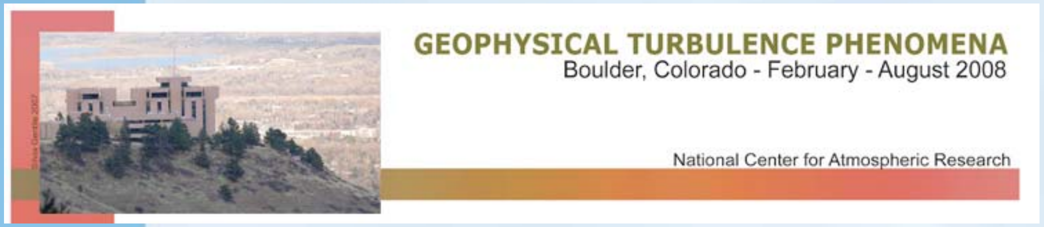
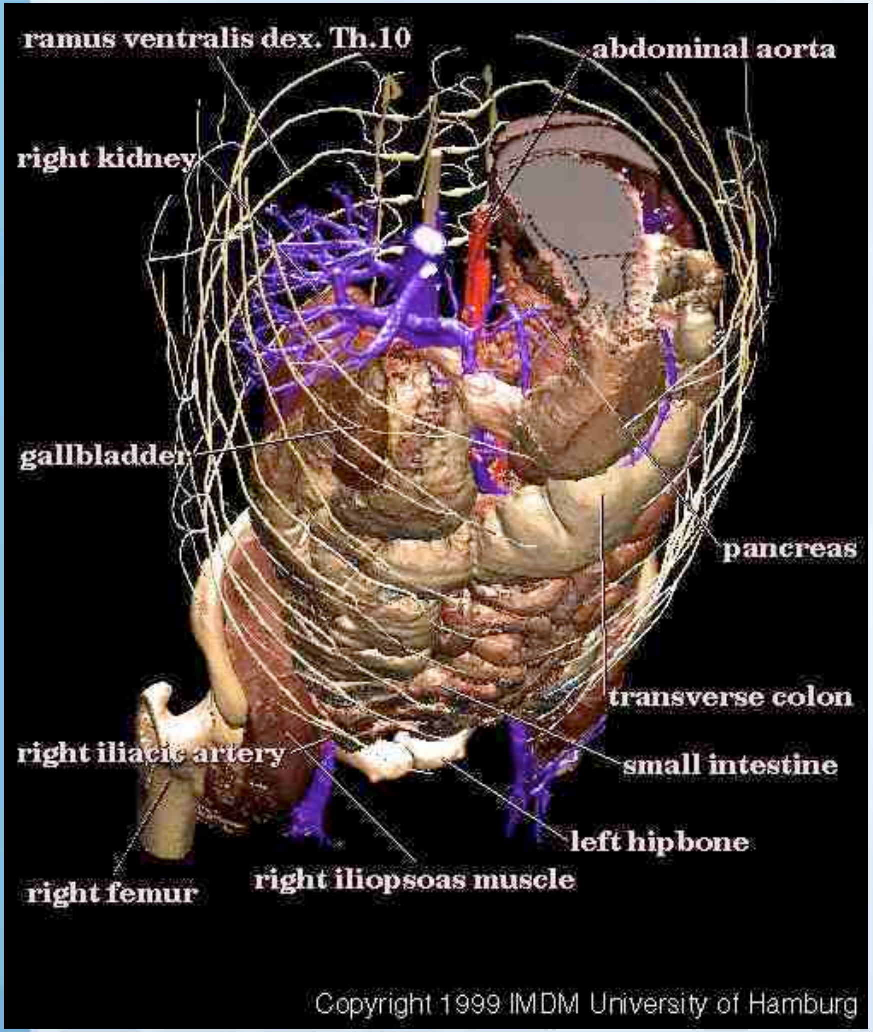

GEOPHYSICAL TURBULENCE PHENOMENA
Boulder, Colorado - February - August 2008
National Center for Atmospheric Research

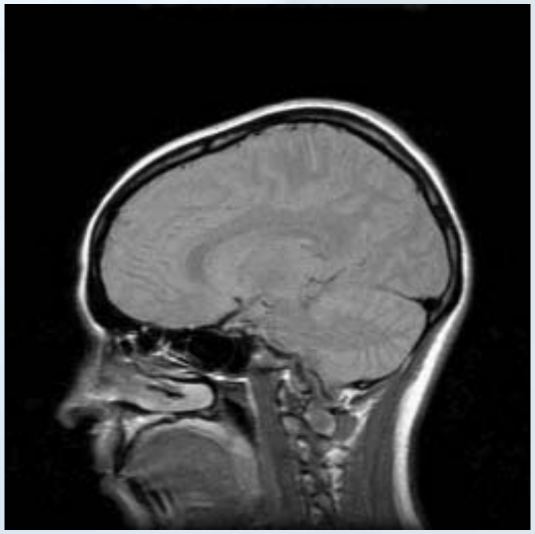
# A Broad Overview of Scientific Visualization with a Focus on Geophysical Turbulence Simulation Data

