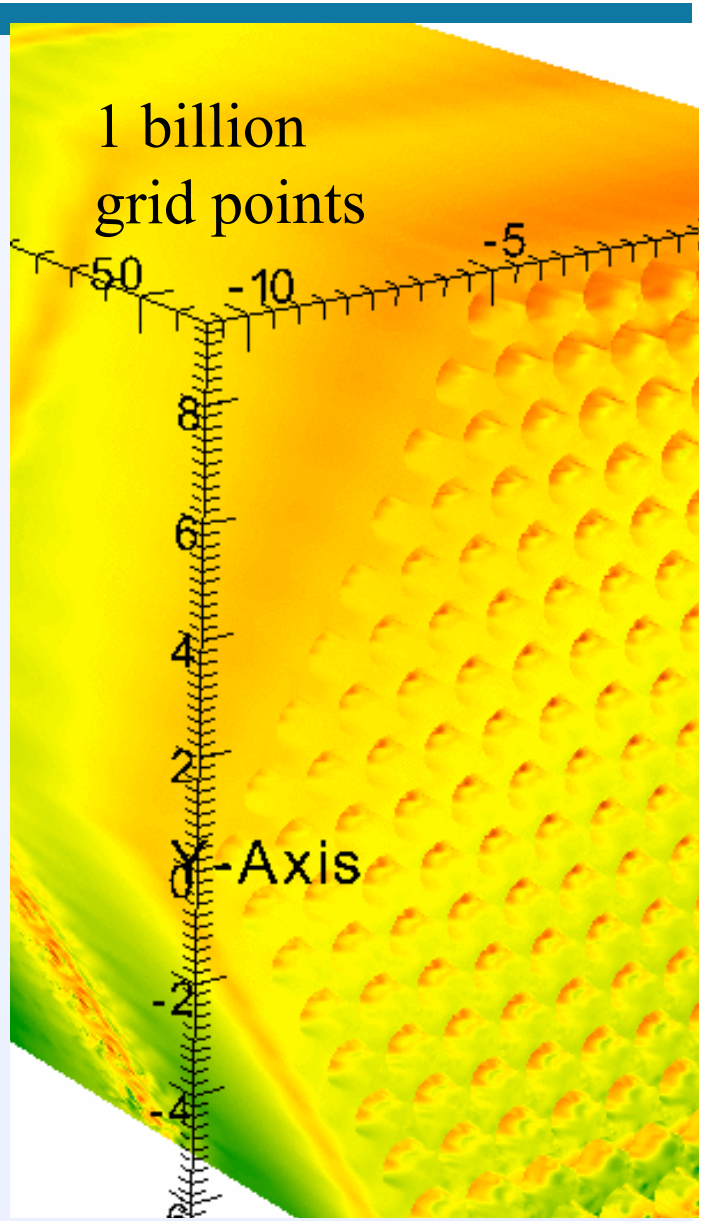
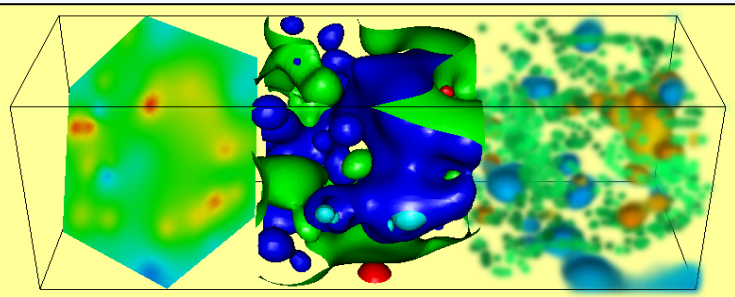
VisIt is a richly featured, turnkey application

- VisIt is an open source, end user visualization and analysis tool for simulated and experimental data
  - Used by: physicists, engineers, code developers, vis experts
  - >100K downloads on web
- R&D 100 award in 2005
- Used on many of the Top 500

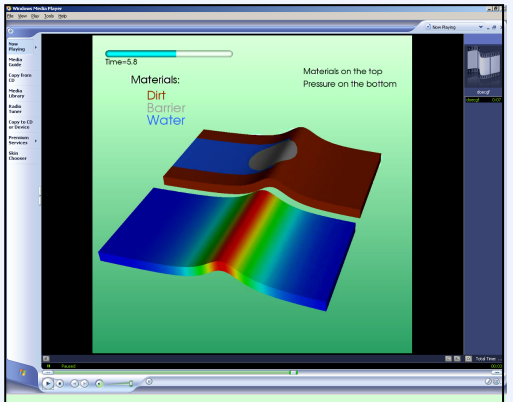


217 pin reactor cooling simulation. Run on ¼ of Argonne BG/P.

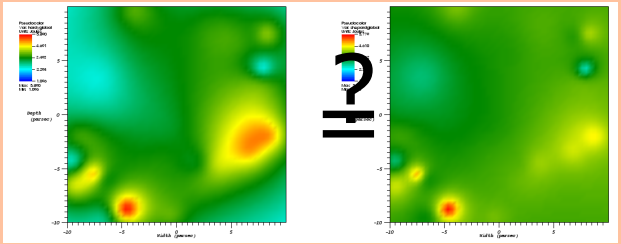
Terribly Named!! Intended for more than just visualization!



