BLUE WATERS

SUSTAINED PETASCALE COMPUTING

2/11/15

Blue Waters Visualization

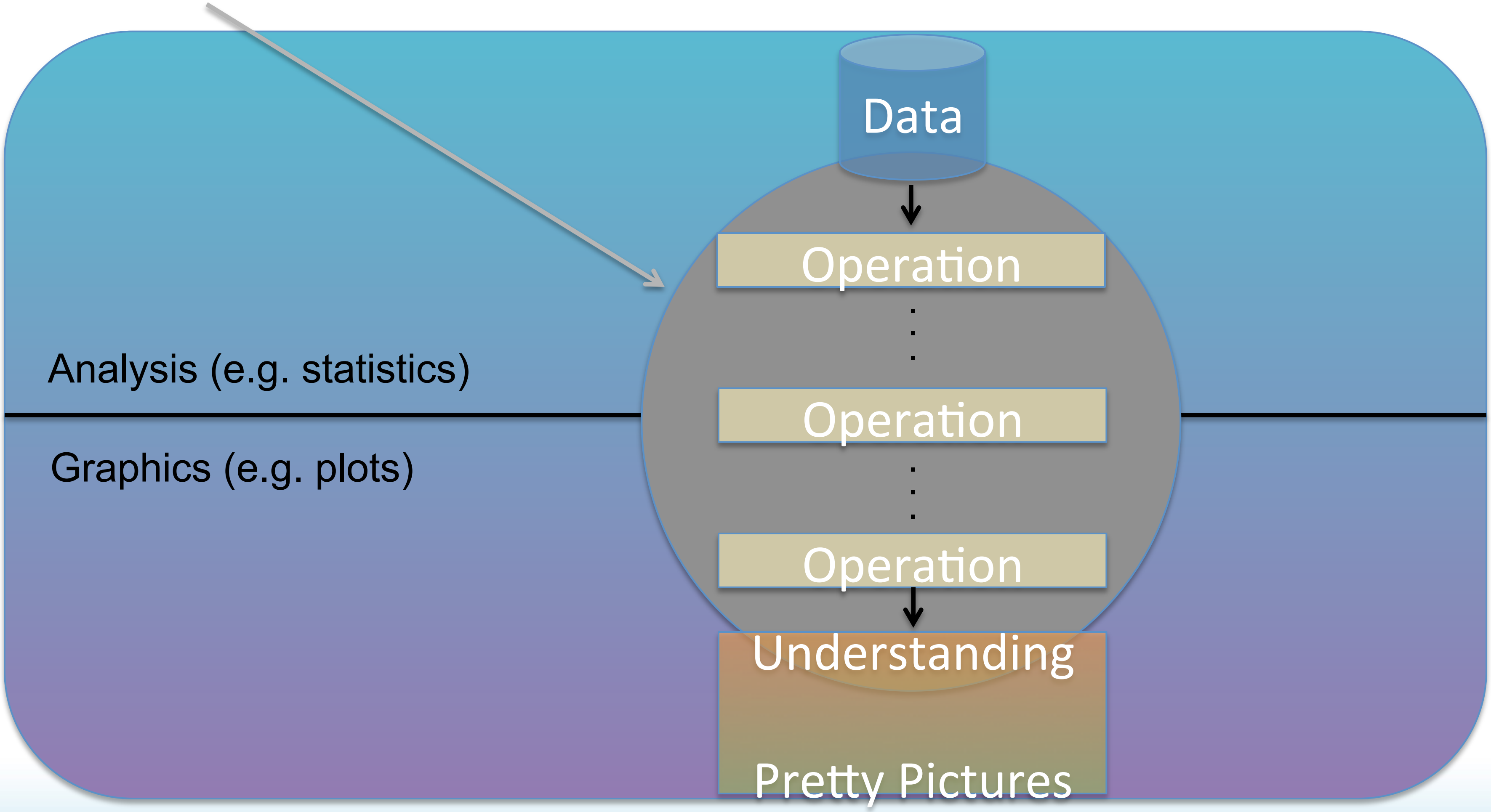
Rob Sisneros

GREAT LAKES CONSORTIUM FOR PETASCALE COMPUTATION

This Talk

- The Basics
  - Visualization support
  - Software
  - People
- Software Capability
- Examples
  - Typical challenges/workload
  - Outreach

Visualization:



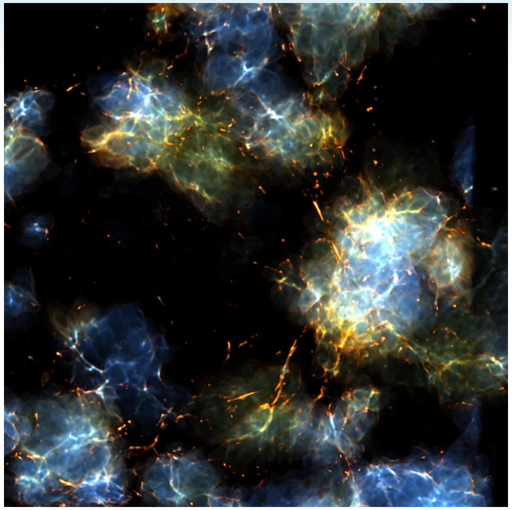
The Blue Waters Visualization and Analysis Team

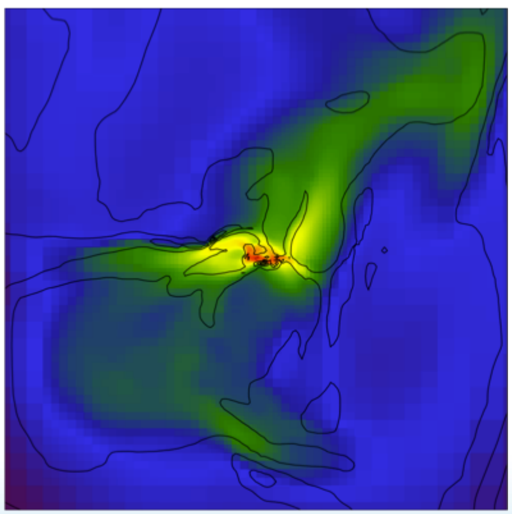
- Software support
  - Installation + maintenance
  - Data preparation
  - Usage
- Research
  - Is this in my data?
  - This is complex, can I show it?
  - Vis for HPC
- Getting your data out there

Supported Software

- Specialized
  - yt
- General, scalable, and open source
  - Paraview
  - VisIt*
- Other
  - IDL, imagemagick
  - Yours? Ask!
    - May decide to officially support
    - Get running *somewhere*

*staff favorite





yt

- Developed to analyze Astrophysics data (Enzo)
- Developed in Python, uses NumPy, Matplotlib, MPI4PY
- Typical analysis
  - Write scripts to derive values
    - Find Halos
    - Create plots
  - Run in batch
- Has in situ support

VisIt

Paraview

- Scalable
  - Scaled > 100K cores
- Offer interactive client/server mode
- Can operate in batch mode
- In situ support
- Rich set of data operators
- Native support for many file formats

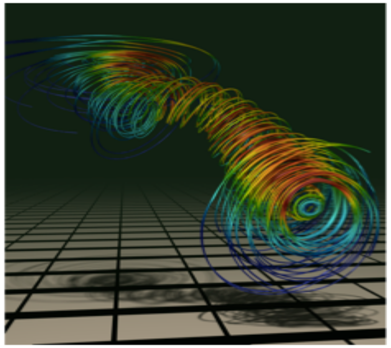
The People

- Dave Bock (20%): Outreach
- Mark Van Moer (20%): ParaView
- Matt Turk (0%): yt
- Rob Sisneros (100%): VisIt, everything else

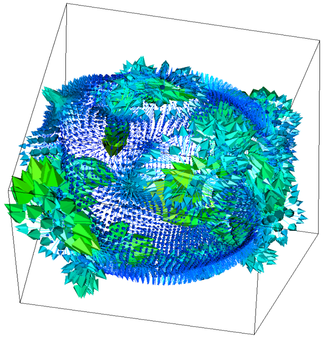
Fun With VisIt

CAPABILITY

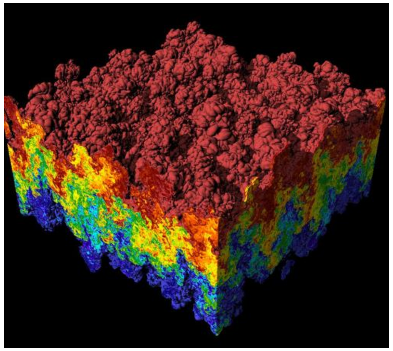
Examples of VisIt's Visualization Capabilities



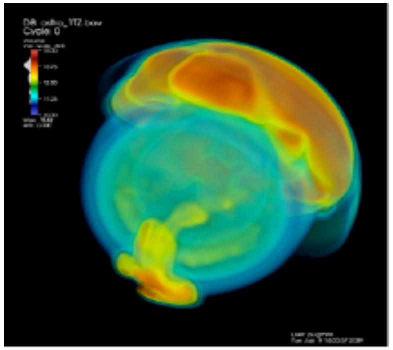
Streamlines



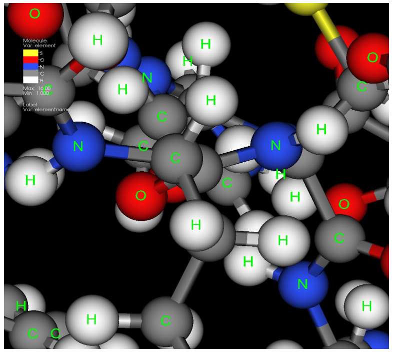
Vector / Tensor Glyphs



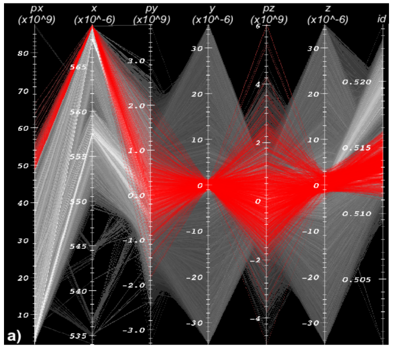
Pseudocolor Rendering



Volume Rendering



Molecular Visualization



Parallel Coordinates

Operators (enabled by default)

