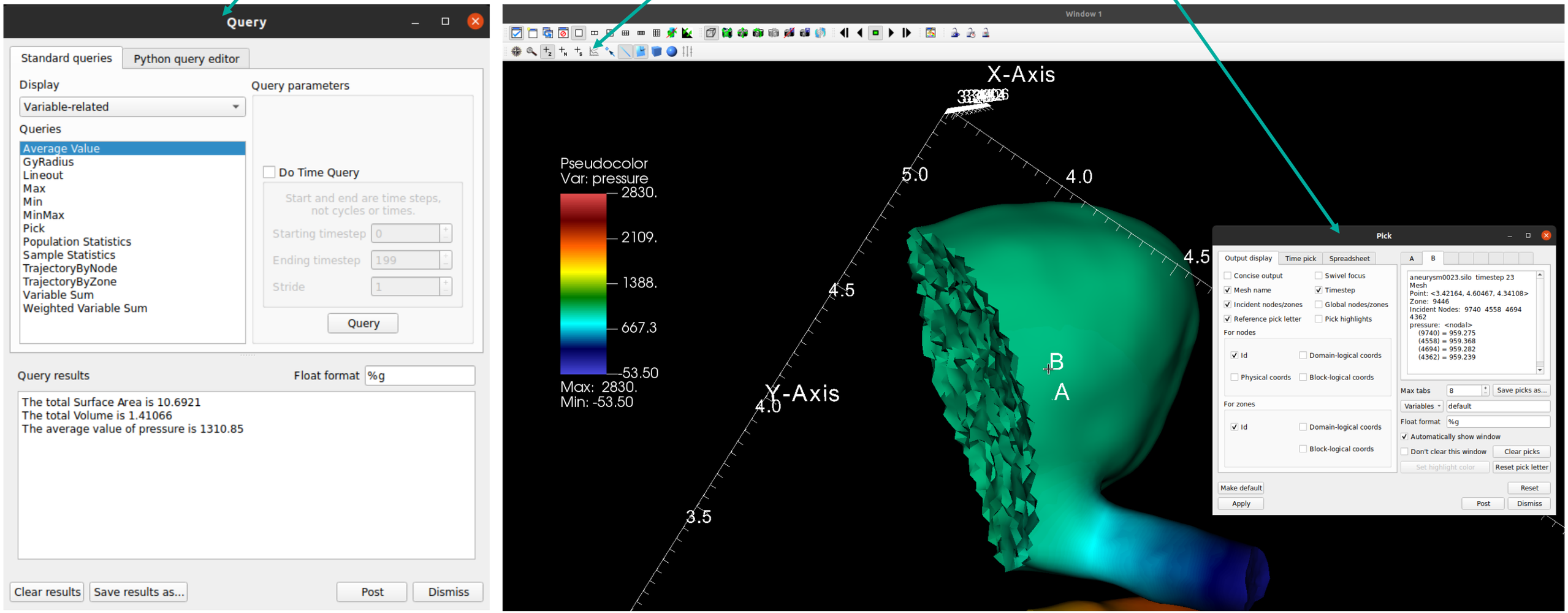


Queries / Pick

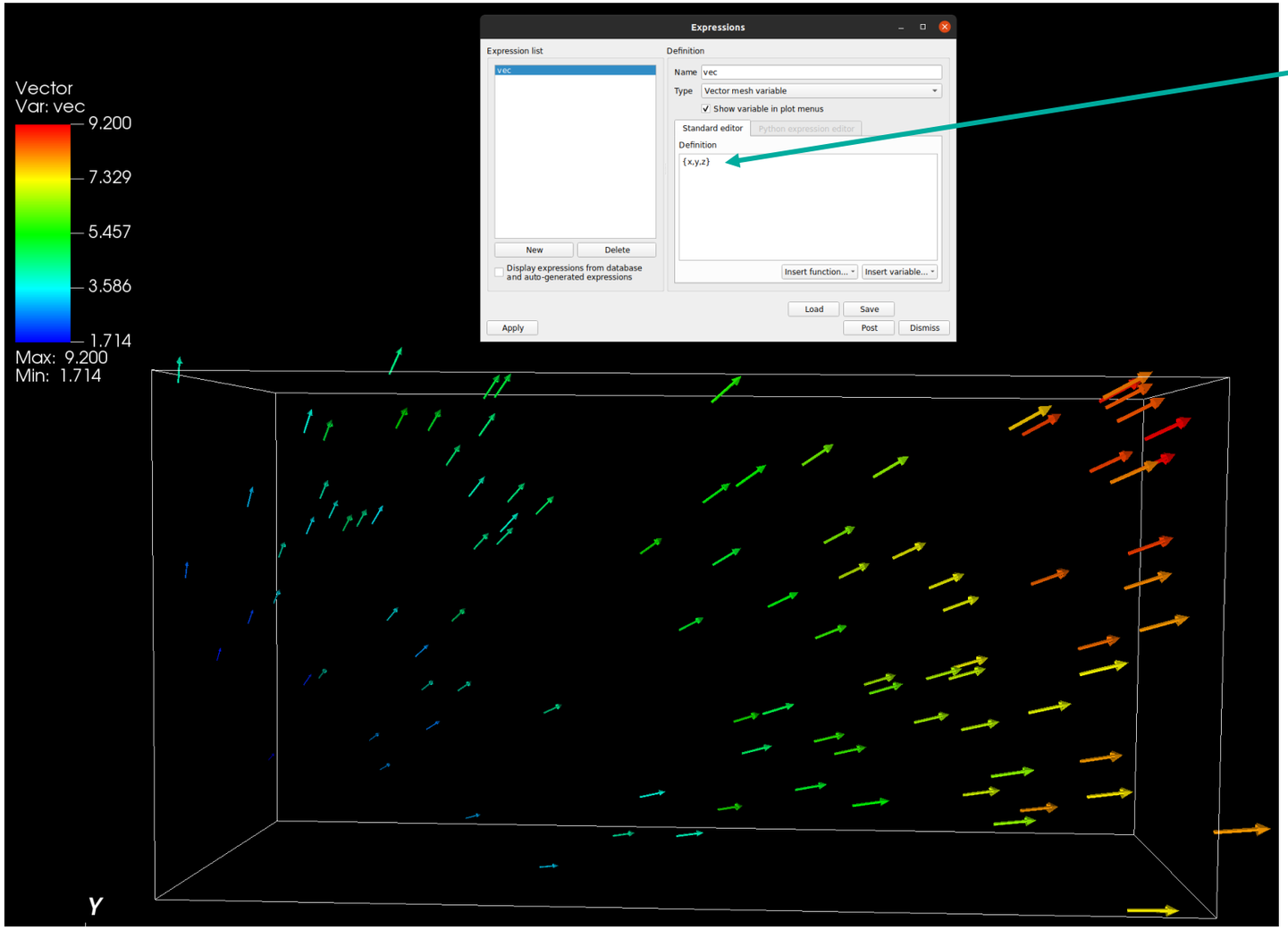
Query Window

Pick Options

Output Window

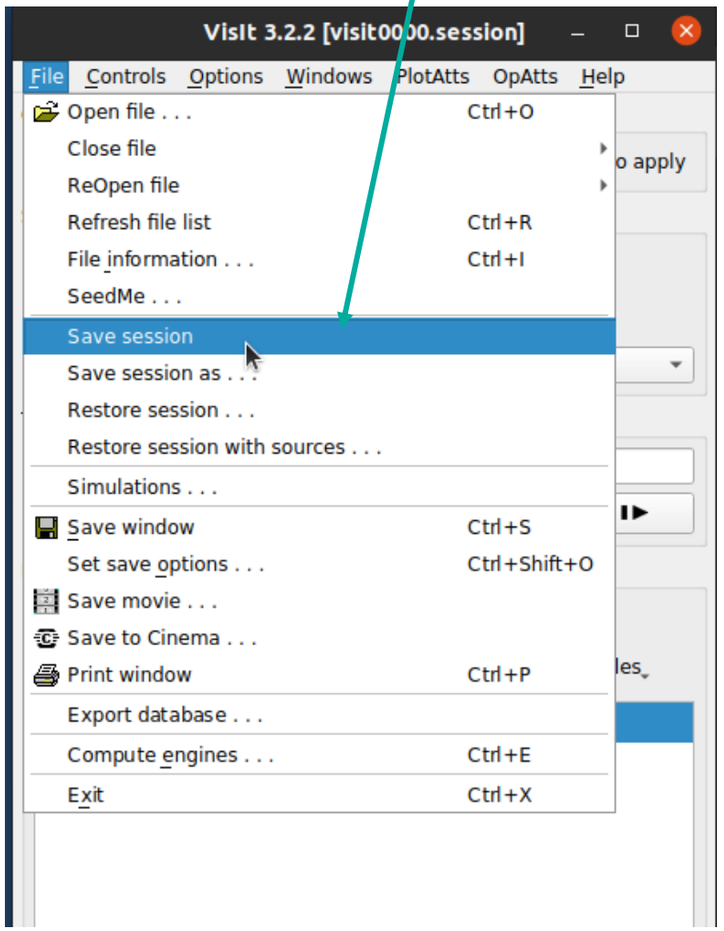


Custom Expressions / Save Session

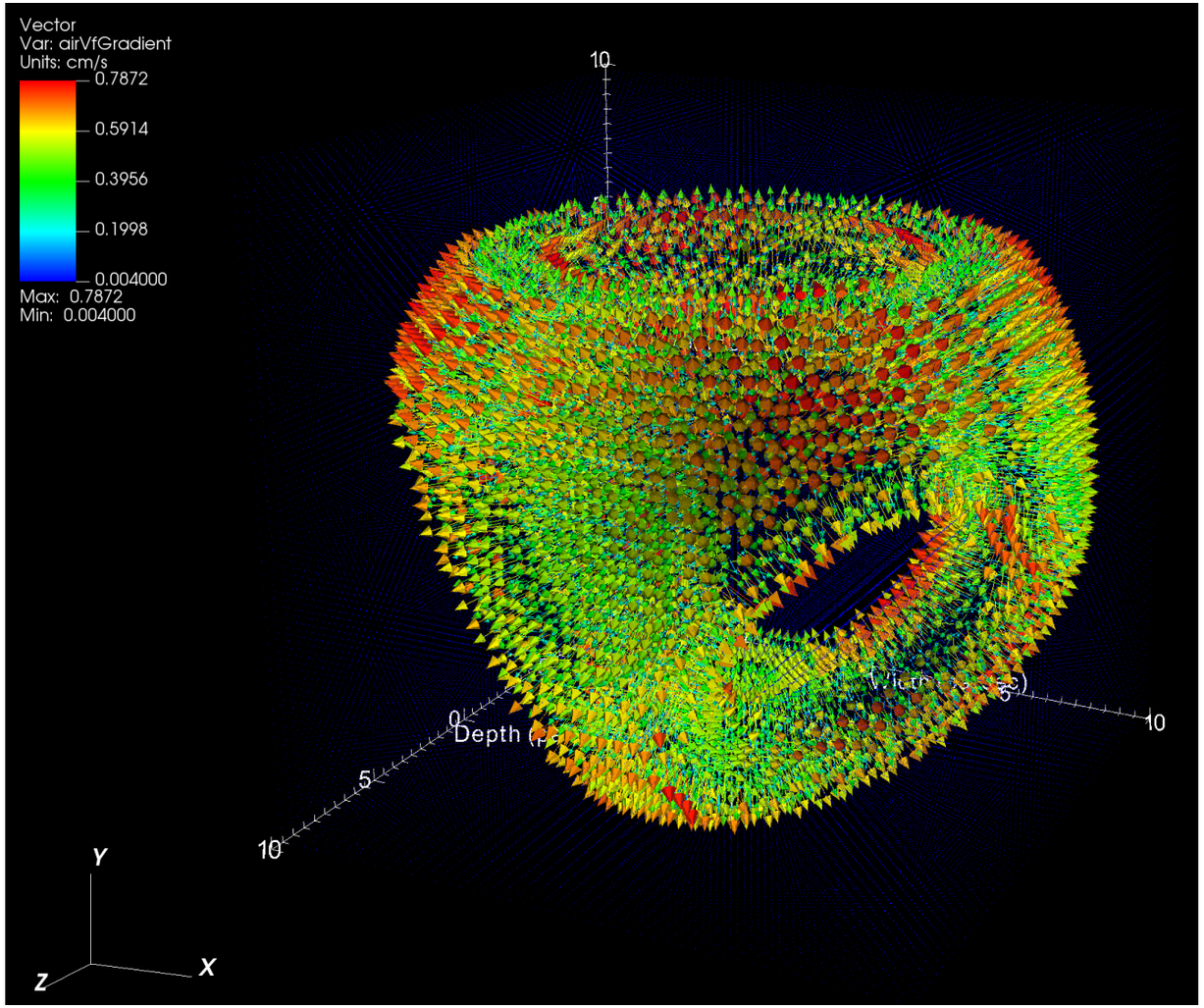


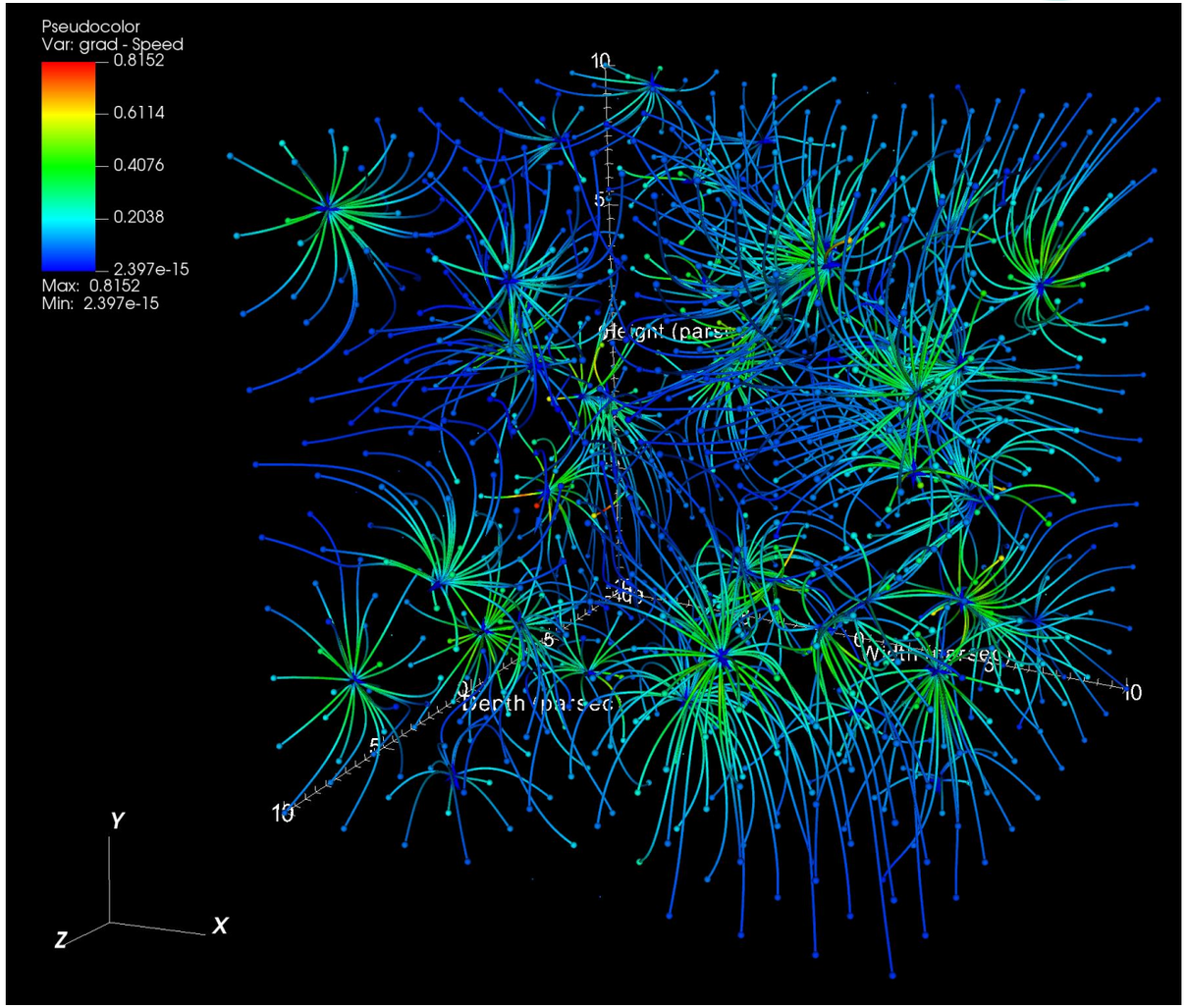
Custom Expression

Save Session Dialogue