(SciVis 101 for Turbulence Researchers)

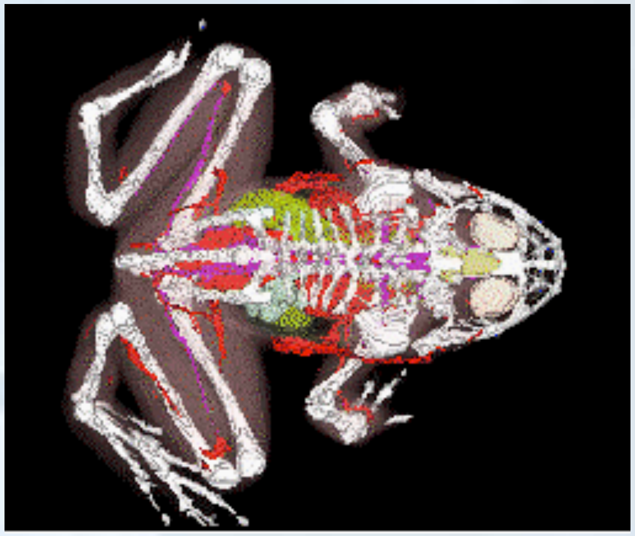
John Clyne
clyne@ucar.edu
National Center for Atmospheric Research