Analysis
- ConnectedComponents
- DataBinning
- DeferExpression
- FTLE
- Flux
- Lineout
- PersistentParticles

Debugging
- Inverse Ghost Zone

Geometry
- CoordSwap
- Dual Mesh
- Edge
- External Surface
- Extrude
- Resample
- Revolve
- Smooth
- SurfaceNormal
- Tube

Molecular
- CreateBonds

Selection
- Box
- Clip
- Cylinder
- Index Select
- Isovolume
- MultiresControl
- Onion Peel
- Threshold

Slicing
- Boundary
- Cone
- Isosurface
- Slice
- Spherical Slice
- ThreeSlice

Transforms
- Cartographic Projection
- Displace
- Elevate
- FFT
- Project
- Reflect
- Replicate
- Stagger
- Transform

Other
- ModelFit
- Lagrangian

And… (wouldn't fit)

Geometry

- Triangulate Regular Points

Honorable Mention

Analysis
- ConnectedComponents
- DataBinning
- DeferExpression
- FTLE
- Flux
- Lineout
- PersistentParticles

Debugging
- Inverse Ghost Zone

Geometry
- CoordSwap
- Dual Mesh
- Edge
- External Surface
- Extrude
- Resample
- Revolve
- Smooth
- SurfaceNormal
- Tube

Molecular
- CreateBonds

Selection
- Box
- Clip
- Cylinder
- Index Select
- Isovolume
- MultiresControl
- Onion Peel
- Threshold

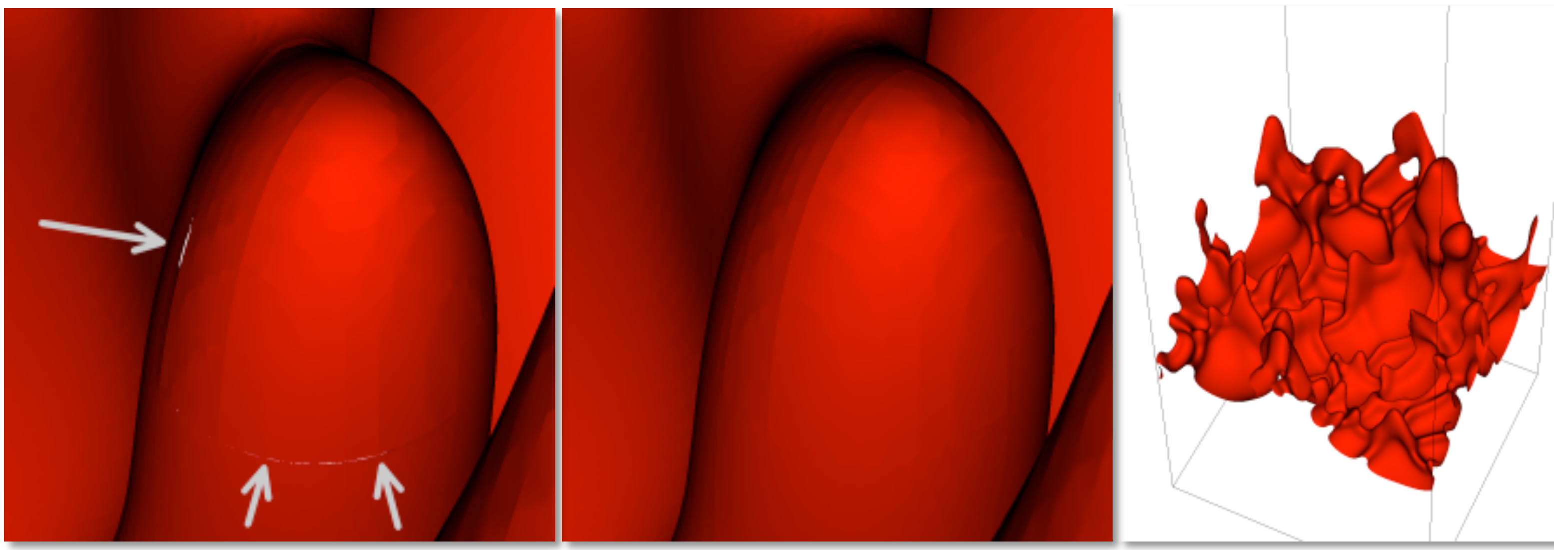
Slicing
- Boundary
- Cone
- Isosurface
- Slice
- Spherical Slice
- ThreeSlice

Transforms
- Cartographic Projection
- Displace
- Elevate
- FFT
- Project
- Reflect
- Replicate
- Stagger
- Transform

Other
- ModelFit
- Lagrangian

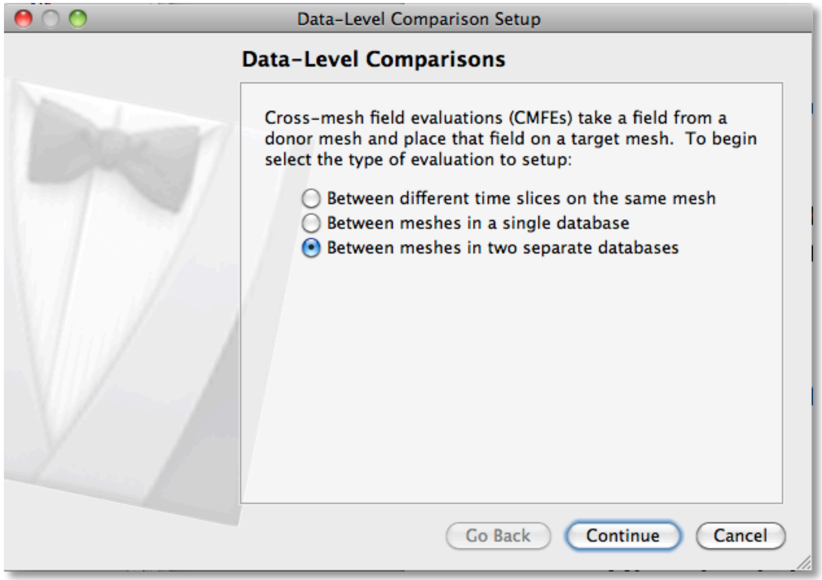
Isosurface

Efficient Parallel Extraction of Crack-free Isosurfaces from AMR Data



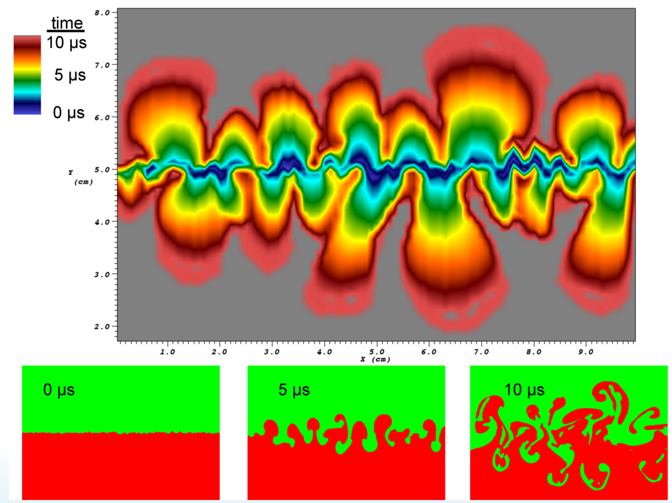
Mesh Comparisons

- Data-level comparison wizard
  - Simplifies comparison of data on different meshes



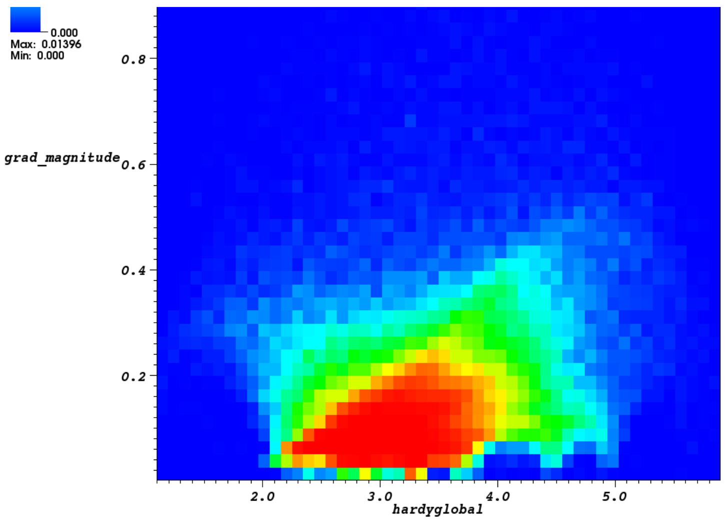
Time Expressions

- New expressions for time iteration
  - Calculate the maximum, minimum or average value at a location
  - Calculate the time when a condition occurs (e.g. time when the maximum value at a location occurs)