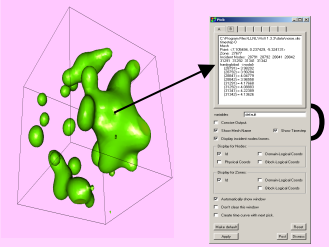
Data Exploration



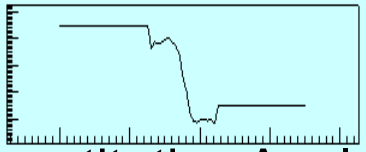
Presentations



Comparative Analysis



Visual Debugging

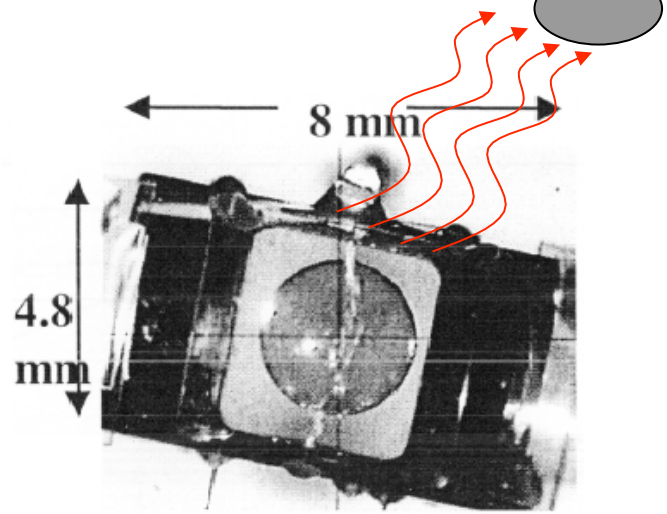


Quantitative Analysis

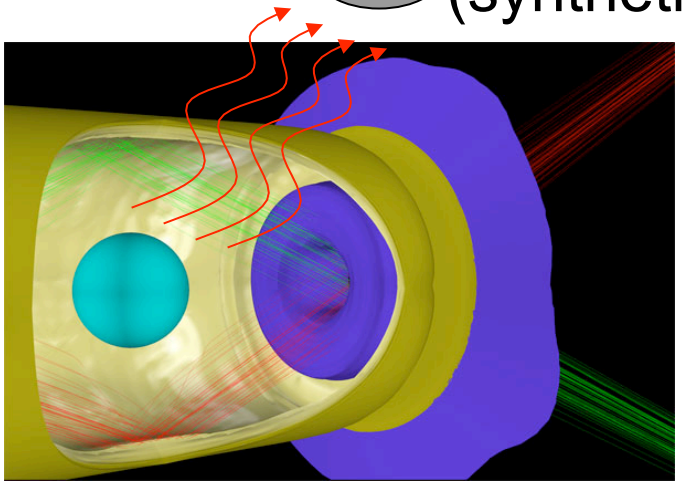
Quantitative analysis means different things to different people.

1) Techniques that span scientific domains (e.g. integration, volumes, surface areas, fluxes, connected components, chord length distributions)
2) Specialized analysis (e.g. hohlraum flux at AGEX)

Detector at AGEX



Detector provided by VisIt (synthetic diagnostic)

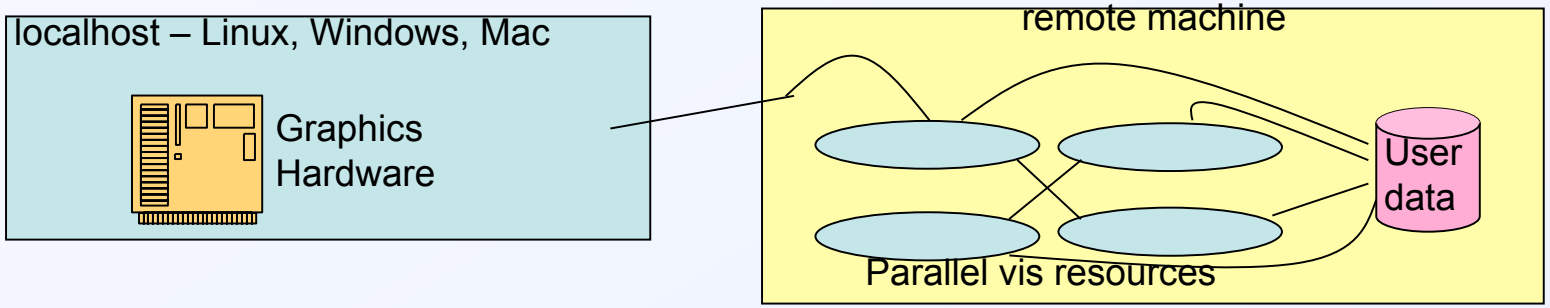


VisIt has a rich feature set.

- Meshes: rectilinear, curvilinear, unstructured, point, AMR
- Data: scalar, vector, tensor, material, species
- Dimension: 1D, 2D, 3D, time varying
- Rendering (~15): pseudocolor, volume rendering, hedgehogs, glyphs, mesh lines, etc…
- Data manipulation (~40): slicing, contouring, clipping, thresholding, restrict to box, reflect, project, revolve, …
- File formats (~85)
- Derived quantities: >100 interoperable building blocks

  +,-,*,/, gradient, mesh quality, if-then-else, and, or, not
- Many general features: position lights, make movie, etc
- Queries (~50): ways to pull out quantitative information, debugging, comparative analysis

VisIt employs a parallelized client-server architecture.