## Examples: Medicine

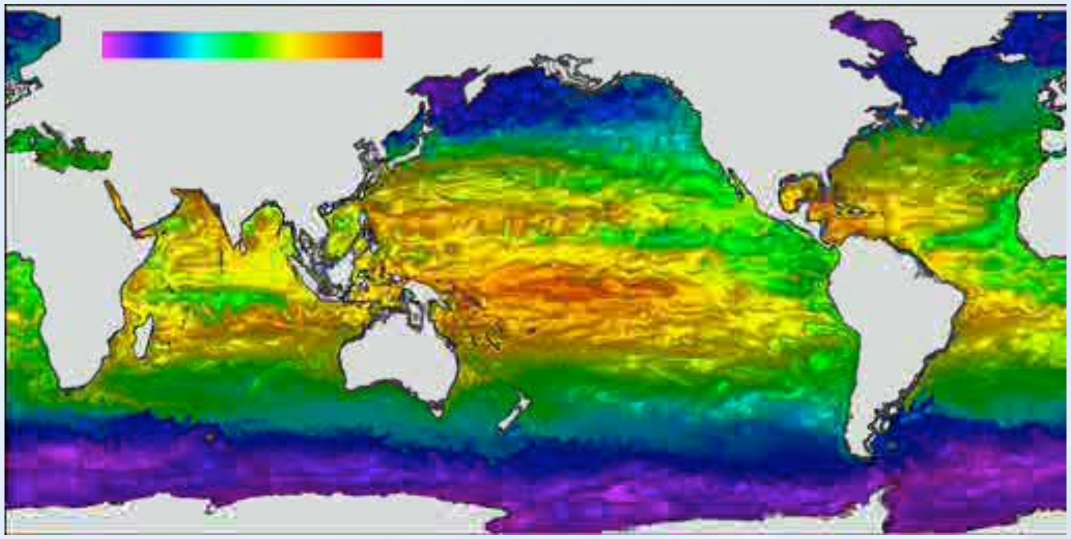


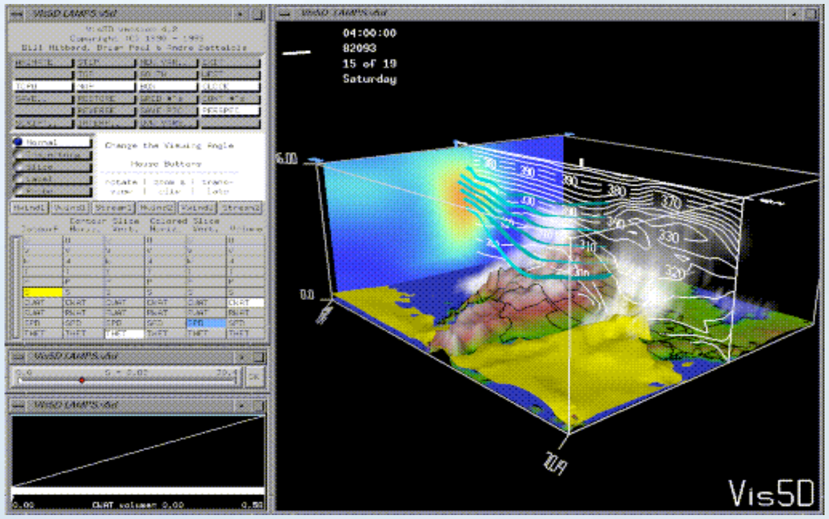


## Examples: Biology

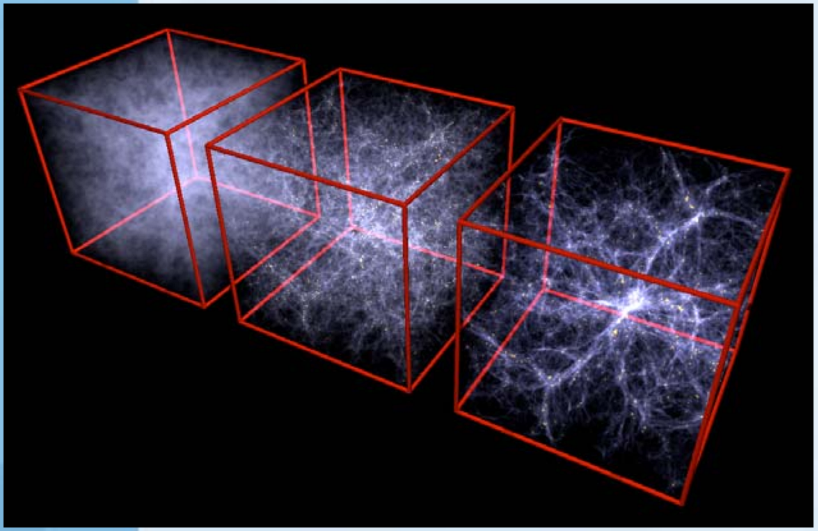


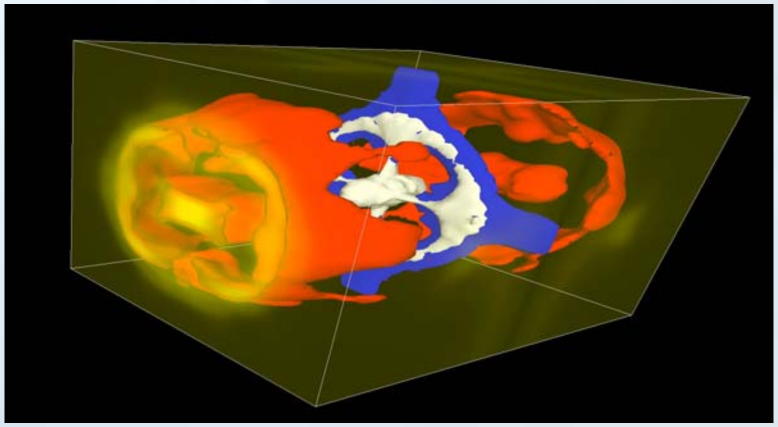
