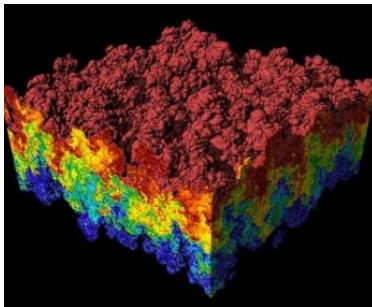
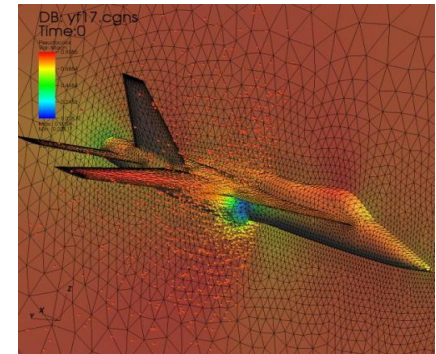
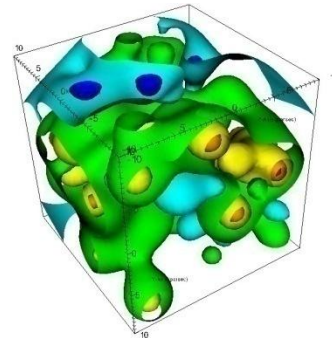


# Overview of VisIt

Workshop on Visualization of Large Scientific Data  
CINECA, Bologna, Italy



Vladimir Slavnić

Scientific Computing Laboratory  
Institute of Physics Belgrade Serbia  
e-mail: slavnic@ipb.ac.rs



# VisIt in general

- VisIt is distributed, parallel, visualization tool for visualizing data defined on two- and three-dimensional structured and unstructured meshes
- It is a free software application developed at Lawrence Livermore National Laboratory (LLNL) for visualizing and analyzing tera- to peta-scale range simulation datasets
- Used by:
  - Visualization experts
  - Simulation code developers
  - Simulation code consumers



# Main features [1/2]

- Rich feature set for scalar, vector, and tensor field visualization
  - Plots
  - Operators
- Quantitative analysis
  - Create derived variables using data from the database
  - Pick
  - Lineout
  - Queries
- Supports multiple mesh types



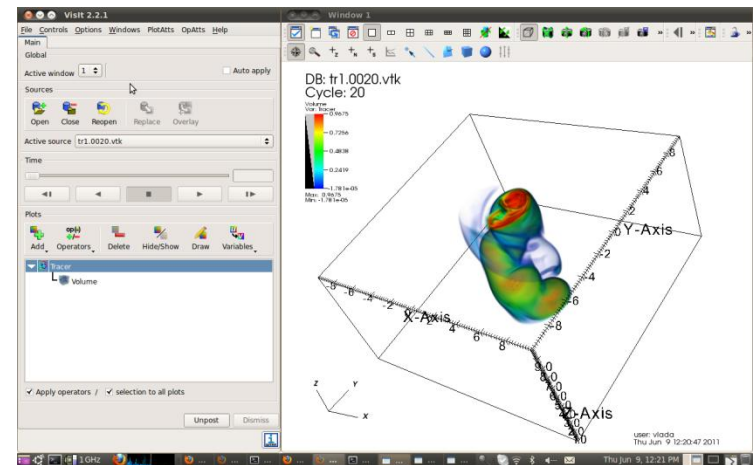
# Main features [2/2]

- Full-featured graphical user interface (GUI)
- Parallel and distributed architecture for visualizing large data sets
- In-situ visualization and analysis capabilities
- Interfaces with C++, Python, and Java
- Extensible with dynamically loaded plug-ins
  - New plots and operators can be developed as plug-ins
- Multi-platform support



# Get VisIt

- Available for download for free on the Web at <https://wci.llnl.gov/codes/visit/>
- Different platforms are supported with the same user interface
  - Binary distributions
    - Windows
    - Linux
    - Mac OS X
    - Java client library
  - Source code
    - build\_visit script available
  - Documentation
  - Frequently asked questions



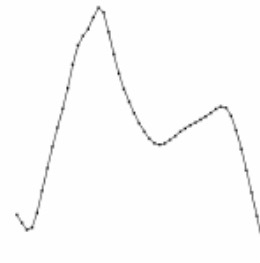
# Data support

- Many types of file formats are supported through VisIt database readers plugins:
  - Silo, VTK, Exodus, PDB, Mili, SAMRAI, BoxLib, Ensight, Vista...
- Database reader plug-ins can be developed for new formats
- Variable types:
  - Scalar
  - Vector
  - Tensor
  - Material

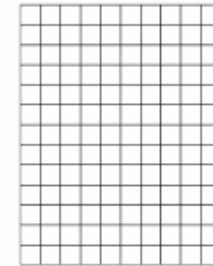


# Mesh types support

- 1D Curves
- 2D/3D meshes
  - Rectilinear
  - Curvilinear
  - Unstructured
  - Points
  - AMR
  - CSG



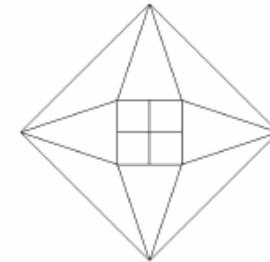
Curve



Rectilinear



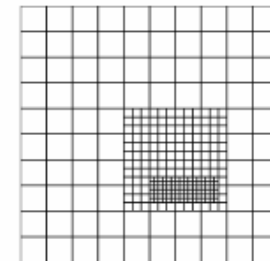
Curvilinear



Unstructured



Points



AMR

# Handling large datasets

- User can run VisIt on desktop computer and have VisIt process large data in parallel on a remote supercomputer
- Client/Server architecture
- Uses fast local graphics hardware
- MPI parallel compute engine
- Scalable rendering in parallel for largest datasets
- VisIt Top 50 - Leading examples of using VisIt at scale
  - [http://www.visitusers.org/index.php?title=VisIt\\_top\\_50](http://www.visitusers.org/index.php?title=VisIt_top_50)



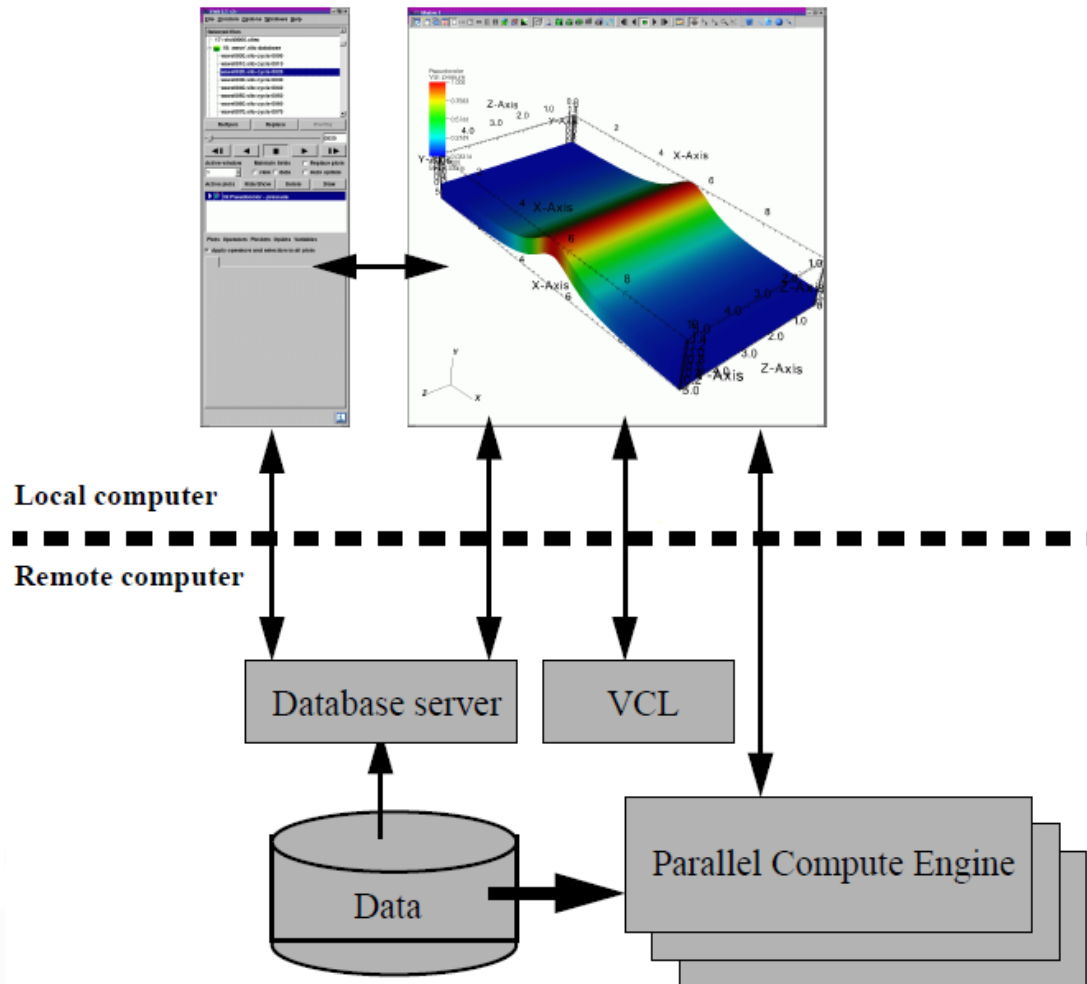


# VisIt architecture [1/2]

- 4 main components
  - Graphical User Interface (GUI)
  - Viewer
  - Database server
  - Compute engine
- GUI and Viewer usually meant to run locally on desktop computer
- Database server and parallel compute engine can run on remote computers where the data files are located and talk to the GUI and viewer running on desktop computer