Data Binning

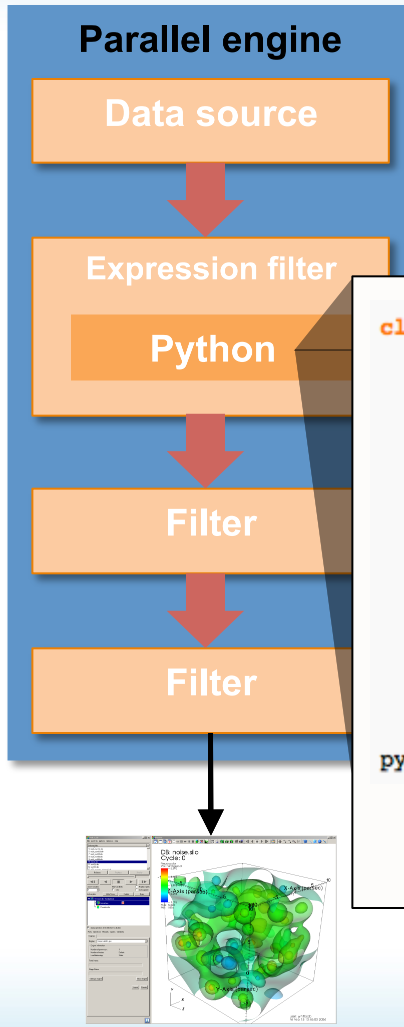
DataBinning operator lets you relate multiple variables and bin them up to compute various quantities per bin



Sum	Standard deviation
Count	Min
PDF	Max
Average	RMS
Variance	

Python Filters

- Write new expressions and queries in Python
- Enter Python expressions into the Expression window's Python tab
- Manipulate VTK data objects directly



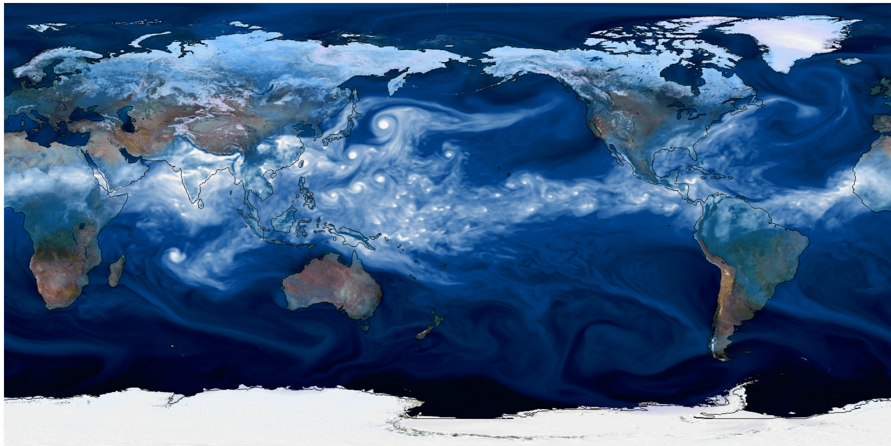
User-defined Python expression

```
class DomainIdExpr(SimplePythonExpression):
    def __init__(self):
        SimplePythonExpression.__init__(self)
        self.name = "DomainIdExpr"
        self.description = "Labeling zones wit
        self.output_is_point_var = False
        self.output_dimension = 1
    def derive_variable(self,ds_in,domain_id):
        ncells = ds_in.GetNumberOfCells()
        res = vtk.vtkFloatArray()
        res.SetNumberOfComponents(1)
        res.SetNumberOfTuples(ncells)
        for i in xrange(ncells):
            res.SetTuple1(i,float(domain_id))
        return res

py_filter = DomainIdExpr
```

Advanced Statistical Analysis

- Inspired by Climate data
  - Statistics over time
  - R is the de facto standard

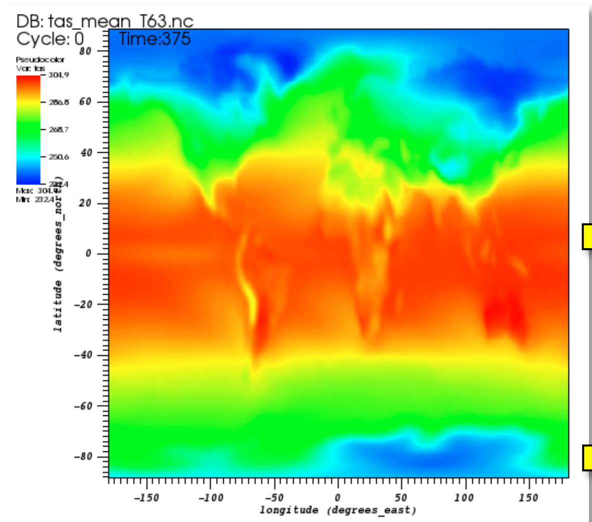


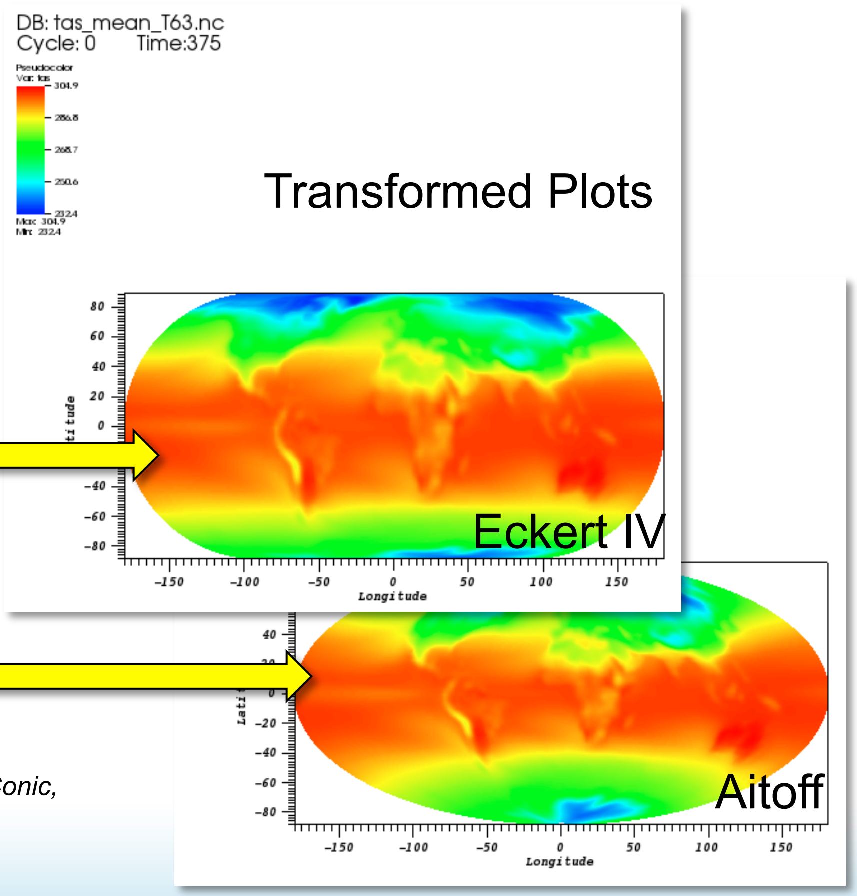
- Additions to VisIt:
  - "Parallelize over time" infrastructure
  - Support for scripting in R

Cartographic Projection Operator

- Transform climate meshes using popular map projections
- Contributed by Jean Favre, CSCS

Original Plot

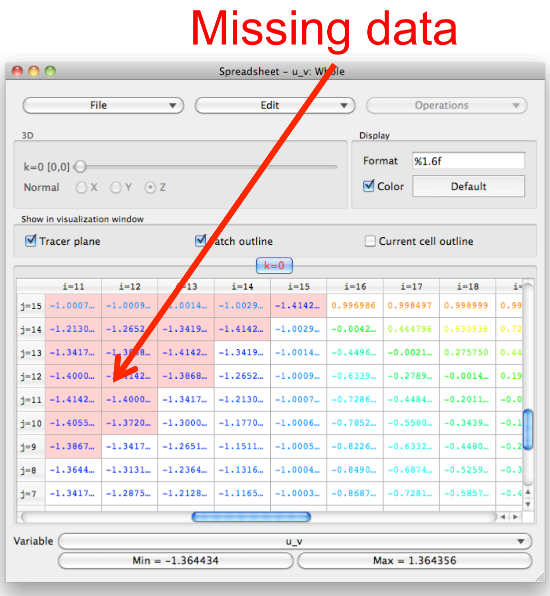


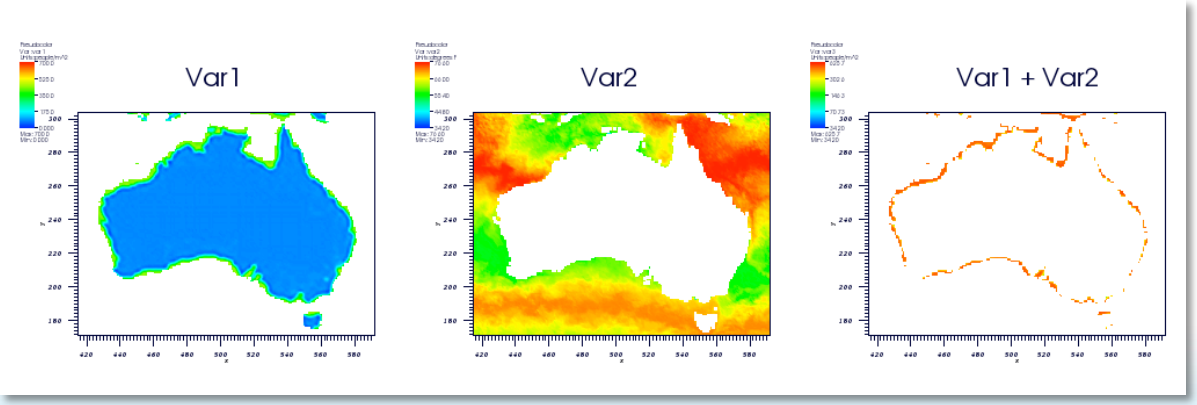


Currently Supported Projections: *Aitoff, Eckert IV, Equidistant Conic, Hammer, Lambert Azimuthal Equal Area, Lambert Conformal Conic, Mercator, Miller Cylindrical, Mollweide, Orthographic, Winkel II*