# Examples: earth sciences



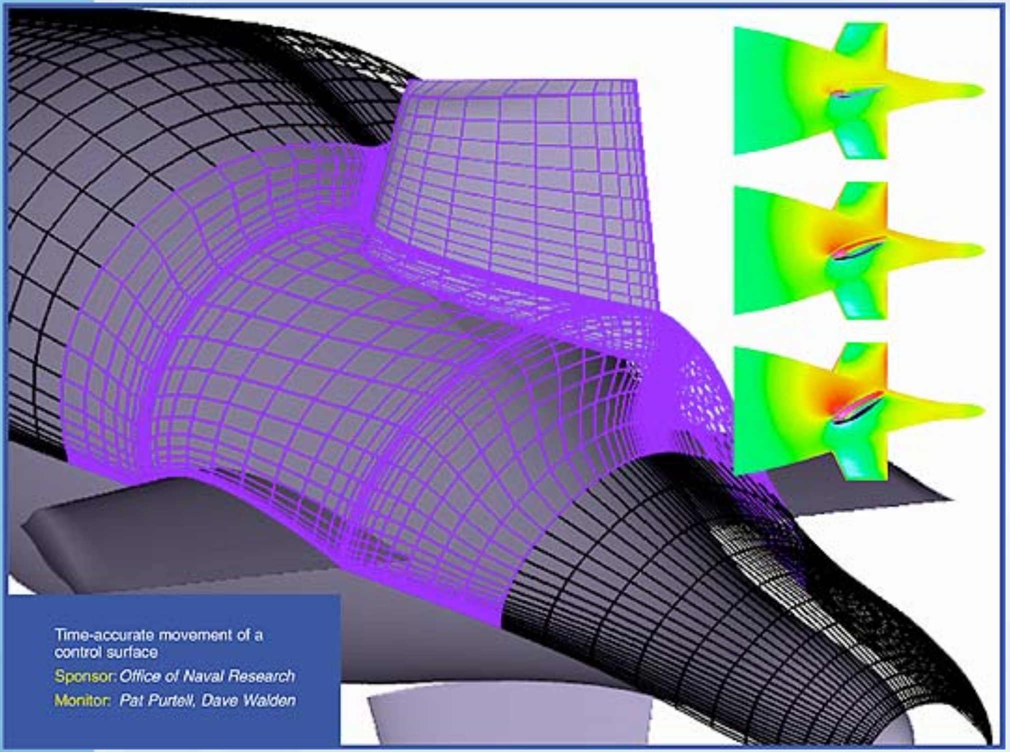


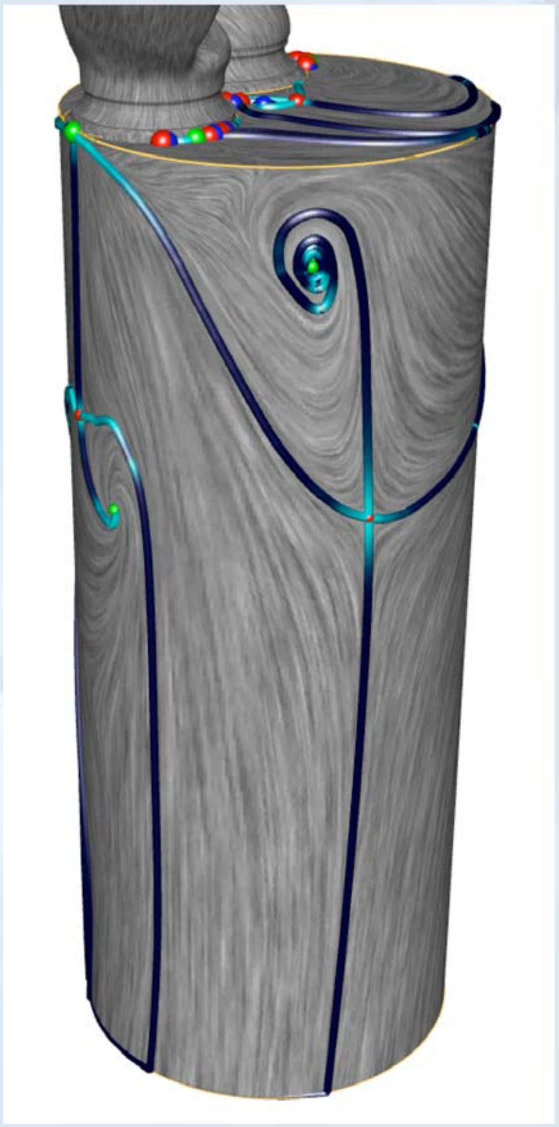
## Examples: cosmology



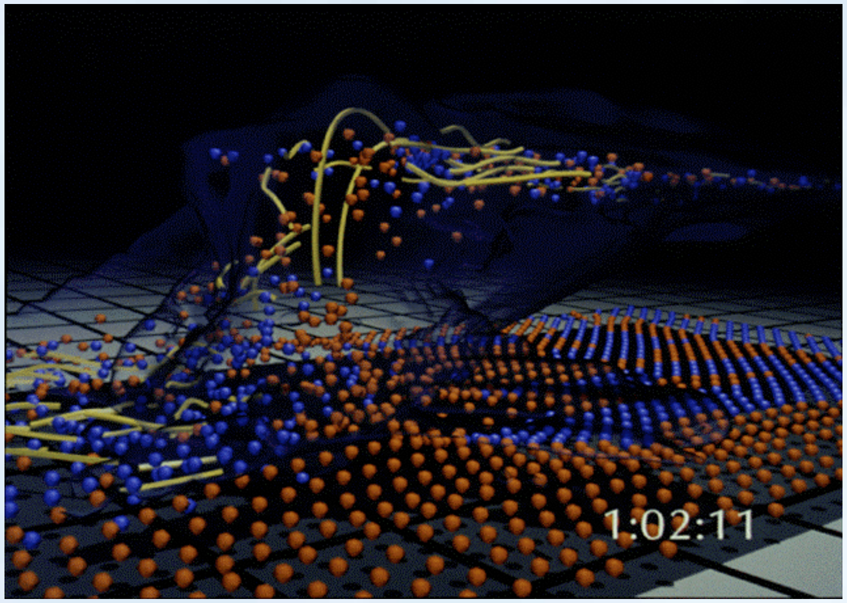