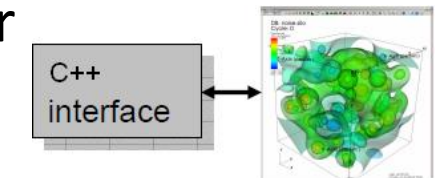
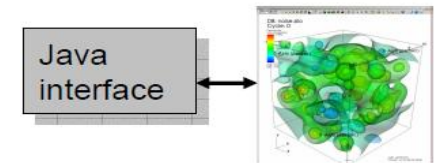
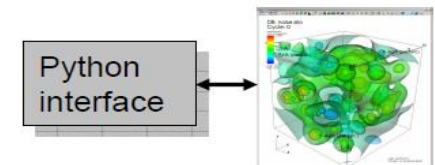
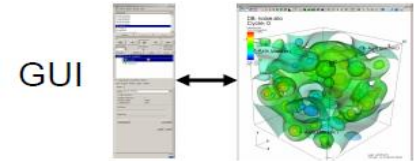


# Visit architecture [2/2]



# Supporting multiple interfaces

- Use VisIt as an application or a library
- Interfaces for controlling VisIt:
  - Graphical user interface
  - Python programming interface
  - Java programming interface
  - C++ programming interface
- All interfaces send commands to the viewer and in turn get the latest state from the viewer
- Use GUI when interaction is required
- Use Python interface to script actions or use VisIt as a batch mode movie generation tool



# Typical VisIt workflow

- Open database
- Create a plot
- Set plot attributes
- Apply operators to plot to modify data
- Set operator attributes
- Compute engine generates plot
- Plot is displayed in vis window

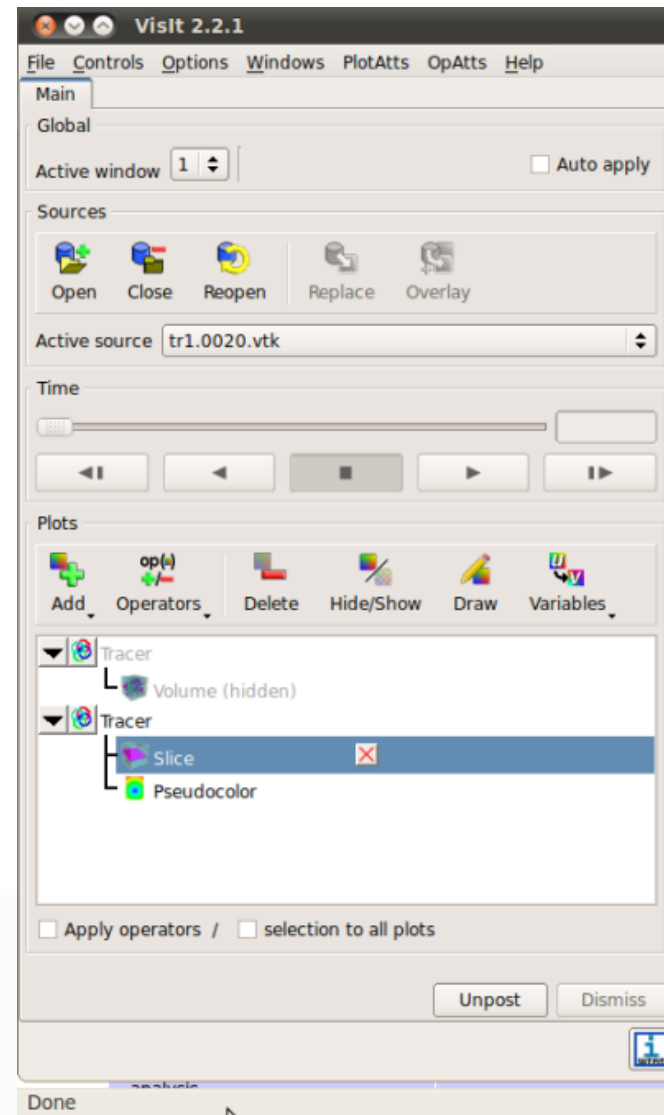


# Visit main windows



# GUI

- Select files to visualize
- Create and manage plots
- Set plot attributes
- Add operators
- Set look and feel for visualization

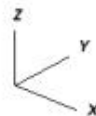
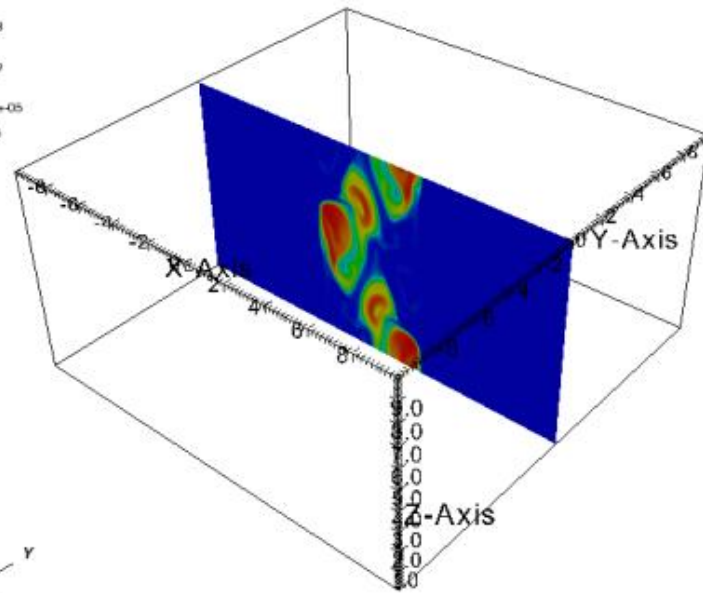


# Viewer

- Viewer windows, or vis windows, display all of the data being visualized
- Mouse navigation
- Up to 16 vis windows
- Popup menu
- Toolbars



DB: tr1.0020.vtk  
Cycle: 20  
Pseudocolor  
Var: tracer  
0.9675  
-0.7256  
-0.4838  
-0.2419  
-1.781e-05  
Max: 0.9675  
Min: -1.781e-05



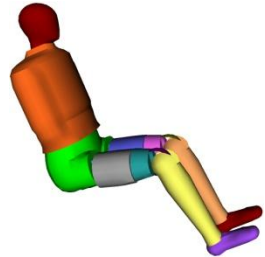
user: vlada  
Thu Jun 9 15:38:19 2011

# Plots



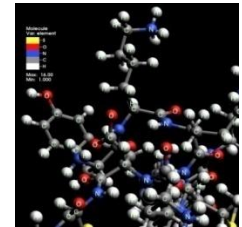


# VisIt plots



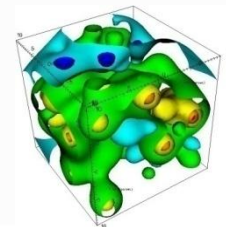
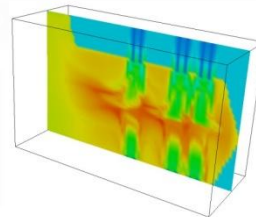
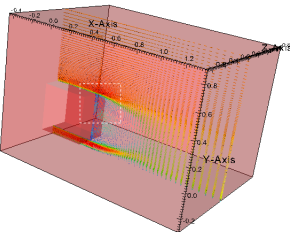
- Viewable objects, created from a database, that can be displayed in a visualization window

- Visual representation of the data being examined



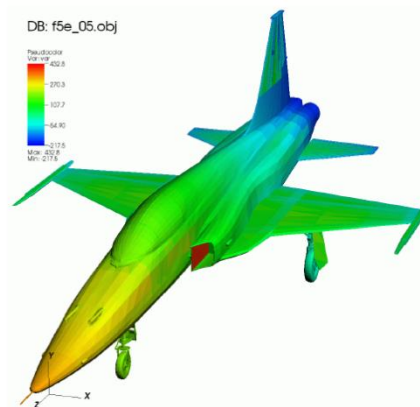
- VisIt has several types of plots, including: Pseudocolor, Mesh, Volume, Subset...