Missing Data

- Indicate parts of the data have no real values and should be ignored
- Implements NETCDF conventions for missing data *(_FillValue, missing_value, valid_min, valid_max, valid_range)*
- Automatically removed from plots
- Shown specially in Spreadsheet plot
- Taken into account when handling multiple fields, each which may have missing data



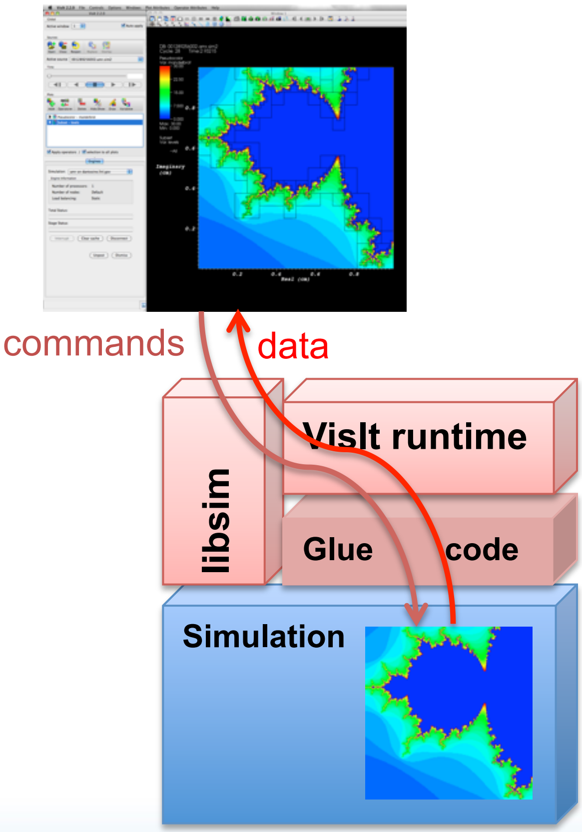


In Situ Visualization with Libsim

- Add functions to your simulation that let VisIt connect
- Add functions to your simulation that expose your arrays as data VisIt will process
- Link your simulation with libsim

- Run the simulation and connect with VisIt
  - You will be able to perform any of VisIt's operations on your simulation data
  - Advance the simulation and watch your plots update

- libsim features
  - Species
  - Vector,Tensor data
  - AMR meshes
  - CSG meshes
  - Users don't allocate memory
  - Additional error checking
  - Write in C, Fortran, or Python
  - Windows support

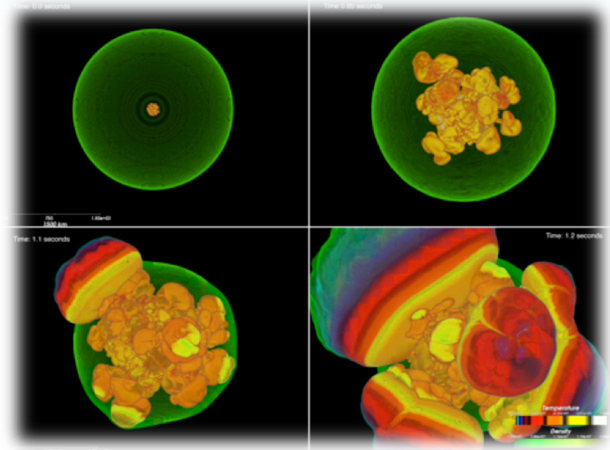


A Slideshow of Some of VisIt's

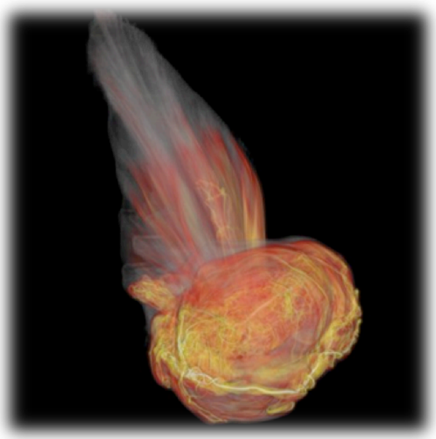
SUPPORTED DATA

Application Code Formats

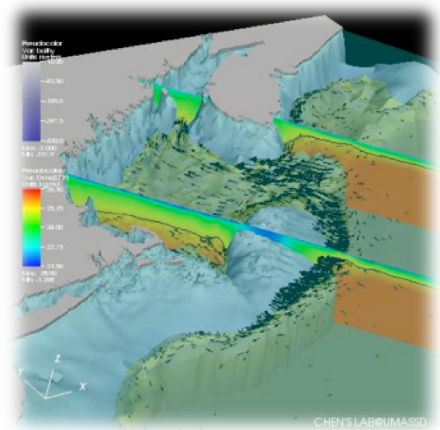
- ANSYS
- Cale
- CASTRO
- CCM
- DDCMD
- Dyna3D
- Enzo
- FLASH
- FVCOM
- Gadget
- LAMMPS
- NASTRAN
- Nek5000
- OVERFLOW
- PATRAN
- Pixie
- S3D
- ZeusMP