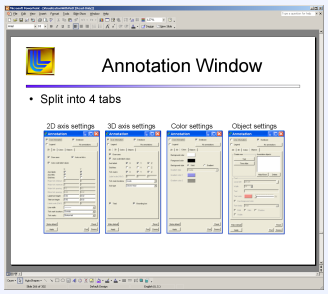
- Client-server observations:
  - Good for remote visualization
  - Leverages available resources
  - Scales well
  - No need to move data
- Additional design considerations:
  - Plugins
  - Heavy use of VTK
  - Multiple UIs: GUI (Qt), CLI (Python), more…

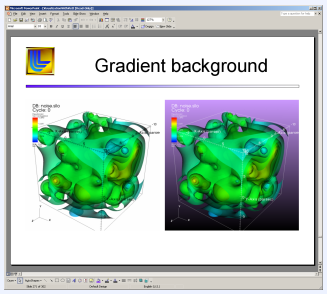
The VisIt team focuses on making a robust, usable product for end users.

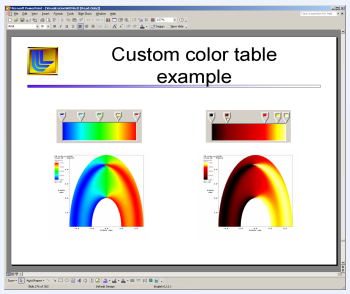
BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY

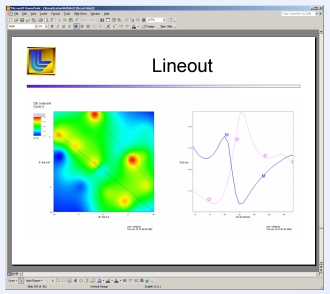
- Manuals
  - 300 page user manual
  - 200 page command line interface manual
  - “Getting your data into VisIt” manual
- Wiki for users (and developers)
- Revision control, nightly regression testing, etc
- Executables for all major platforms
- Day long class, complete with exercises

Slides from the VisIt class



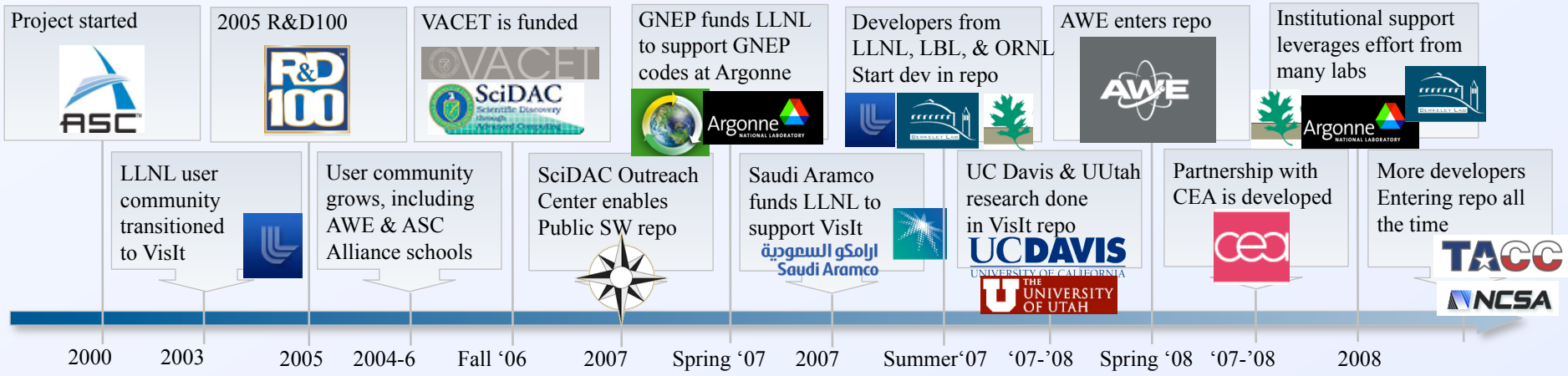






VisIt is a vibrant project with many participants.

- Over 75 person-years of effort
- Over 1.5 million lines of code
- Partnership between: Department of Energy's Office of Nuclear Energy, Office of Science, and National Nuclear Security Agency, and among others



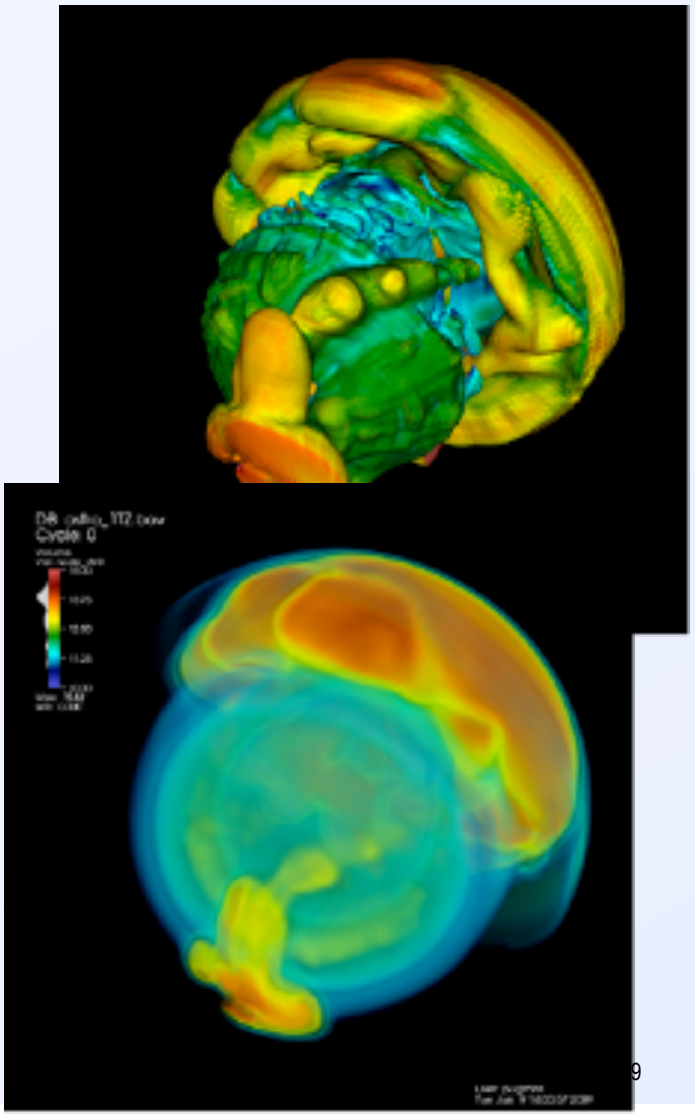
VisIt: What's the Big Deal?