- Plots come from plug-ins so user can extend VisIt's plotting capabilities by writing a new plug-in

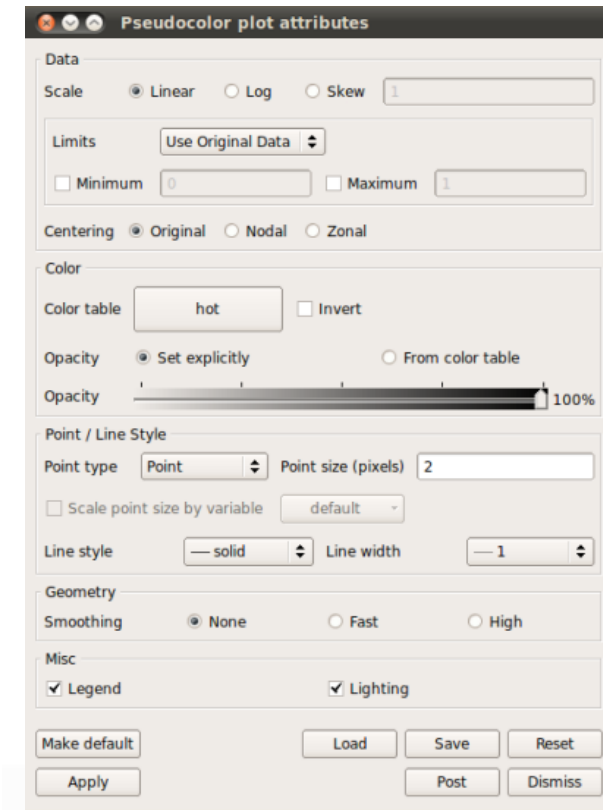
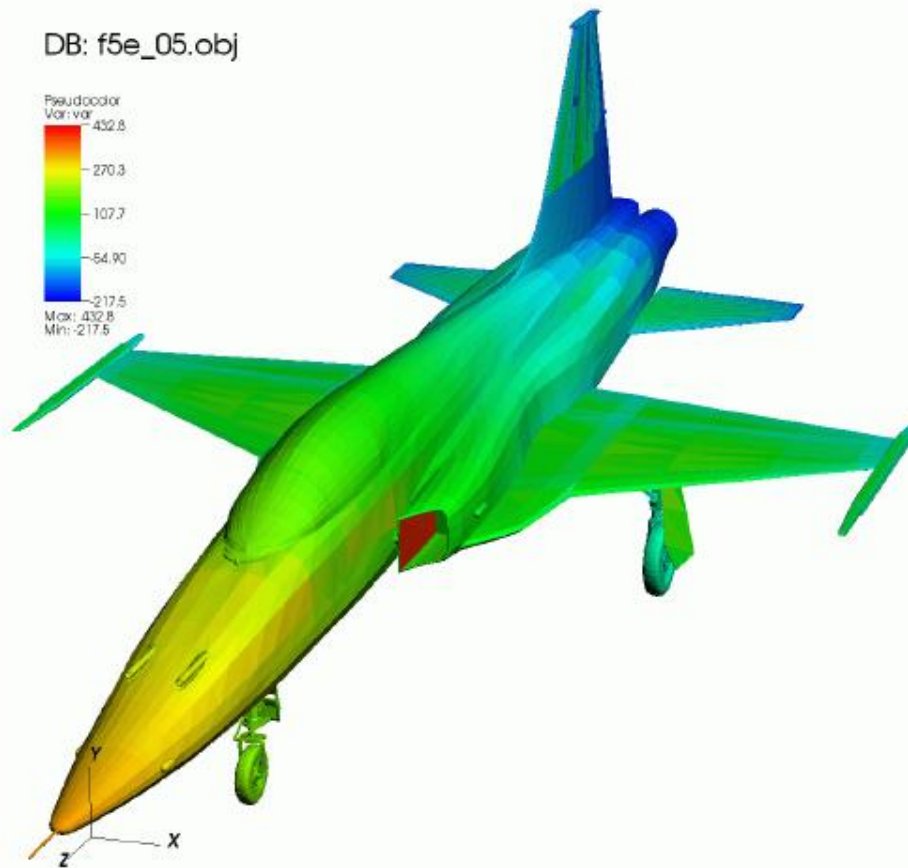


# Pseudocolor plot

- This plot maps a scalar variable to colors and uses the colors to “paint” values onto the variable’s mesh
- This plot is used when user wants to investigate the behavior of a scalar variable
- Pseudocolor plot accepts scalar variables

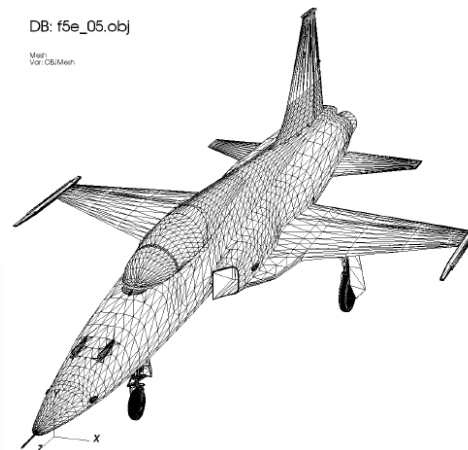


# Pseudocolor plot example



# Mesh plot

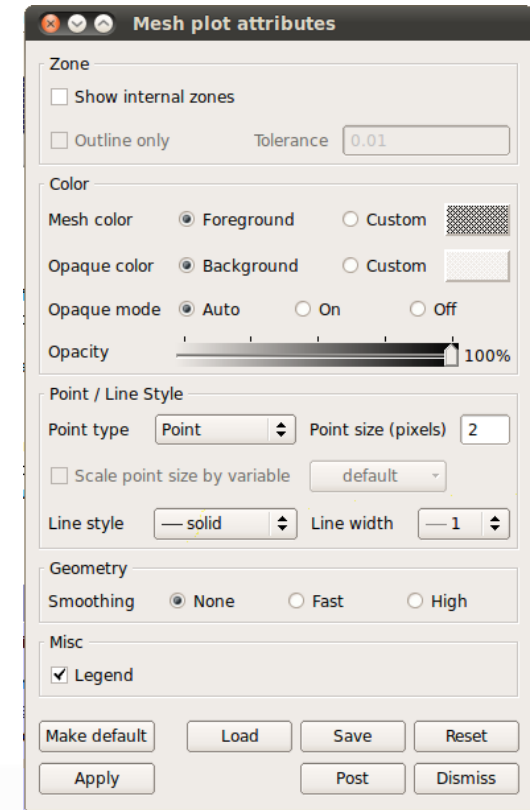
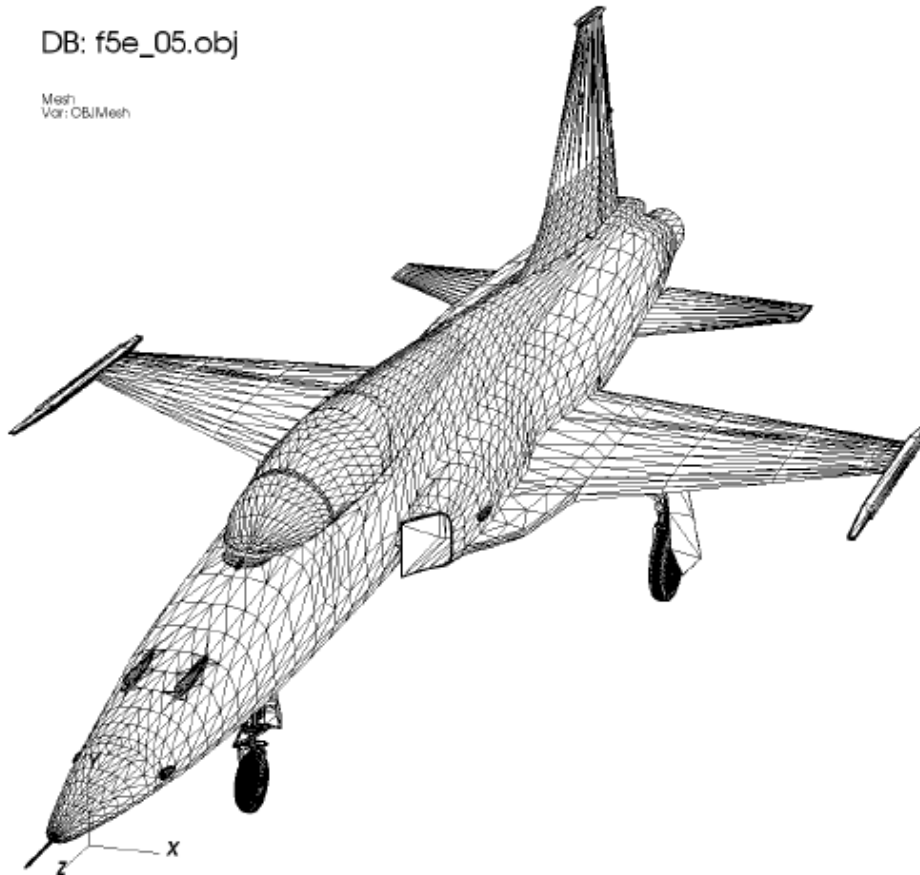
- Displays the computational mesh over which a database's variables are defined
- Often added to the visualization window when other plots are visualized to allow individual cells to be clearly seen.
- This plot accepts meshes