FLASH



CASTRO

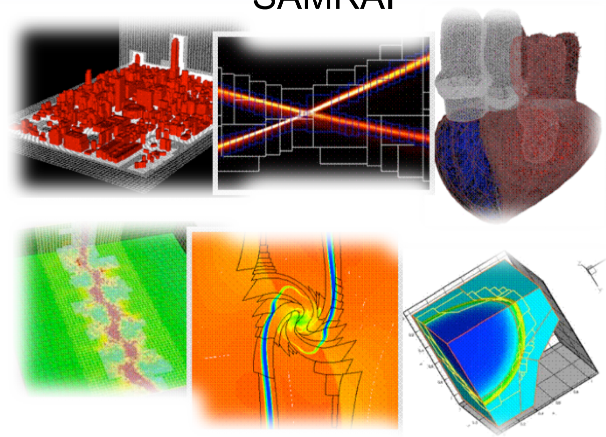


FVCOM

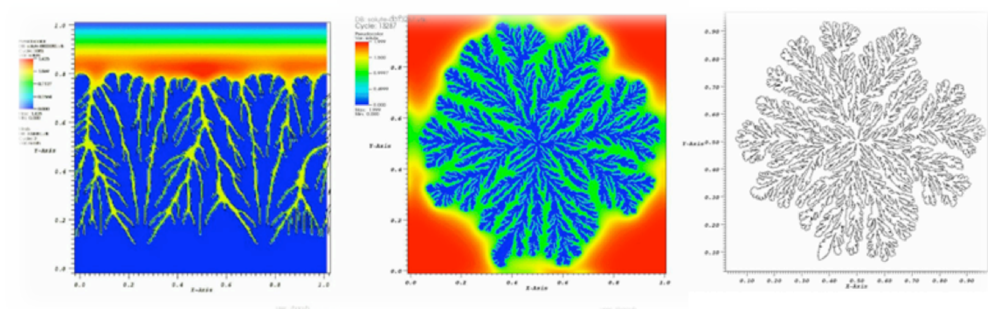
Application Toolkit Formats

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

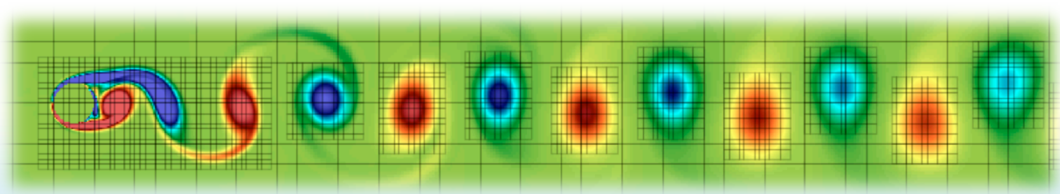
SAMRAI



ITAPS



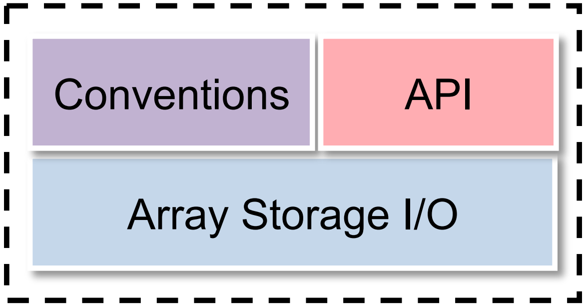
Chombo



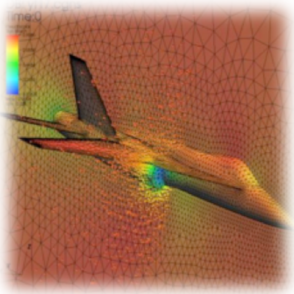
General Scientific Data Formats

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

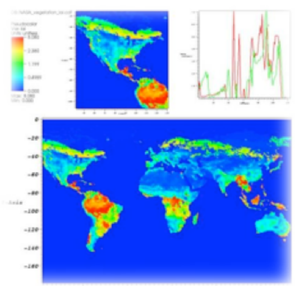
Common Structure



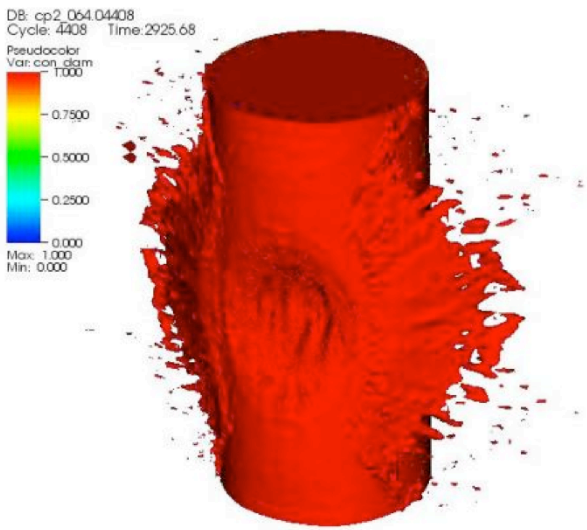
CGNS



NETCDF



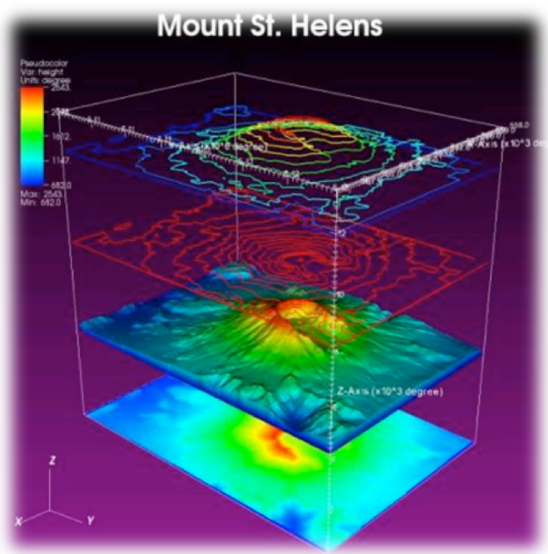
Silo / Ale3d

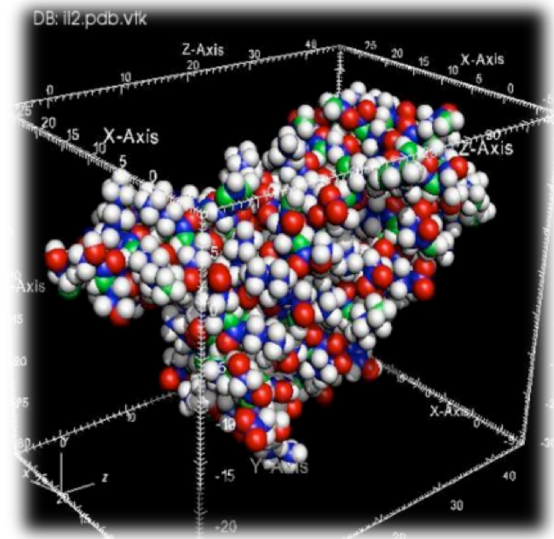


Specialized Scientific Data Formats

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

DEM from GDAL

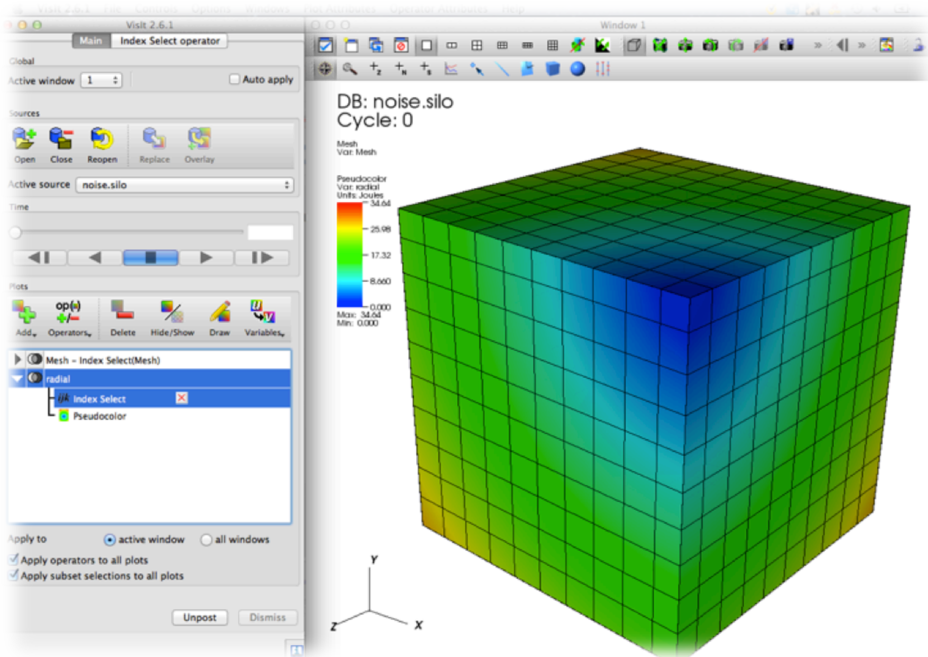




Protein Data Bank

Visualization Formats

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



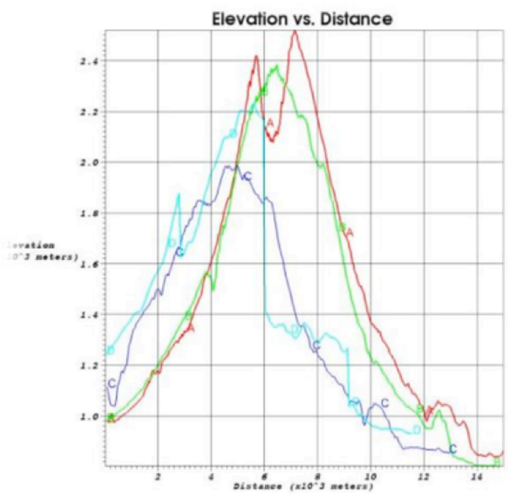
Graphics Formats

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

Carina Nebula

General ASCII Data Formats

- Curve2D
- Lines
- PlainText
- Point3D

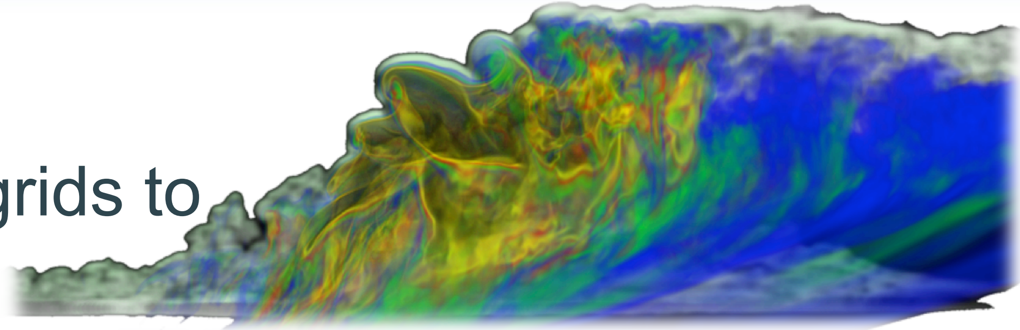


	i=0	i=1	i=2	i=3	i=4	i=5
j=7	2.517243	2.550414	2.581495	2.609803	2.634335	2.653569
j=6	2.472034	2.503052	2.531701	2.557125	2.578064	2.592629
j=5	2.427398	2.456259	2.482482	2.505081	2.522616	2.532976
j=4	2.383583	2.410415	2.434426	2.454567	2.469347	2.476640
j=3	2.340819	2.365857	2.388012	2.406262	2.419193	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

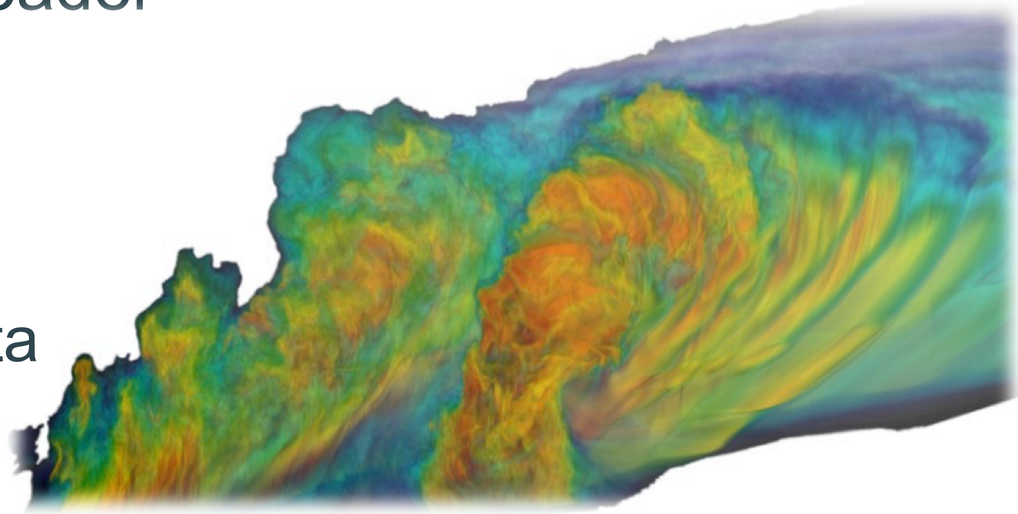
```
2.72727272727 7.57851239669
2.77777777778 7.78703703704
2.82828282828 7.99938781757
2.87878787879 8.21556473829
2.92929292929 8.4355677992
2.9797979798 8.65939700031
3.0303030303 8.8870523416
3.08080808081 9.11853382308
3.13131313131 9.35384144475
3.18181818182 9.59297520661
3.23232323232 9.83593510866
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
```