- Everything works at scale
- Robust, usable tool
- Vis to code development to scientific insight
- Healthy future: vibrant developer and user communities

VisIt was recently demonstrated to show good performance at unprecedented scales.

- Weak scaling study:
  ~62.5M cells/core

Machine	Model	Problem Size	#cores
Franklin	Cray XT4	1T, 2T	16K, 32K
Dawn	BG/P	4T	64K
JaguarPF	Cray XT5	2T	32K
Juno	X86_64	1T	16K
Purple	IBM P5	0.5T	8K
Ranger	Sun	1T	16K



Three Ways To Get Data Into VisIt

- (1) Write to a known output format
  - FLASH, ENZO, Nek, Stacks of images, some netcdf
- (2) Write a plugin file format reader
- (3) Integrate VisIt "in situ"
  - "lib-VisIt" is linked into simulation code
    - (Note: Memory footprint issues with implementation!)
  - Use model:
    - simulation code advances
    - at some time interval (e.g. end of cycle), hands control to lib-VisIt.
    - lib-VisIt performs vis & analysis tasks, then hands control back to simulation code
    - repeat

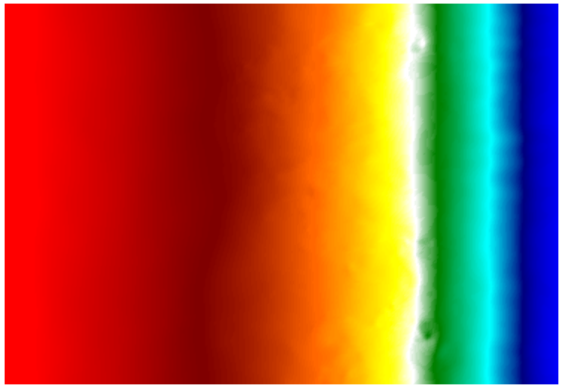
Before we begin…

- Tutorial:
  - Two tutorials coming up @ 

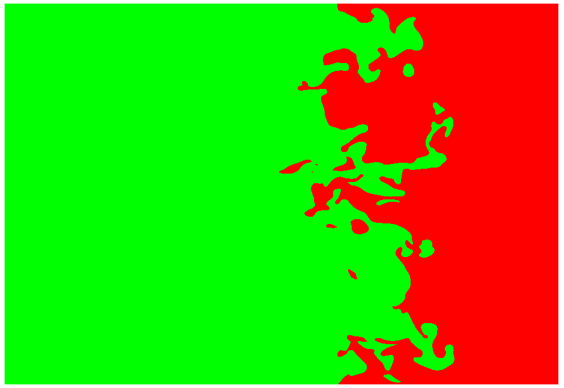
- User resources:
  - Main website: http://www.llnl.gov/visit
  - Wiki: http://www.visitusers.org
  - Email: visitusers@ornl.gov
  - Email: visit-help-scidac@ornl.gov
- Development resources:
  - Email: visit-developers@ornl.gov
  - SVN: http://portal.nersc.gov/svn/visit

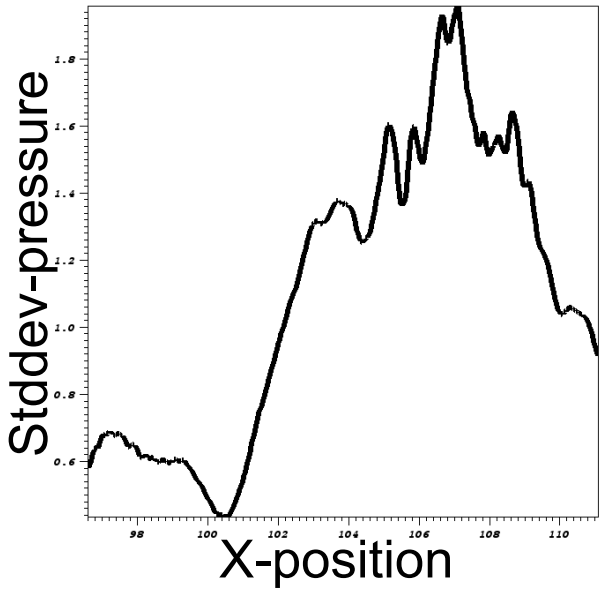
Equivalence Class Functions: Rictmeyer-Meshkov