# Mesh plot example

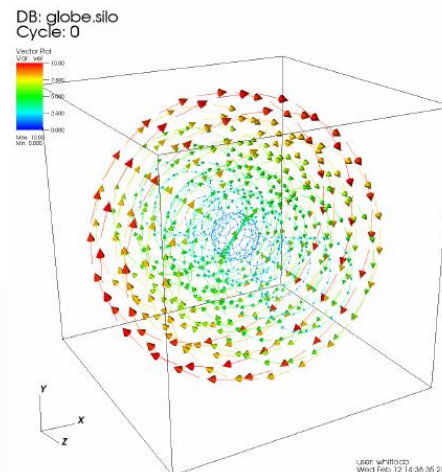
DB: f5e\_05.obj

Mesh  
Var: CBI/Mesh



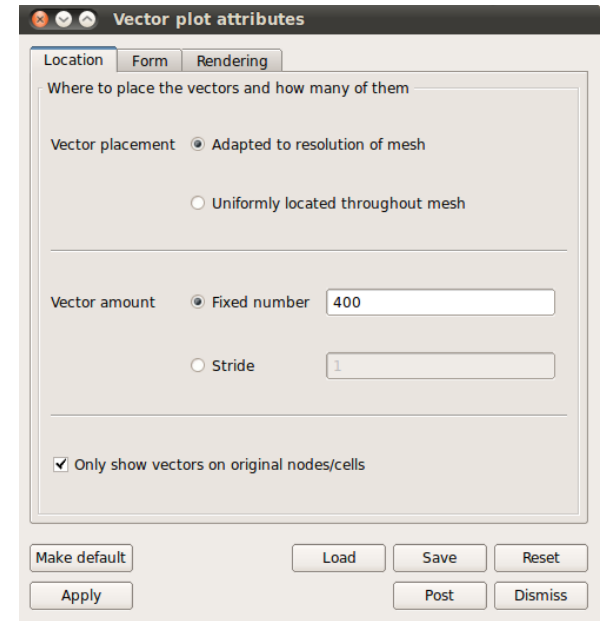
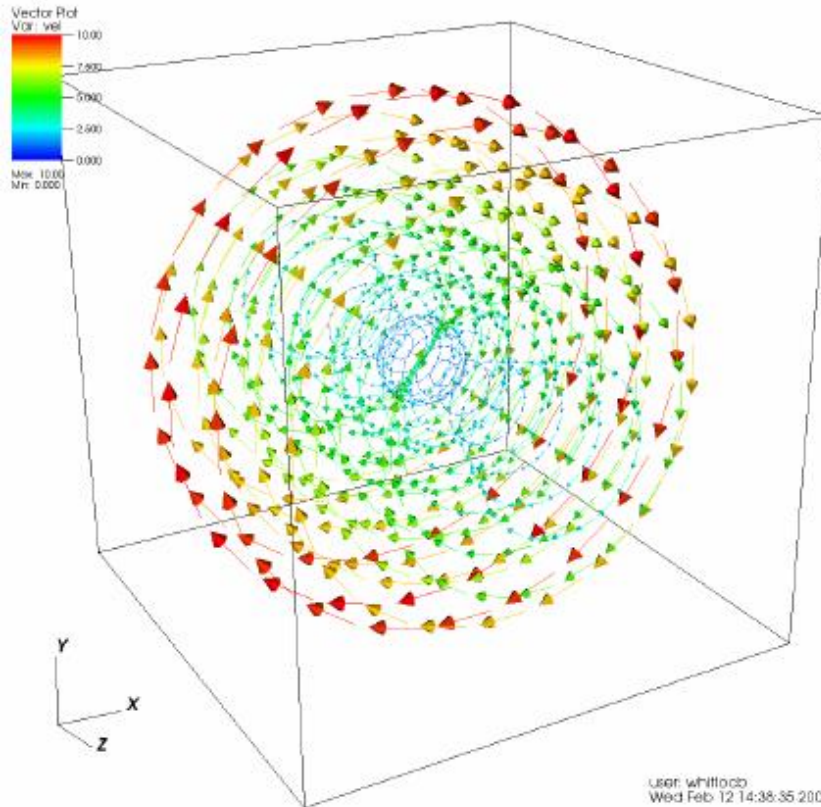
# Vector plot

- Vector plot displays vector variables as small glyphs that indicate the direction and magnitude of vectors in a vector field
- This plot is used when user wants to investigate the behavior of a vector variable
- This plot accepts vector variables



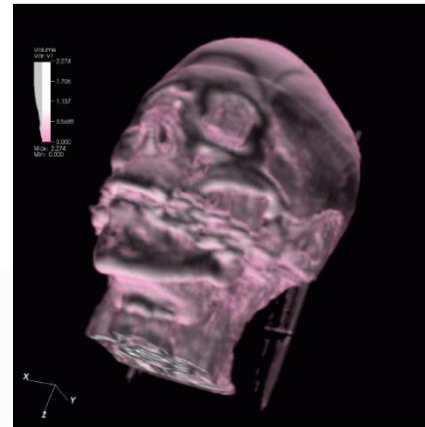
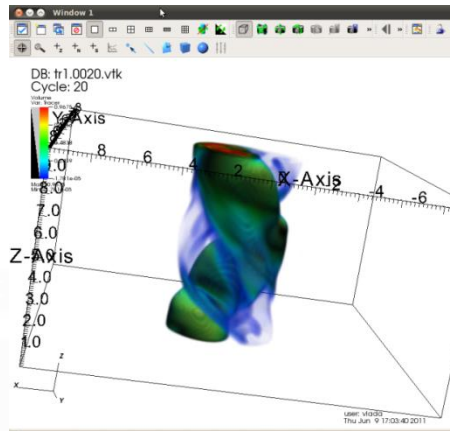
# Vector plot example

DB: globe.silo  
Cycle: 0



# Volume plot

- Volume plot uses both color and transparency to visualize 3D scalar variables
- This plot should be used user wants to look at internal features of a scalar variable while keeping all of the plot at least partially visible
- This plot accepts 3D scalar variables



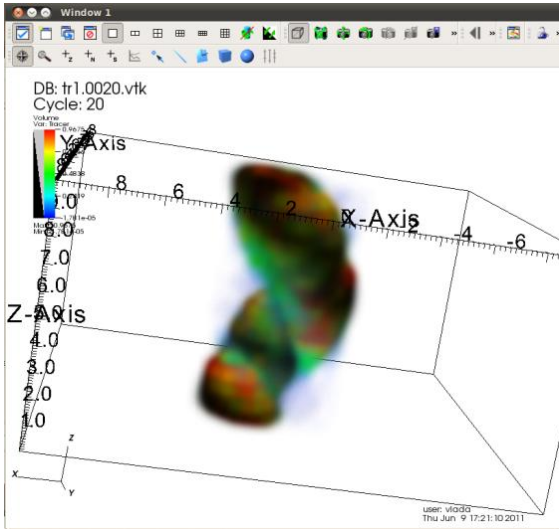


# Volume plot rendering methods

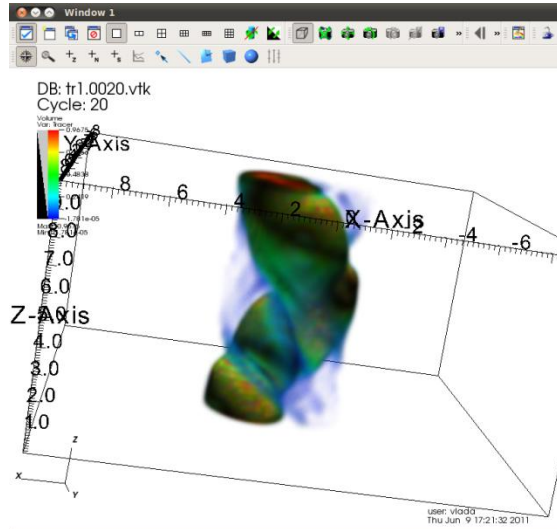
- Volume plot has different rendering methods
  - Splatting renderer
    - Not accurate
    - Hardware accelerated and moderately fast
  - 3D texturing renderer
    - More accurate than splatting
    - Hardware accelerated and very fast
    - 3D texturing must be supported on the graphics card
  - Software raycasting renderer
    - Very accurate
    - Slow
    - Memory intensive
    - Parallelized
    - Can handle larger data sizes
  - SLIVR
    - Volume rendering library that uses shaders on the graphics card to produce images approximating high-quality software volume rendering



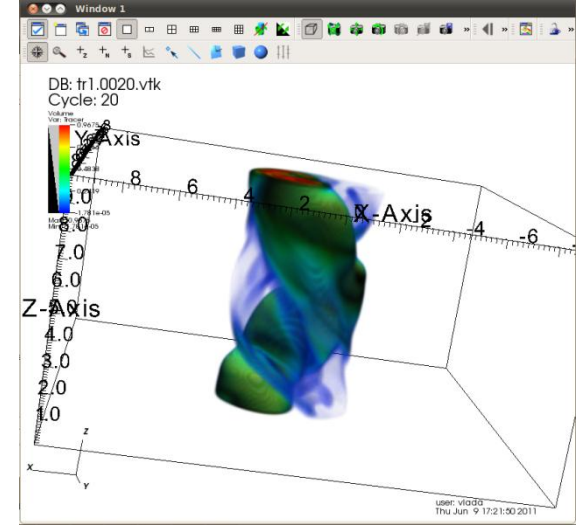
# Splatting samples number example



50,000 samples



500,000 samples



5,000,000 samples

# Operators



# Operators [1/2]

- An operator is a filter that is applied to a database variable before the compute engine uses that variable to generate a plot
- VisIt provides several standard operator types that allow various operations to be performed on plot data.
- User can extend VisIt's data manipulation capabilities by writing a new operator plug-in



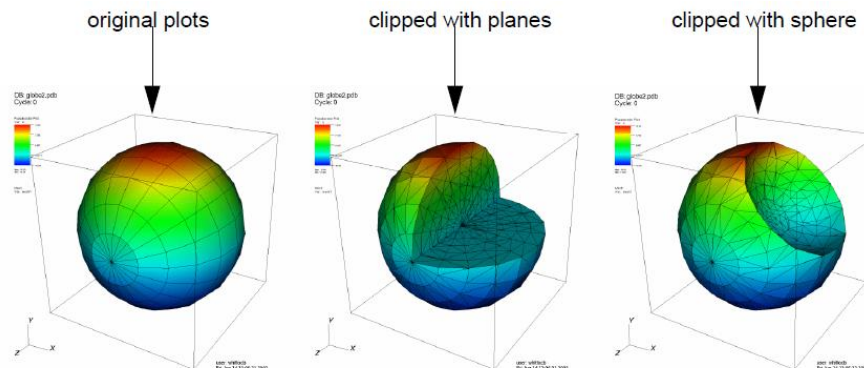
# Operators [2/2]