Vector Plots / Streamlines





VisIt Wrap-up

Best Practices

Best Practices

How do I use ParaView or VisIt?

- If your data is small/manageable
  - Do your visualizations on your laptop, desktop, or IT Remote Workstation
- If your data is medium/large
  - Do interactive visualization on Ibex
    - Run it on your local machine and connect directly to Ibex to load/process/visualize
    - https://gitlab.kaust.edu.sa/kvl/KAUST_Visualization_Vignettes/-/tree/master/ParaView_Vignettes#using-paraview-interactively-on-ibex
    - https://gitlab.kaust.edu.sa/kvl/KAUST_Visualization_Vignettes/-/tree/master/VisIt_Vignettes#using-visit-interactively-on-ibex
- If your data is large/huge and you have a defined workflow
  - Do batch visualization on Shaheen
    - https://gitlab.kaust.edu.sa/kvl/KAUST_Visualization_Vignettes/-/tree/master/VisIt_Vignettes#expy
    - https://gitlab.kaust.edu.sa/kvl/KAUST_Visualization_Vignettes/-/tree/master/ParaView_Vignettes#expy
- If you have repeatable repetitive tasks
  - Do scripted or batch visualization

Thanks!

Contacts:

james.kress@kaust.edu.sa

help@vis.kaust.edu