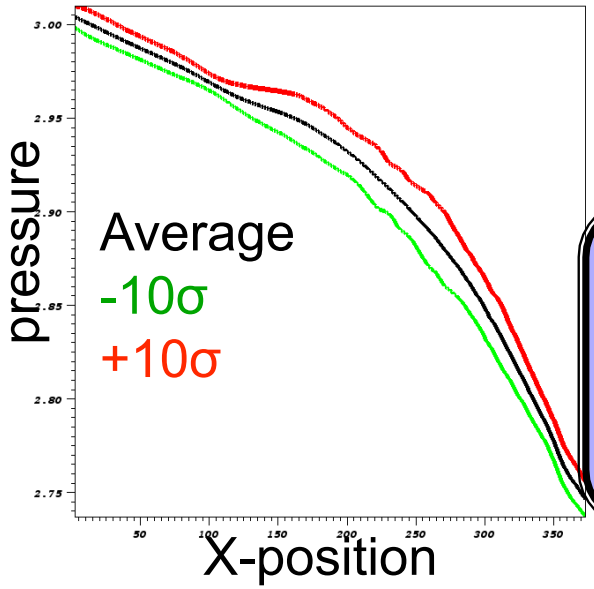
Pressure



Interface between gases







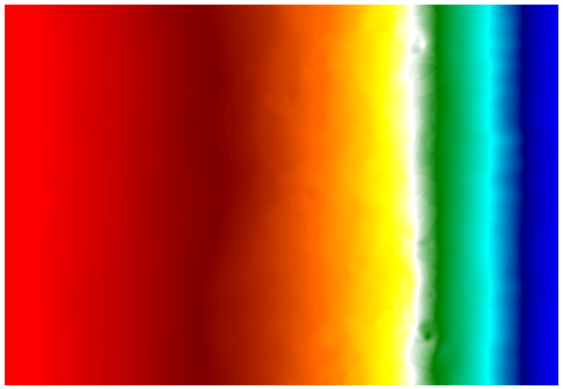
ECF1 =
Average pressure over
Equivalence classes
In X-position

ECF2 =
Stddev pressure over
Equivalence classes
In X-position

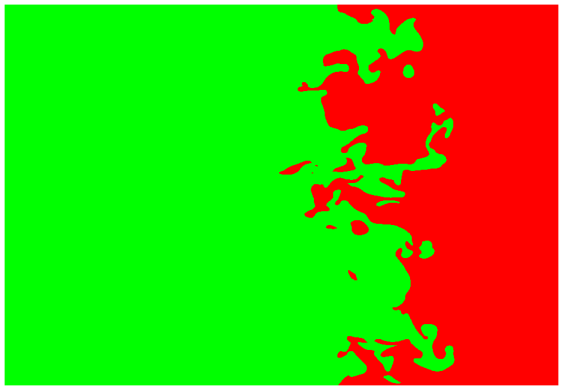
This is analysis-oriented ECF usage.