- When an operator is applied to a plot, it modifies the data that the plot uses to generate a visualization
- Any number of operators can be applied to a plot
- By using a series of operators, user can create very sophisticated visualizations

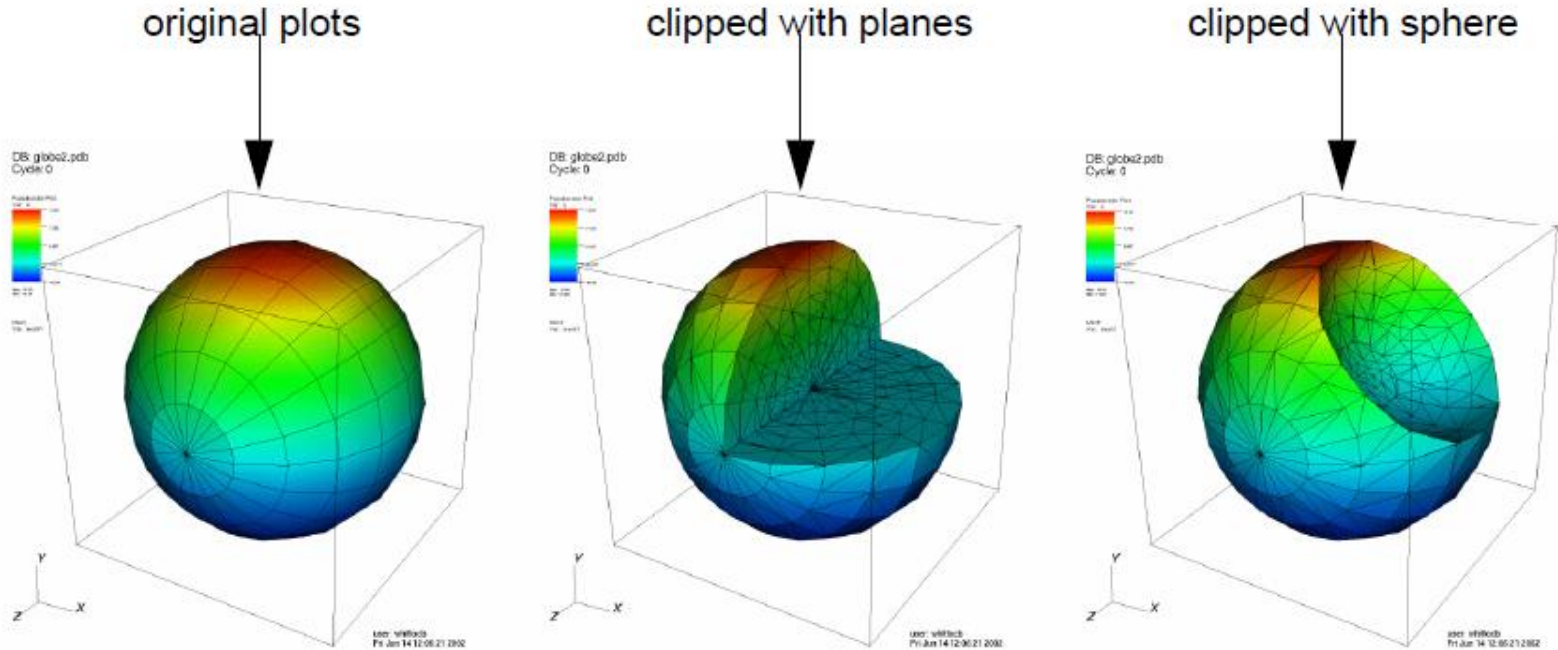


# Clip operator

- The Clip operator clips 2D or 3D plots against planes or a sphere to remove sections of the plots
- Use this operator when you want to see a cross section of a 3D plot, while still leaving the plot in 3D
- Clip operator also clips 2D databases.

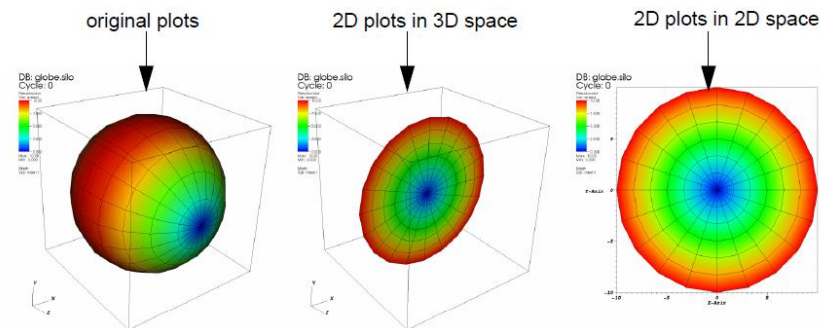


# Clip operator example



# Slice operator

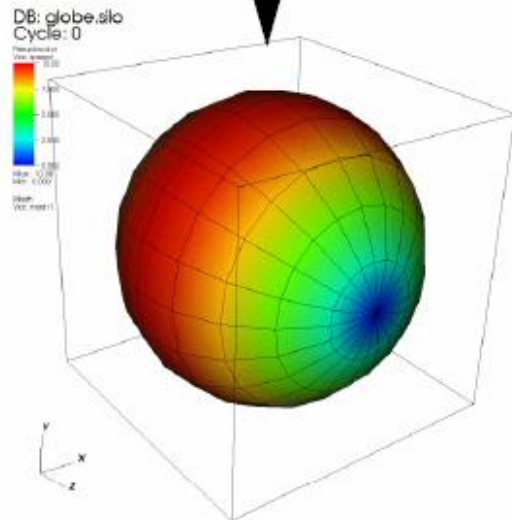
- This operator slices a 3D database with a plane that can have an arbitrary orientation
- Plots to which the Slice operator has been applied are turned into 2D planar surfaces that are coplanar with the slice plane
- The resulting plot can be left as a 2D slice in 3D space or it can be projected to 2D space where other operations can be done to it



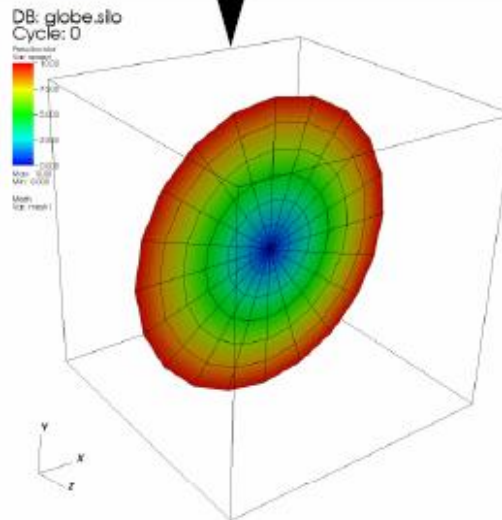


# Slice operator example

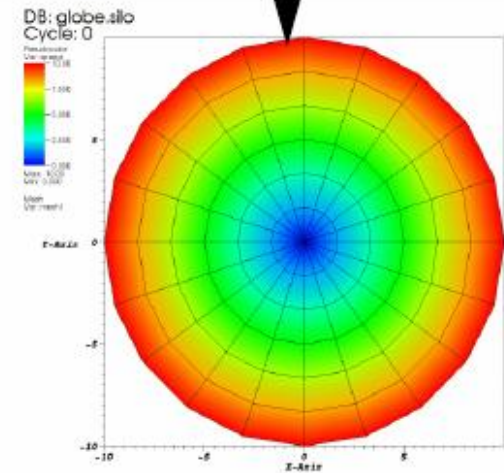
original plots



2D plots in 3D space

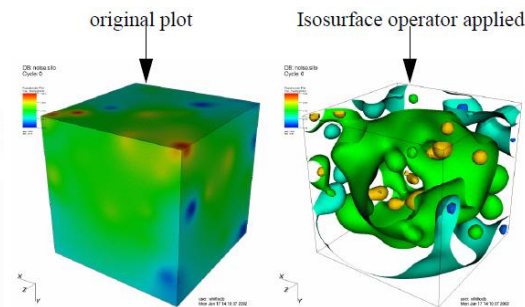


2D plots in 2D space



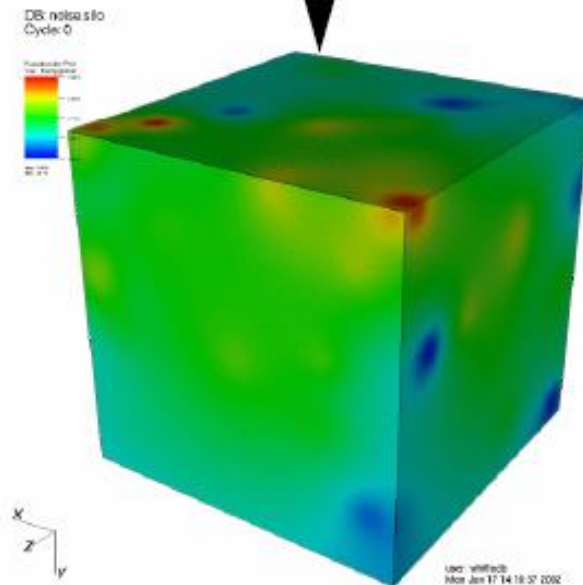
# Isosurface operator

- The Isosurface operator extracts surfaces from 2D or 3D databases and allows them to be plotted
- The Isosurface operator takes as input a database and a list of values and creates a set of isosurfaces through the database
- An isosurface is a surface where every point on the surface has the same data value