Blue Waters Task 1:

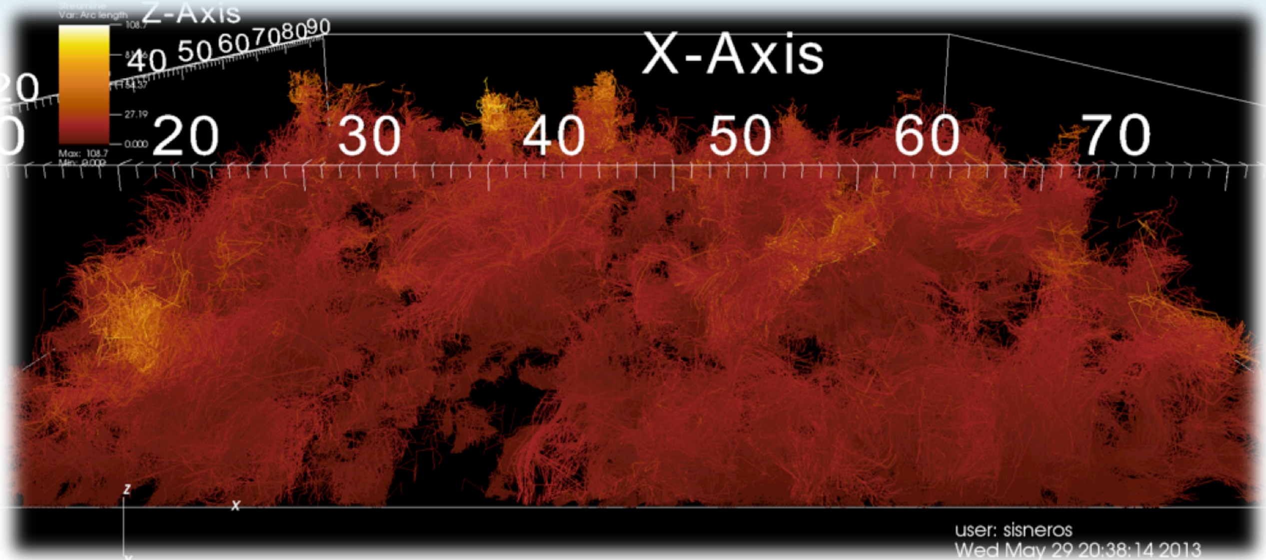
- "Getting Data into VisIt"
- Added support for AMR grids to VisIt's Enzo data reader



- New parallel CM1 data reader
  - I/O library (Leigh Orf) applied to CM1 to aggregate writes – nice performance
  - Interesting chunks of data no longer readable by VisIt

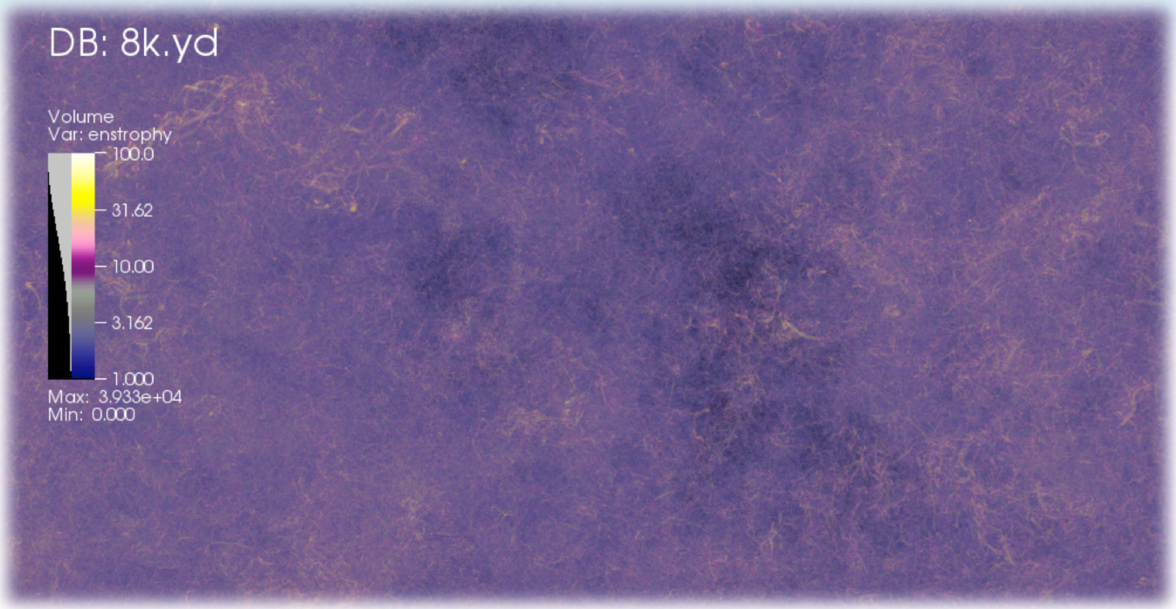


New Reader 2: Binary Support



- Current support
  - Single variable
  - Single timestep
- Range of Block files (.rob)
  - Any number of variables
  - Any number of timesteps
  - Any ordering these and spatial dimensions

New Reader 3: Data Rearrange



- Current data
  - 2D slabs
  - Many files
  - Creating ghost zones more than doubles data size
- New files
  - Read files of any dimension
  - Dynamic resize from file dimensions to nice vis-shaped 3D blocks
  - Number of blocks need not match number of files

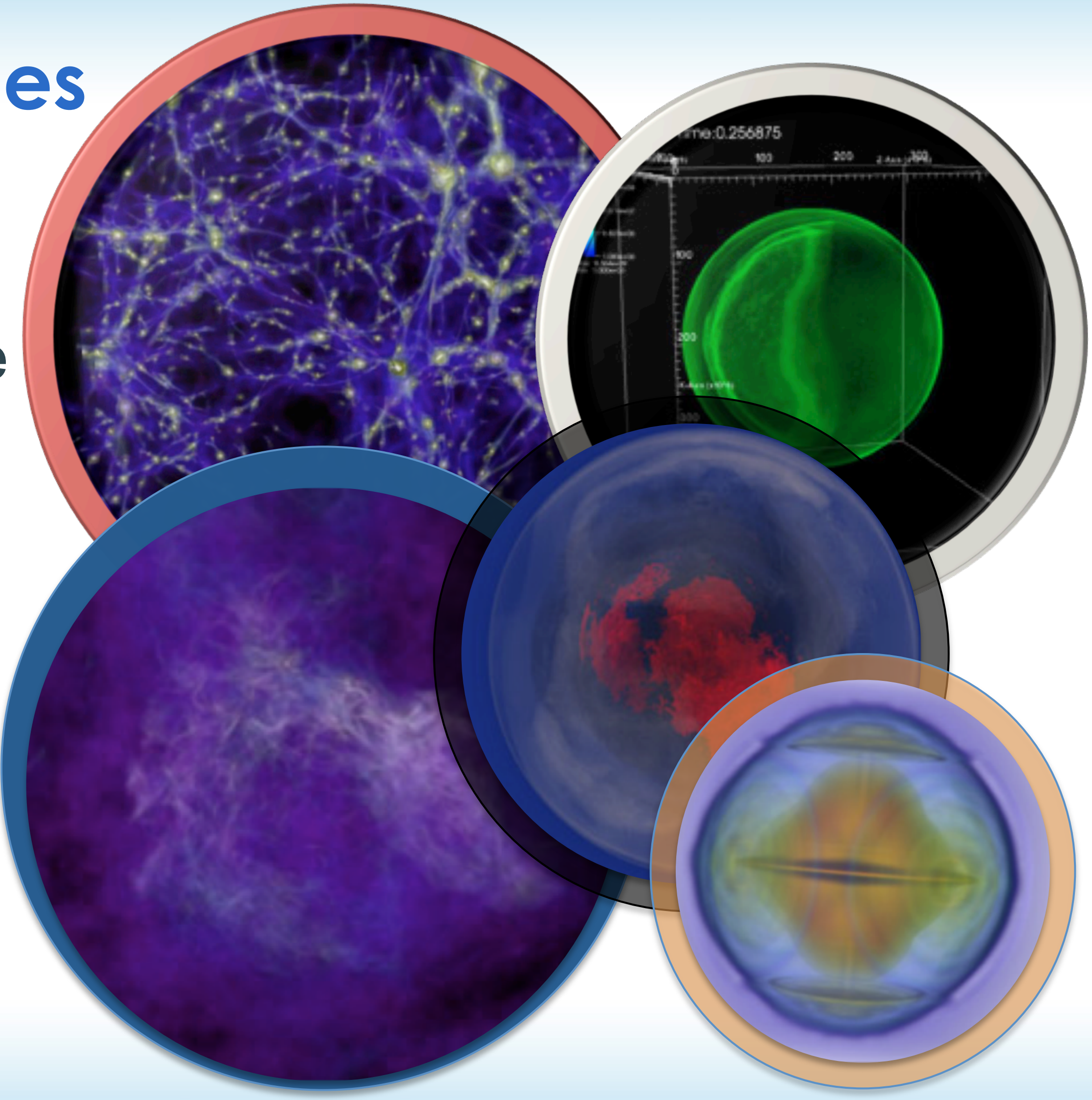
VisIt As-Is:

OUTREACH

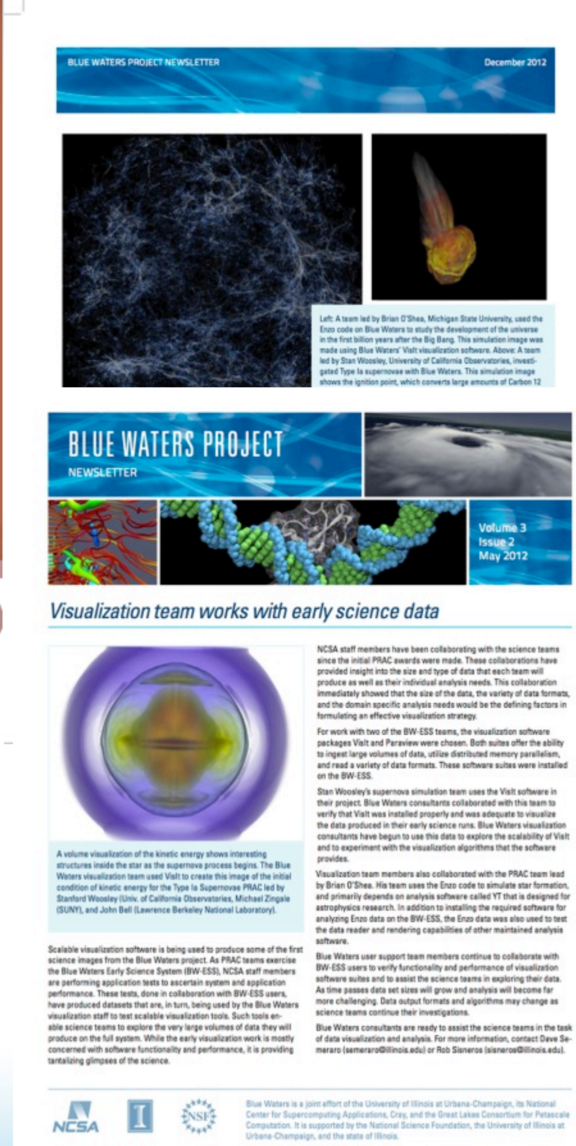
NCSA Online Articles

“Blue Waters visualization team provides first science images”

“Simulating supernovae with Blue Waters”



Newsletters
Posters
Videos



BLUE WATERS PROJECT NEWSLETTER December 2012

Left: A team led by Brian O'Shea, Michigan State University, used the Enzo code on Blue Waters to study the development of the universe in the first billion years after the Big Bang. This simulation image was made using Blue Waters' VisIt visualization software. Above: A team led by Stan Woosley, University of California Observatories, investigated Type Ia supernovae with Blue Waters. This simulation image shows the ignition point, which converts large amounts of Carbon 12

BLUE WATERS PROJECT
NEWSLETTER

Volume 3
Issue 2
May 2012

Visualization team works with early science data

A volume visualization of the kinetic energy shows interesting structures inside the star as the supernova process begins. The Blue Waters visualization team used VisIt to create this image of the initial condition of kinetic energy for the Type Ia Supernovae PRAC led by Stanford Woosley (Univ. of California Observatories, Michael Zingale (SUNY), and John Bell (Lawrence Berkeley National Laboratory).

Scalable visualization software is being used to produce some of the first science images from the Blue Waters project. As PRAC teams exercise the Blue Waters Early Science System (BW-ESS), NCSA staff members are performing application tests to ascertain system and application performance. These tests, done in collaboration with BW-ESS users, have produced datasets that are, in turn, being used by the Blue Waters visualization staff to test scalable visualization tools. Such tools enable science teams to explore the very large volumes of data they will produce on the full system. While the early visualization work is mostly concerned with software functionality and performance, it is providing tantalizing glimpses of the science.

NCSA staff members have been collaborating with the science teams since the initial PRAC awards were made. These collaborations have provided insight into the size and type of data that each team will produce as well as their individual analysis needs. This collaboration immediately showed that the size of the data, the variety of data formats, and the domain specific analysis needs would be the defining factors in formulating an effective visualization strategy.

For work with two of the BW-ESS teams, the visualization software packages VisIt and Paraview were chosen. Both suites offer the ability to ingest large volumes of data, utilize distributed memory parallelism, and read a variety of data formats. These software suites were installed on the BW-ESS.

Stan Woosley's supernova simulation team uses the VisIt software in their project. Blue Waters consultants collaborated with this team to verify that VisIt was installed properly and was adequate to visualize the data produced in their early science runs. Blue Waters visualization consultants have begun to use this data to explore the scalability of VisIt and to experiment with the visualization algorithms that the software provides.

Visualization team members also collaborated with the PRAC team lead by Brian O'Shea. His team uses the Enzo code to simulate star formation, and primarily depends on analysis software called YT that is designed for astrophysics research. In addition to installing the required software for analyzing Enzo data on the BW-ESS, the Enzo data was also used to test the data reader and rendering capabilities of other maintained analysis software.

Blue Waters user support team members continue to collaborate with BW-ESS users to verify functionality and performance of visualization software suites and to assist the science teams in exploring their data. As time passes data set sizes will grow and analysis will become far more challenging. Data output formats and algorithms may change as science teams continue their investigations.

Blue Waters consultants are ready to assist the science teams in the task of data visualization and analysis. For more information, contact Dave Semeraro (semeraro@illinois.edu) or Rob Sisneros (sisneros@illinois.edu).

Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign, its National Center for Supercomputing Applications, Cray, and the Great Lakes Consortium for Petascale Computation. It is supported by the National Science Foundation, the University of Illinois at Urbana-Champaign, and the state of Illinois.

And…

DRAWING NUMBERS

Access' Barbara Jewett visits with Dave Semeraro, leader of NCSA's Advanced Digital Services visualization team, to learn how his team enables research.

Q. HOW WOULD YOU DESCRIBE VISUALIZATION TO SOMEONE WHO IS NOT FAMILIAR WITH IT?

A. You need to understand that a computer just generates numbers. That's all the computer knows how to do. Visualization turns the numbers the computer generates into something visual we can more easily relate to. Humans are primarily visual interpreters of information. Most of our cognition comes from our visual perception of the world. When you are simulating something physical on the computer, like the climate or the formation of a new star in a galaxy, it's meaningful to look at that from a visual standpoint rather than a table of numbers. So if I can take a representation of the globe and color the surface of the globe based on temperature, you can see over time how the average temperature in different regions of the globe changes, based on, for example, added greenhouse gases. You could graph those numbers, but it's more engaging and insightful to look at them in a visual sense.

Q. ARE THERE DIFFERENT TYPES OF VISUALIZATION?

A. The type of visualization that we do at NCSA is what is called scientific visualization. There's another discipline of visualization called information visualization. Those are the sorts of visualizations that you might see on the internet or in a magazine or newspaper. A good example would be an infographic. A classic infographic is the pie chart, such as one that shows where your income tax dollars go—which slice of the pie goes to defense, which slice of the pie goes to education, and so on. Here at NCSA, researchers we assist frequently have observational data. For example, water levels of rivers over time or how much precipitation certain areas received. Your local Doppler radar is a very important time-related visualization that's from observed data. We have researchers who are combining Doppler radar data and simulation to better predict severe storms. That's a combination of observed data and simulated data analysis and visualization.

Q. IS VISUALIZATION DIFFICULT FOR SCIENTISTS TO LEARN HOW TO DO?

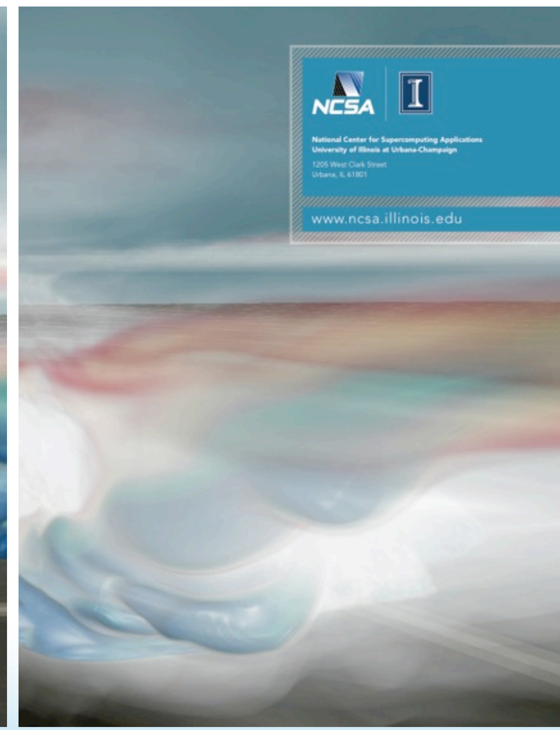
A. No! It's really not. In fact, one of the things that people don't realize is that a scientist, during the course of developing his science and simulation applications, usually develops an entire suite of analysis and visualization tools and techniques that go along with them. One key aspect of that is the analysis part. Visualization without analysis is semi-useless. You could misrepesent the data. You could visualize the cloud surface of a severe storm without actually finding out what's going on inside the cloud. Visualization experts know how to represent that information visually.