Equivalence Class Functions: Rictmeyer-Meshkov

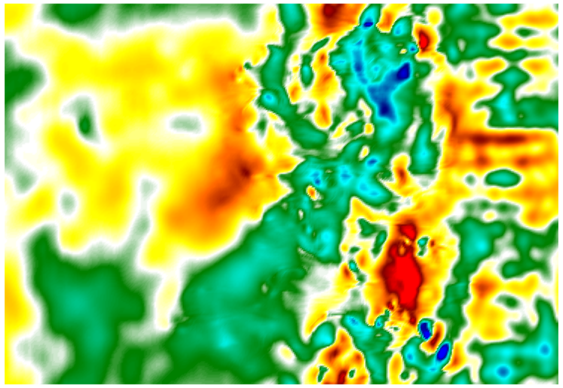
Pressure



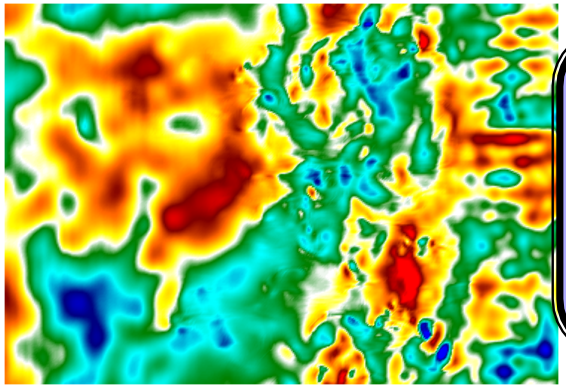
Interface between gases



Pressure – ECF1



(Pressure – ECF1) / ECF2

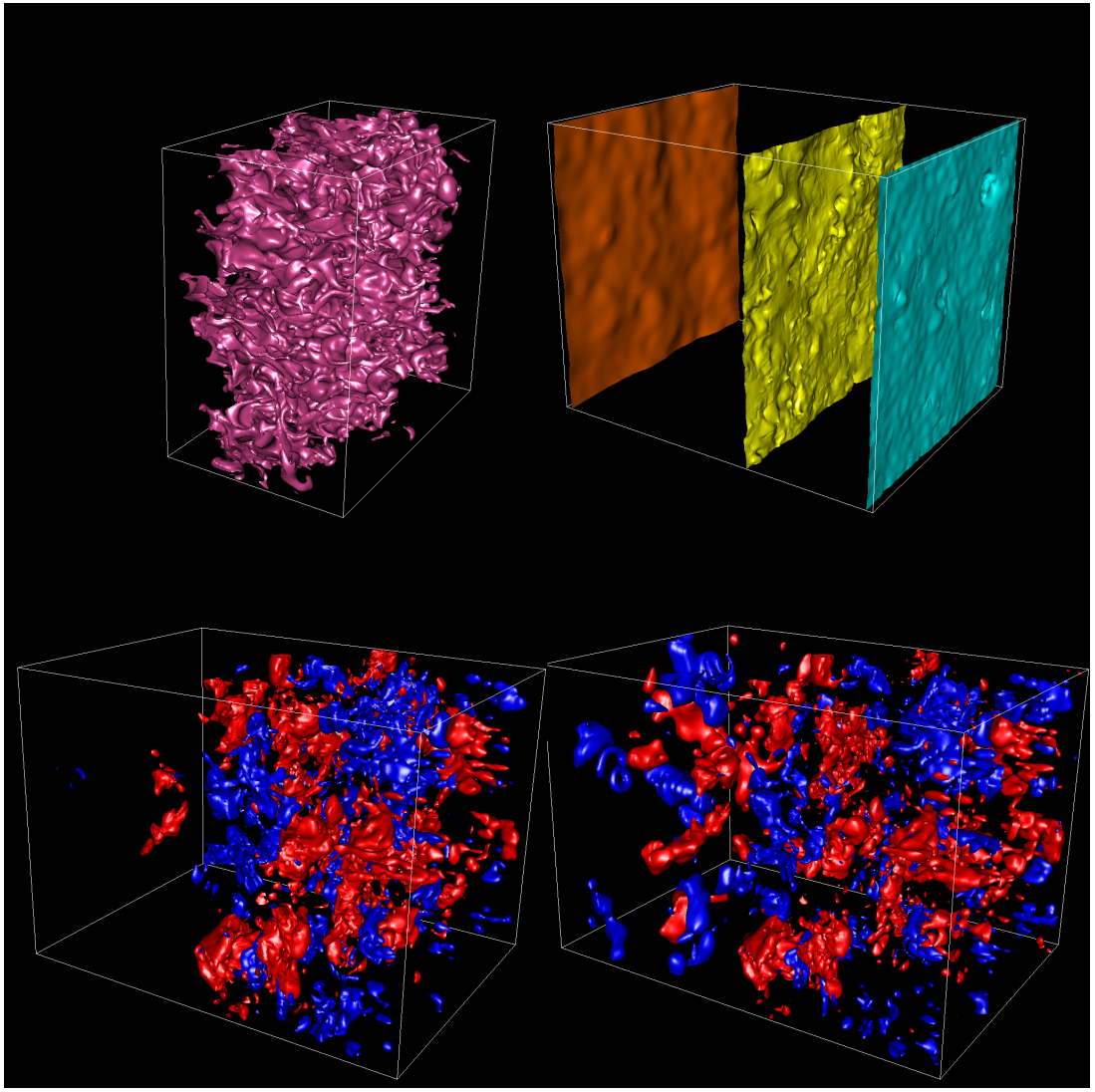


ECF1 = Average pressure over Equivalence classes In X-position

ECF2 = Stddev pressure over Equivalence classes In X-position

This is synthesis-oriented ECF usage.