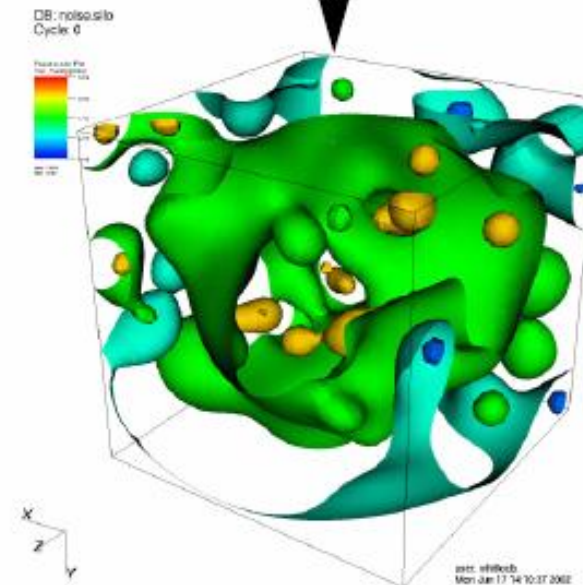


# Isosurface operator example

original plot

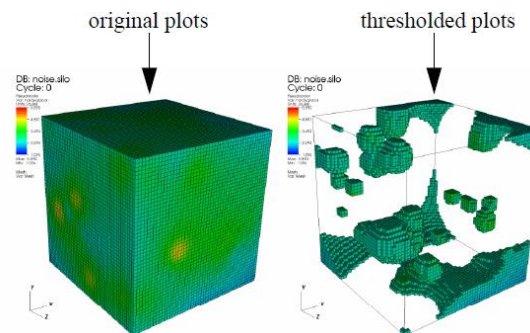


Isosurface operator applied



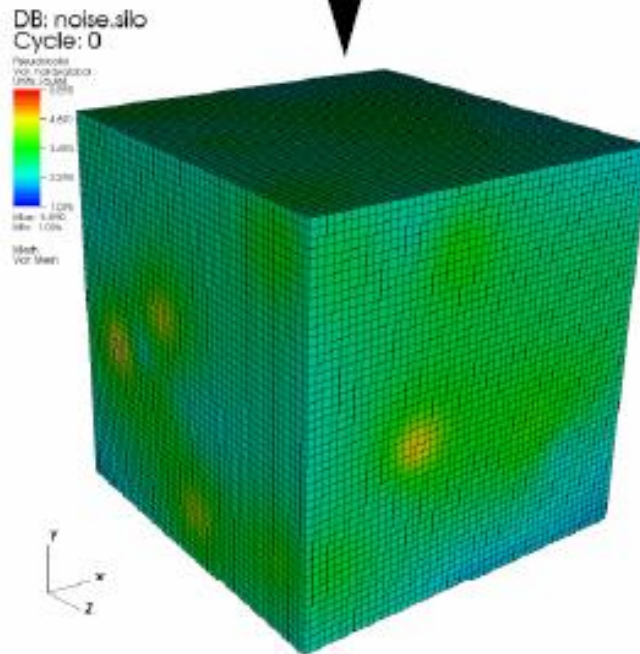
# Threshold operator

- This operator extracts cells from 2D and 3D databases where the plotted variable falls into a specified range
- User might use this operator when searching for cells with certain values. One such example is searching for the cell with the minimum or maximum value for the plotted variable

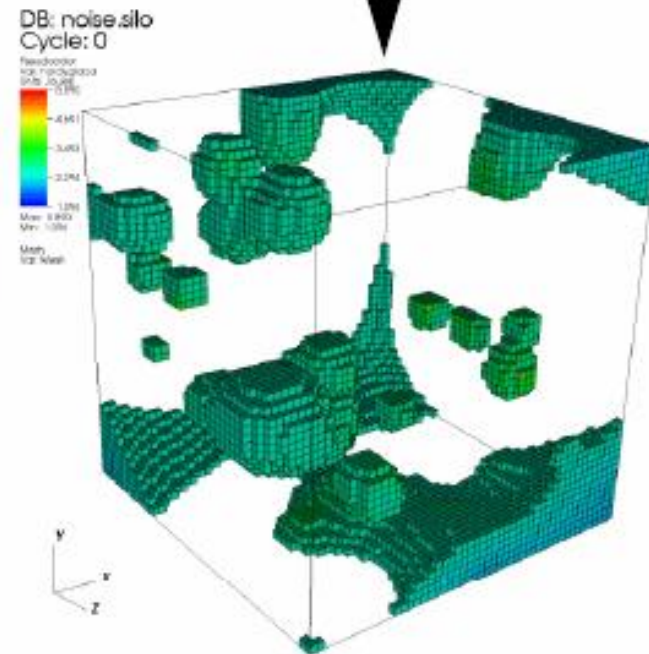


# Threshold operator example

original plots



thresholded plots



# Quantitative Analysis



# Expressions [1/2]

- VisIt allows you to create new derived variables from values in your database using VisIt's expression language
- Use expressions to derive quantities that were not stored in your database
- Expressions can operate on scalars, vectors, tensors, or on meshes
- VisIt provides built-in math functions
  - Trigonometric functions
  - Logarithmic functions
  - Vector functions



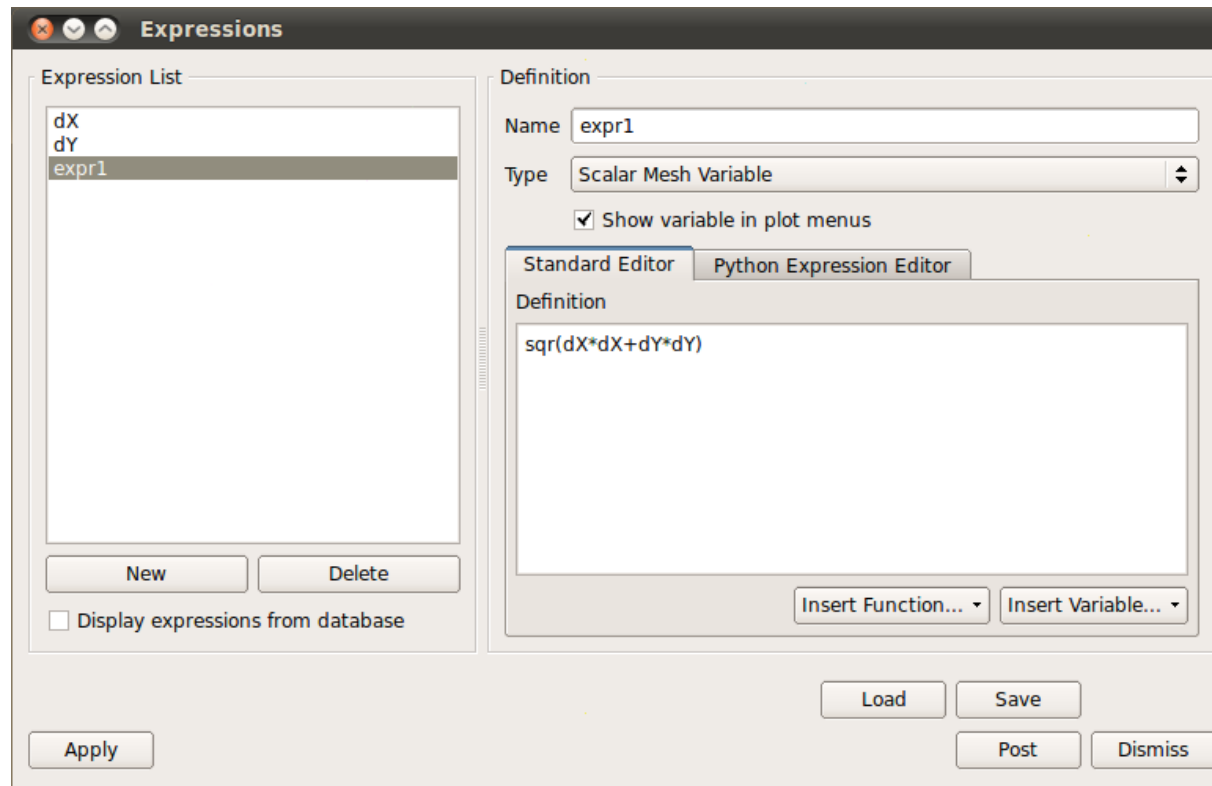
# Expressions [2/2]

- Expressions are extremely powerful because they allow you to analyze new data without necessarily having to rerun a simulation.
- Variables created using expressions behave just like variables stored in a database - they appear in the plot menu and can be visualized using VisIt's plots.