Q. IF SCIENTISTS KNOW HOW TO DO THEIR OWN VISUALIZATIONS, WHY DO WE HAVE VISUALIZATION EXPERTS?

A. Well, science teams don't typically employ people with a background in information presentation or visualization representation. They hire grad students in their discipline who are experts in simulating the physics the researcher wants to simulate but usually not knowledgeable in the physics of visualization representation. It's a big enough job to keep up with the physics and mathematics of your field without having to worry about the extra burden of different visualization techniques. That's why visualization consultants exist. You can't be up on all the different visualization packages and still be a good scientist unless you are really, really good or a very artistic scientist.

04

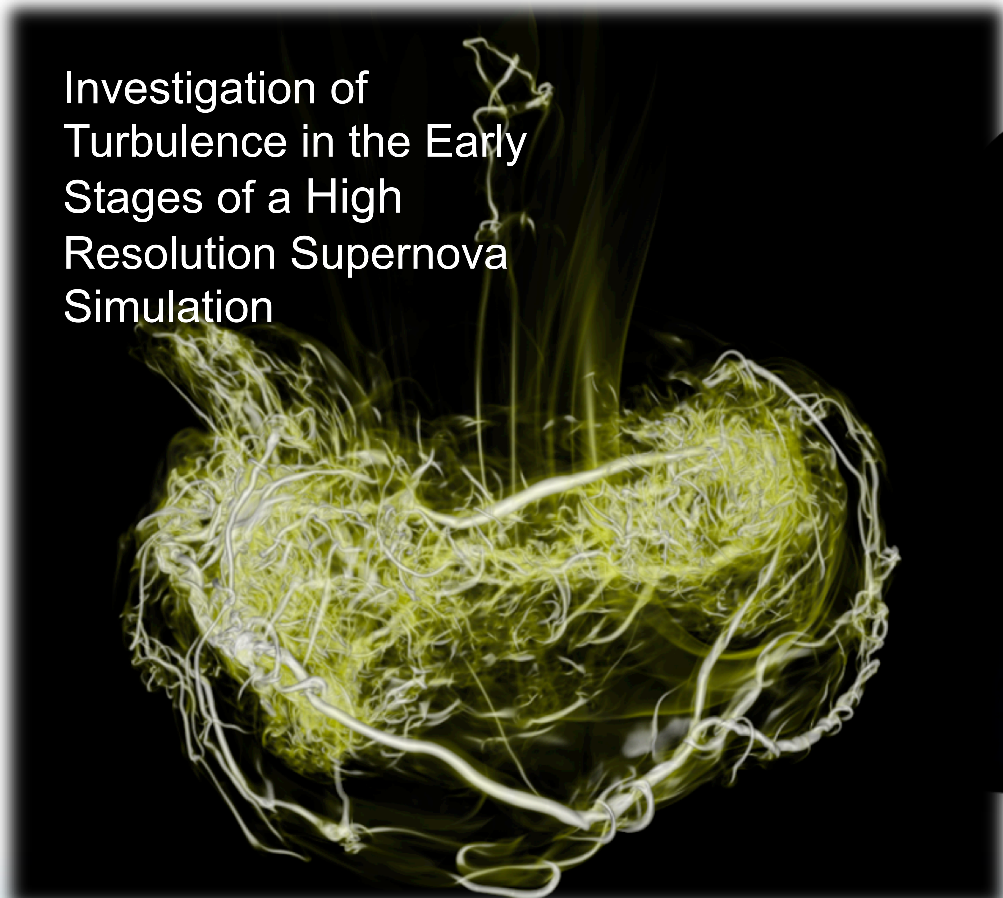
NCSA Access Magazine

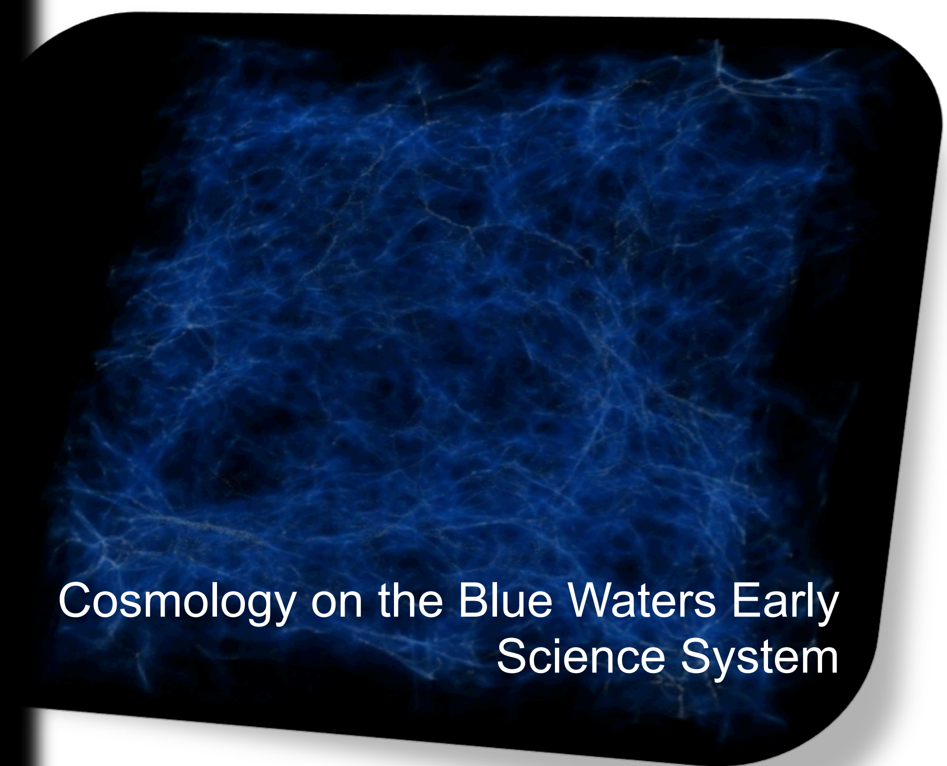


Fall 2012

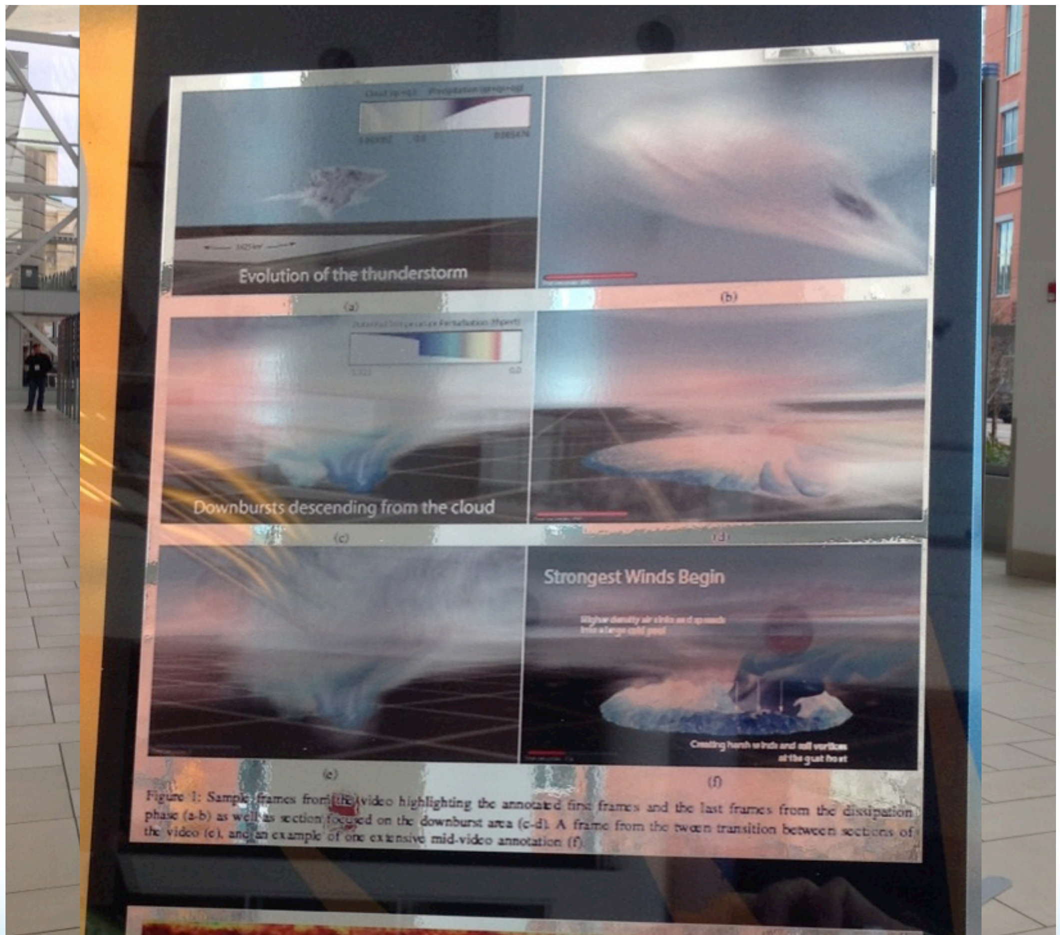
Spring/Summer 2013

SC 12 Scientific Visualization Showcase





SC 13 Scientific Visualization Showcase



Contact Us

- Help+bw to open tickets as usual

- Rob Sisneros (sisneros@illinois.edu)
- Mark Van Moer (mvanmoer@illinois.edu)
- Dave Bock (dbock@illinois.edu)