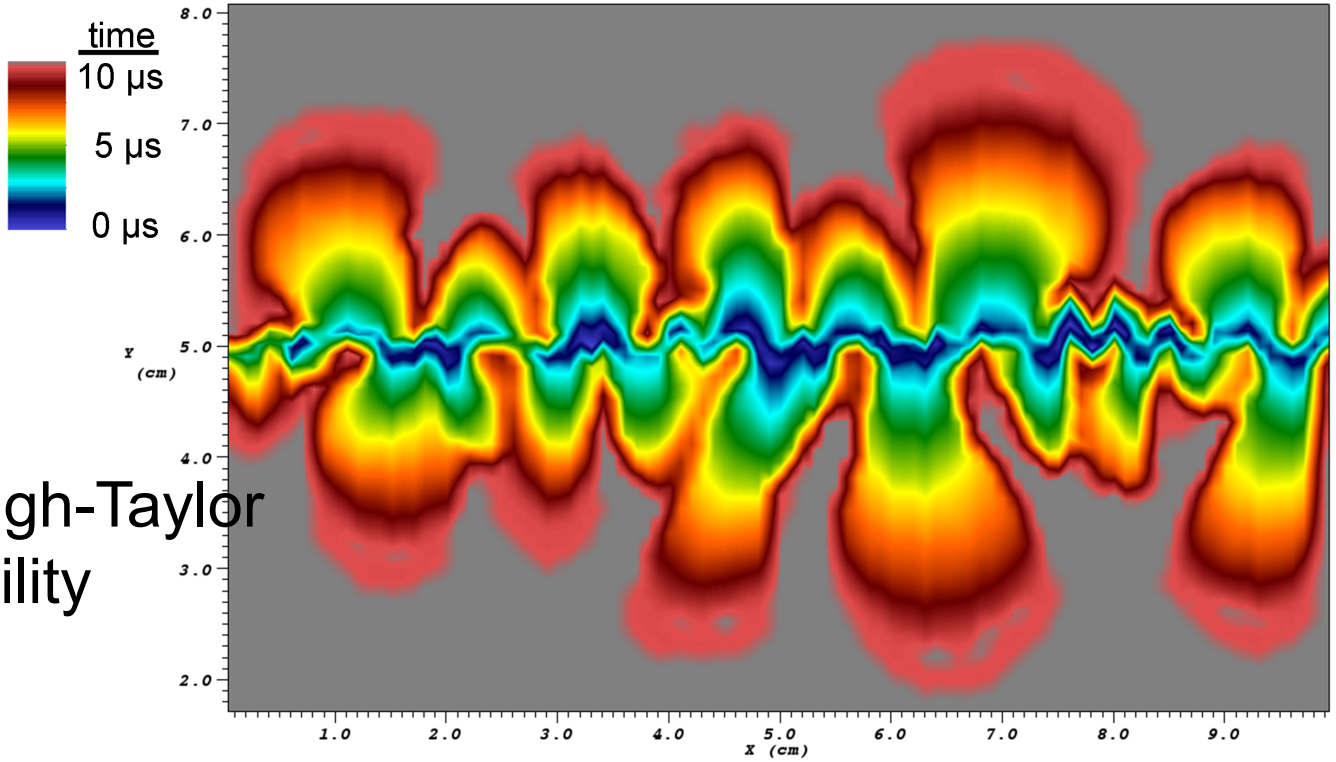
Equivalence Class Functions



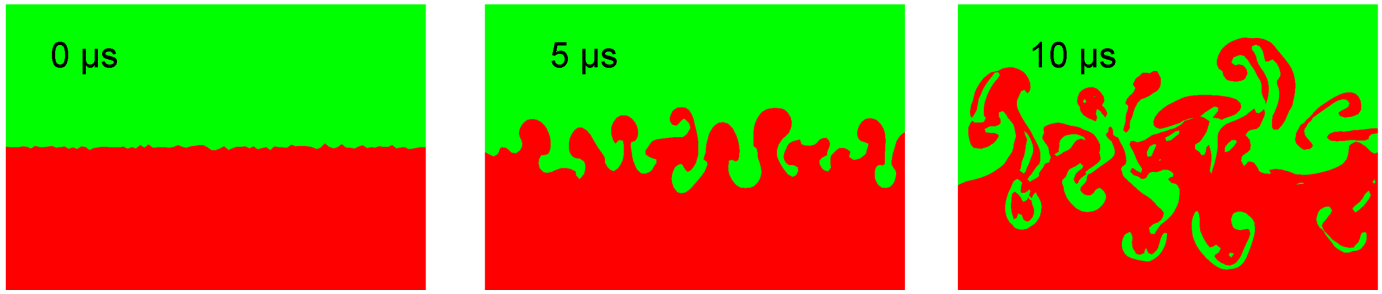
Particle advection

- Large investment in particle advection.
- Will discuss in Thursday's talk (IDAV view)

Comparative techniques have applications to better visualization of time-varying data.



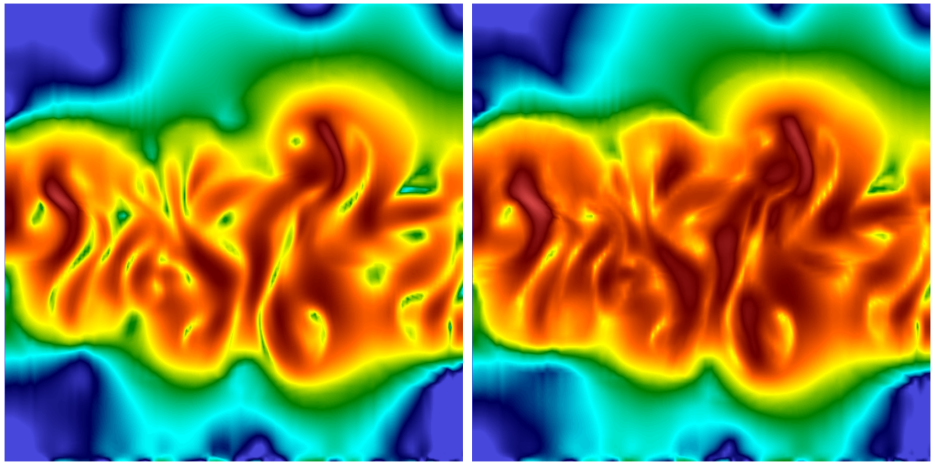
Rayleigh-Taylor Instability



Comparative techniques have applications with parameter studies/ensembles

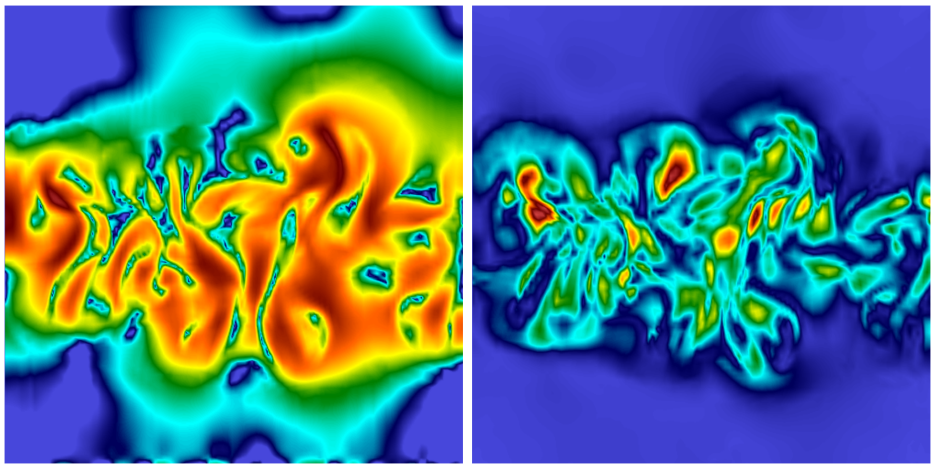
Studying 25 Rayleigh-Taylor Instability calculations (all at 10us)
Two “knobs”: turbulent viscosity coefficient, buoyancy coefficient
Five values for each knob, 25 pairs total