# Expressions window



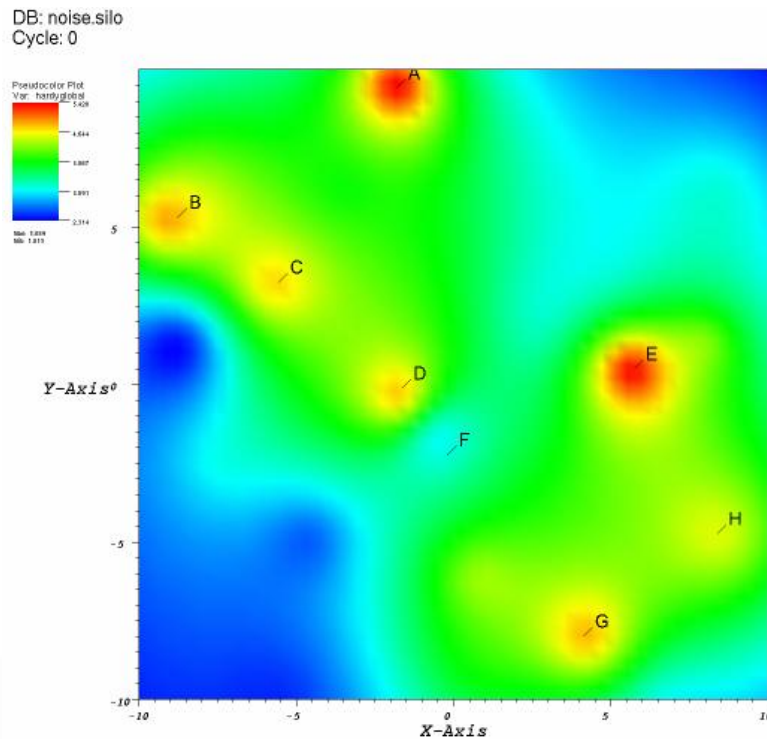
# Pick [1/2]

- User is able to Interactively pick values from visualized data using vis window's Pick mode
- Each click causes VisIt to determine variable values for selected plot at pick point
- Essential tool for performing data analysis
- Enter a pick mode using vis window's mode menu



# Pick [2/2]

- Each pick point leaves a marker that you can use to match with the pick information displayed in the Pick window



Pick

A B C D E

```
tr1.0020.vtk timestep 0
mesh
Point: <-4.51934, 5.4792, 9.975>
Zone: 16381152
Incident Nodes: 16495253 16495254 16495541
16495542 16578197 16578198 16578485
16578486
Tracer: <nodal>
(16495253) = -0
(16495254) = -0
```

Max Tabs: 8 Save Picks as...

Variables: default

Float Format: %g

Concise Output

Show Mesh Name  Show Timestep

Display incident nodes/zones

Display global nodes/zones

Display reference pick letter

Display for Nodes

Id  Domain-Logical Coords

Physical Coords  Block-Logical Coords

Display for Zones

Id  Domain-Logical Coords

Block-Logical Coords

Automatically show window

Don't clear this window Clear Picks

Create time curve with next pick Repeat Pick

Time curve use picked coordinates

user: whitlock  
Tue Jun 18 14:34:03 2002



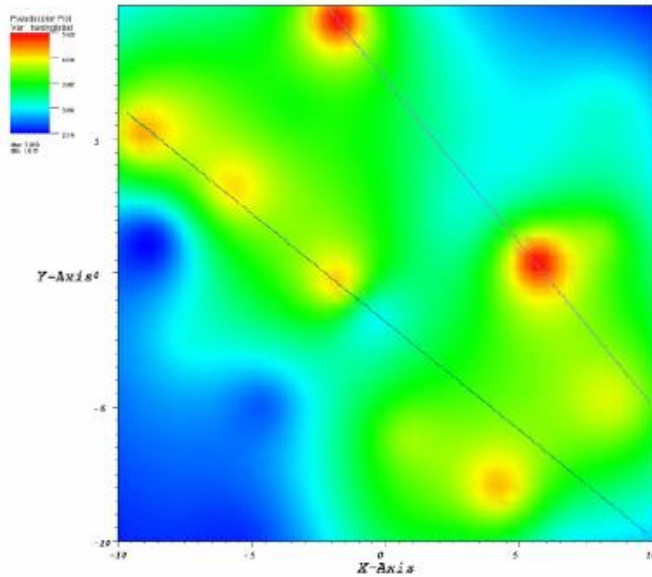
# Lineout [1/2]

- Extracts 1D curves from higher dimensional data
- Curves are easy to compare
- Curve plots are often more useful than 2D Pseudocolor plots because they allow the data along a line to be seen spatially as a 1D curve instead of relying on differences in color to convey information

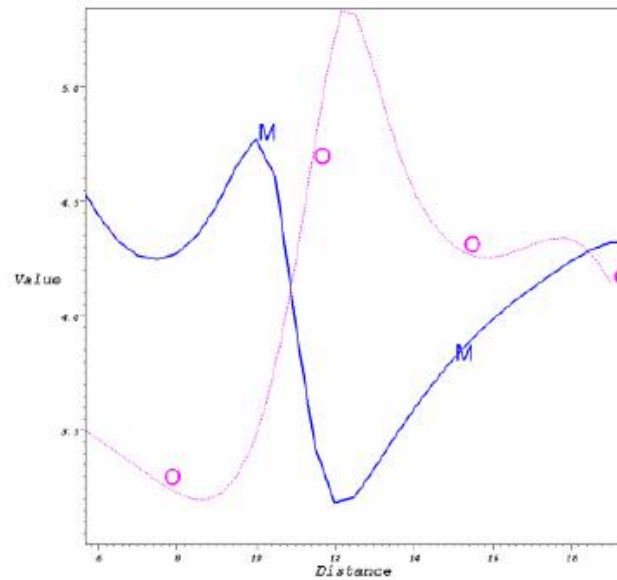


# Lineout [2/2]

DB noise silo  
Cycle: 0



user: whiflob  
Tue Jun 13 14:34:03 2002



user: whiflob  
Tue Jun 10 14:41:32 2002



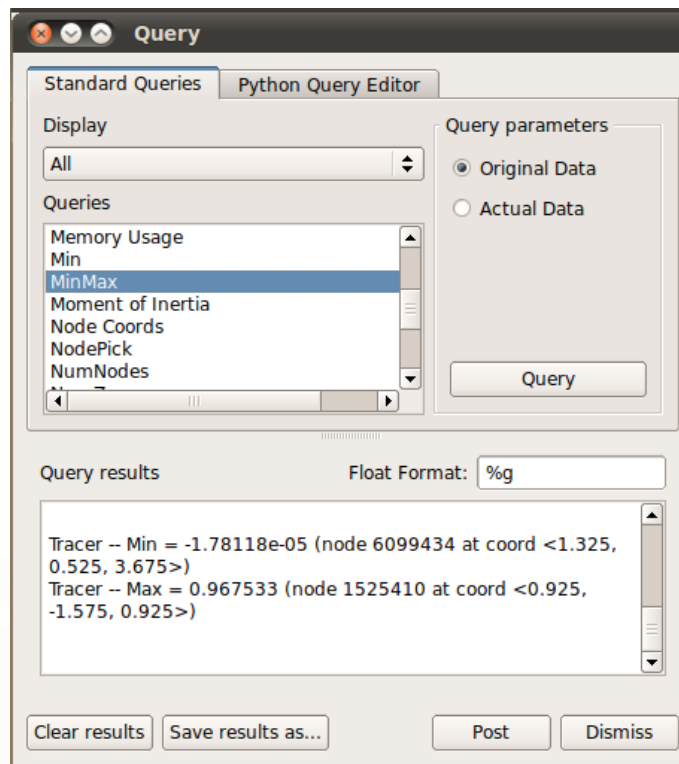
# Queries

- VisIt allows gathering of quantitative information from the database being visualized through the use of queries.
- A query is a type of calculation that can either return values from the database or values that are calculated from data in the database
- User can compute values about:
  - An entire database
  - A plot
  - A point in a database
  - A linear path through a database



# Query window

- Lists all available queries
- Lets user enter query parameters
- Displays query output



# Remote visualization





# Remote visualization overview

- Simulations are almost always run on a powerful supercomputer
- Databases usually reside on the computer that generated the data
- Run VisIt on local workstation but do the data I/O and processing on the computer that generated the data
  - The GUI and viewer run locally while the database server and parallel compute engine run on the remote supercomputer
  - Moving data is not necessary



# Run in distributed mode

- Procedure for running VisIt in distributed mode is no different than it is for running in single-computer mode
- To run in Distributed mode, in file selection window user just should specify remote host (by typing or from the Hosts list)
- Once the database server is launched on the remote computer, the files for the remote computer are available to user
- To connect to a simulation .sim (or .sim2) files corresponding to the simulations to which user wants to connect must be selected
- Create plots as usual using a selected remote files

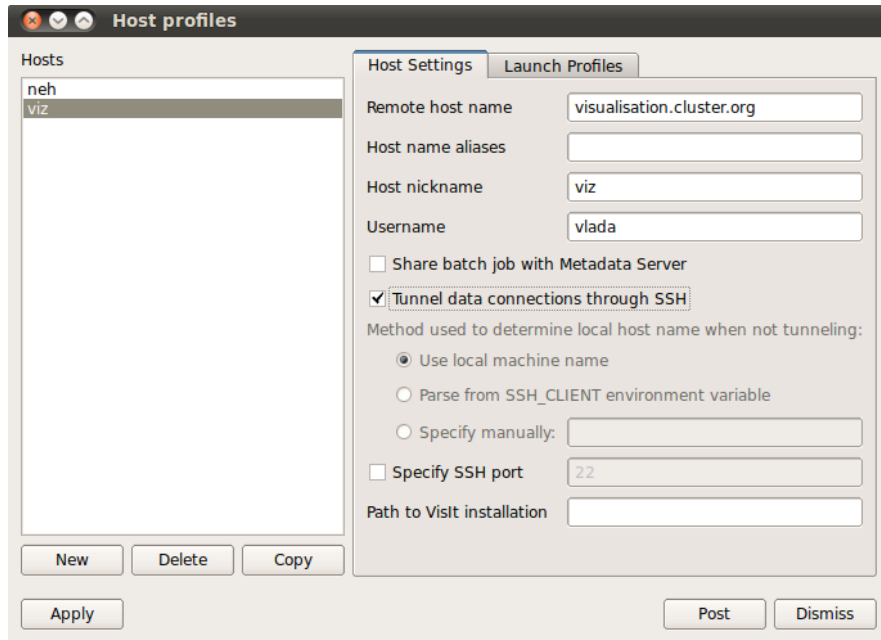


# Host profiles

- When VisIt launches a compute engine, it looks for a host profile
  - Contains information VisIt uses to launch a compute engine on a remote computer
    - remote user name
    - number of processors
    - parallel launch method
    - (other options)
- User can define multiple host profiles for any given computer
  - Common to have separate host profiles for running VisIt in serial and parallel