Average Speed over all 25



Max Speed over all 25

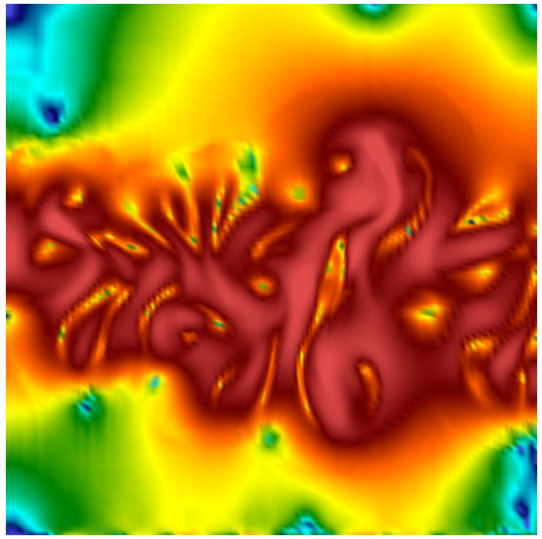
Min Speed over all 25

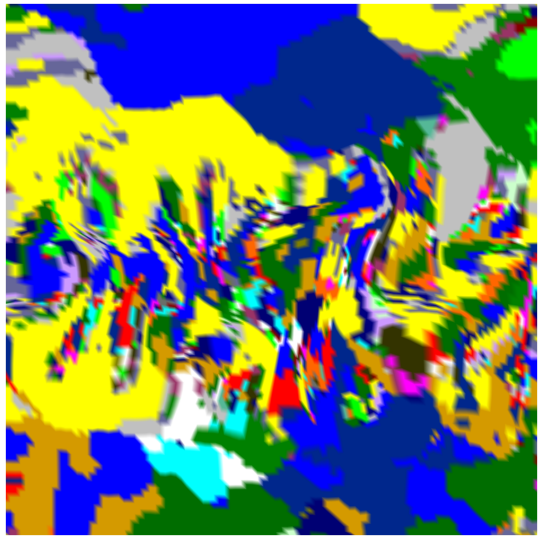


Biggest difference over all 25
(is this uncertainty quantification?)

Comparative techniques have applications with parameter studies/ensembles

Speed for one simulation





Coloring by Simulation ID with maximum speed

K0=V0, K1=V0 →

K0=V0, K1=V1 →

...

K0=V4, K1=V4 →

Coloring by "Knob 0" (buoyancy) with maximum speed

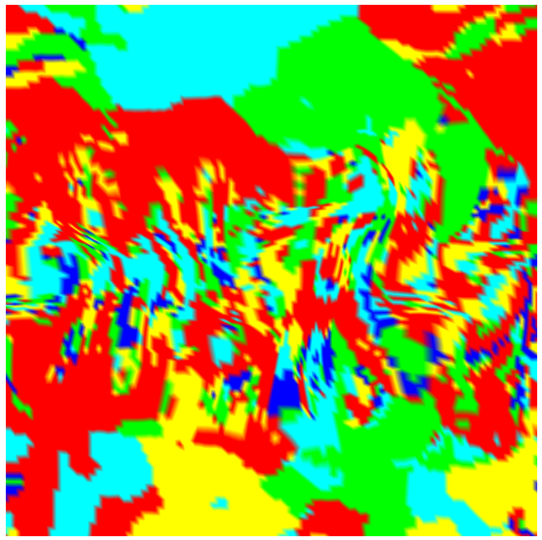
K0=V0→

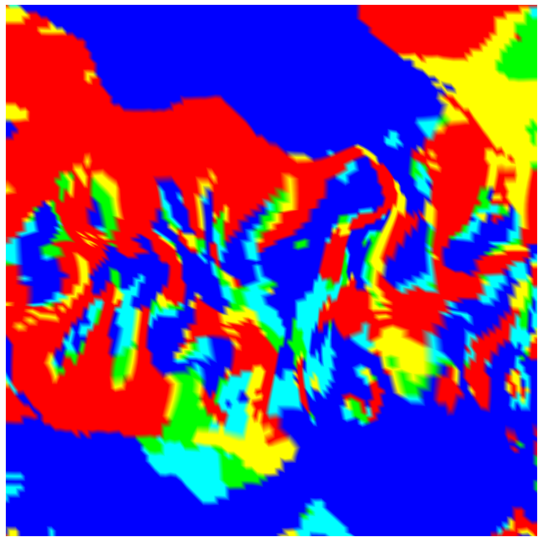
K0=V1→

K0=V2→

K0=V3→

K0=V4→





Coloring by "Knob 1" (viscosity) with maximum speed

K1=V0→

K1=V1→

K1=V2→

K1=V3→

K1=V4→

VisIt Resources

- Tutorial:
  - Two tutorials coming up @ 

- User resources:
  - Main website: http://www.llnl.gov/visit
  - Wiki: http://www.visitusers.org
  - Email: visitusers@ornl.gov
  - Email: visit-help-scidac@ornl.gov
- Development resources:
  - Email: visit-developers@ornl.gov
  - SVN: http://portal.nersc.gov/svn/visit
- Hank Childs, hchilds@lbl.gov