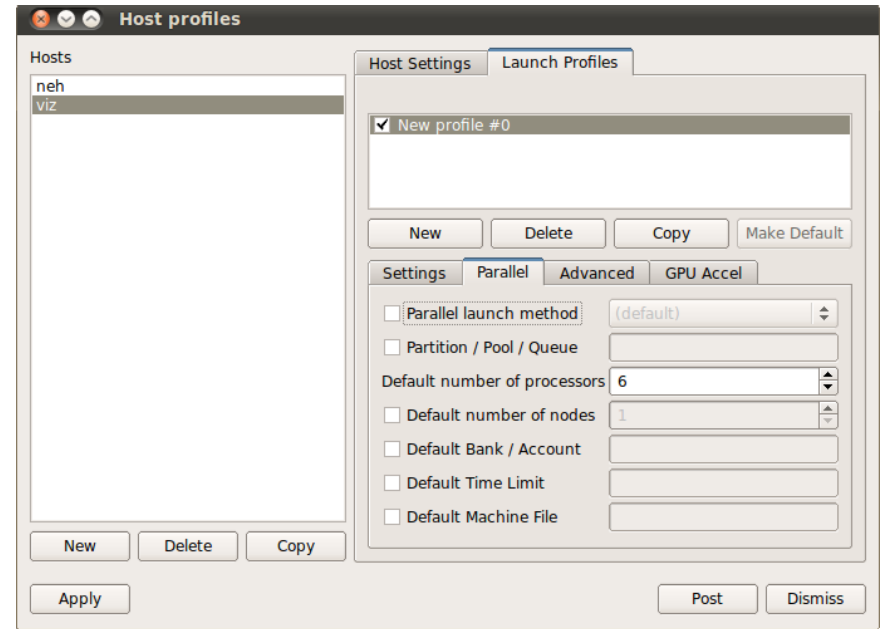
# Host profiles window



Host profiles window showing the Host Settings tab. The Hosts list contains 'neh' and 'viz'. The Host Settings form includes:

- Remote host name: visualisation.cluster.org
- Host name aliases: (empty)
- Host nickname: viz
- Username: vlada
- Share batch job with Metadata Server
- Tunnel data connections through SSH
- Method used to determine local host name when not tunneling:
  - Use local machine name
  - Parse from SSH\_CLIENT environment variable
  - Specify manually: (empty)
- Specify SSH port: 22
- Path to Visit installation: (empty)

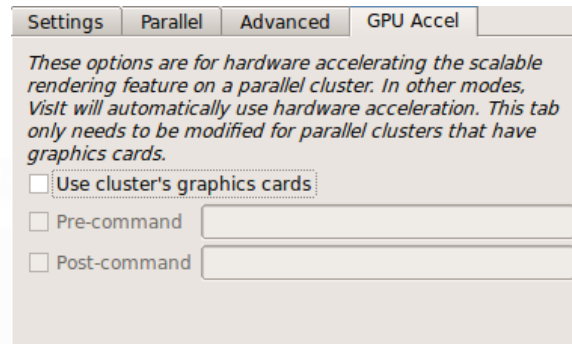
Buttons: New, Delete, Copy, Apply, Post, Dismiss



Host profiles window showing the Parallel tab. The Hosts list contains 'neh' and 'viz'. The Parallel settings form includes:

- New profile #0
- Buttons: New, Delete, Copy, Make Default
- Settings: Parallel, Advanced, GPU Accel
- Parallel launch method: (default)
- Partition / Pool / Queue: (empty)
- Default number of processors: 6
- Default number of nodes: 1
- Default Bank / Account: (empty)
- Default Time Limit: (empty)
- Default Machine File: (empty)

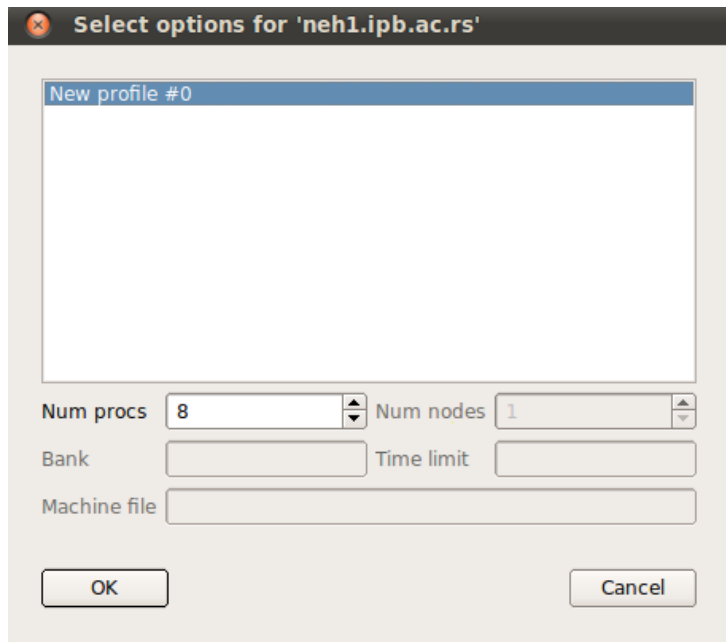
Buttons: New, Delete, Copy, Apply, Post, Dismiss



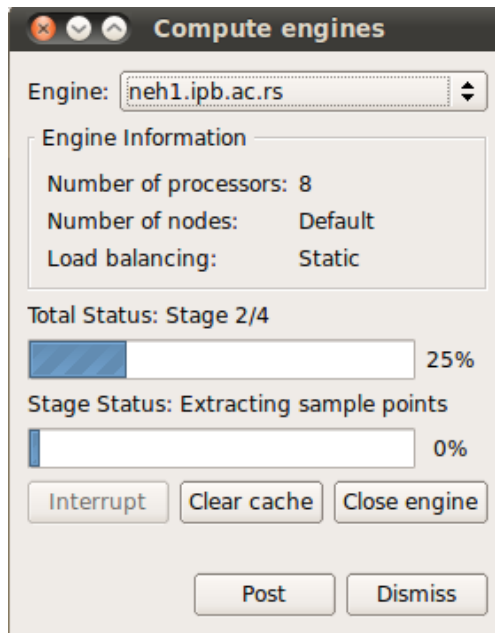
Host profiles window showing the GPU Accel tab. The GPU Accel settings form includes:

- Use cluster's graphics cards
- Pre-command: (empty)
- Post-command: (empty)

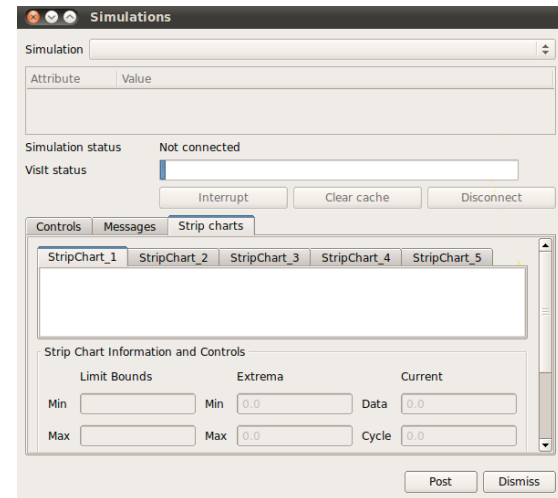
# Remote visualization windows



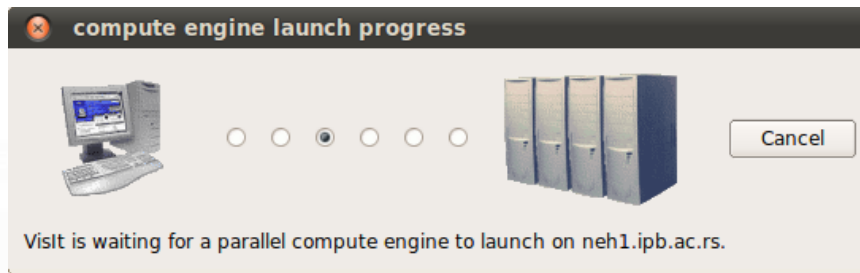
Engine option window



Compute engine window



Simulation window



Connection progress dialog

# Visit can offer more!

- In-situ visualization:
  - Adds visualization capabilities inside the simulation so it can visualize the data
  - Visualization routines get direct access to the simulation's memory
- Movies creation
  - Different animation and keyframing options
- Interactive tools
- Scripting
- Database correlations and comparison
- Various rendering options
- ... and much more!



# Help and docs

- Documentation

<https://wci.lnl.gov/codes/visit/doc.html>

- FAQ

<https://wci.lnl.gov/codes/visit/FAQ.html>

- Visit user community web site

<http://www.visitusers.org>

- Visit users forum

<http://www.visitusers.org/forum/forum.pl>



# Questions?