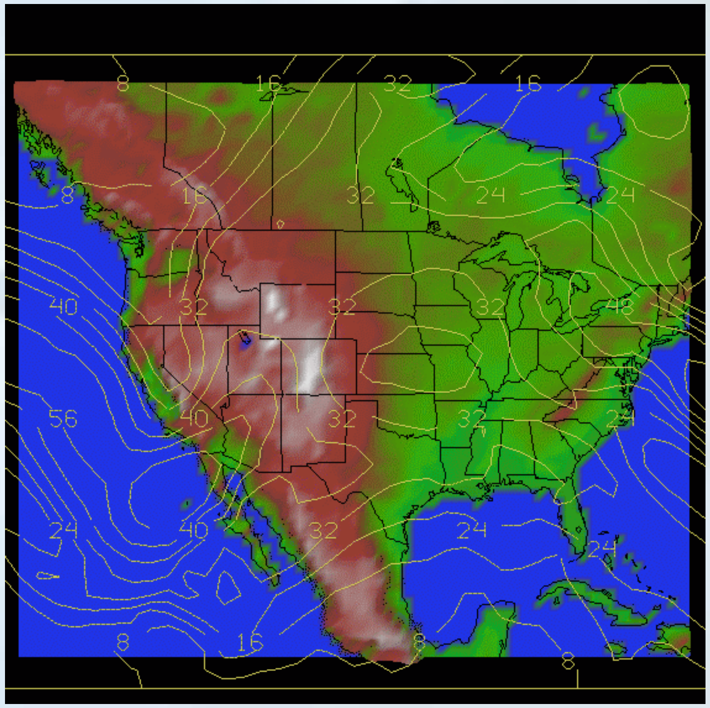
# Examples: engineering


Time-accurate movement of a control surface
Sponsor: Office of Naval Research
Monitor: Pat Purtell, Dave Walden

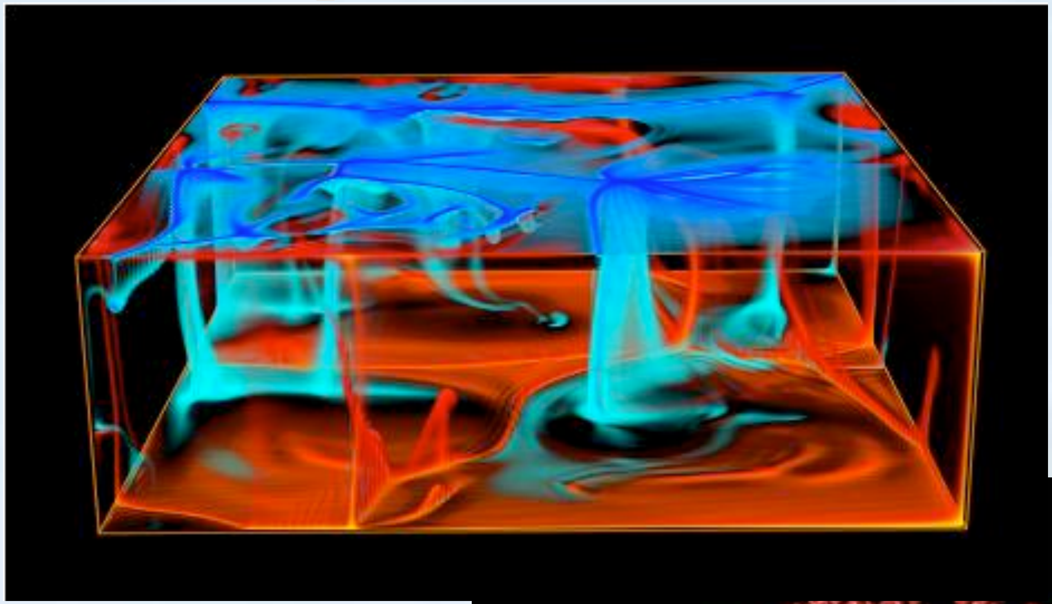


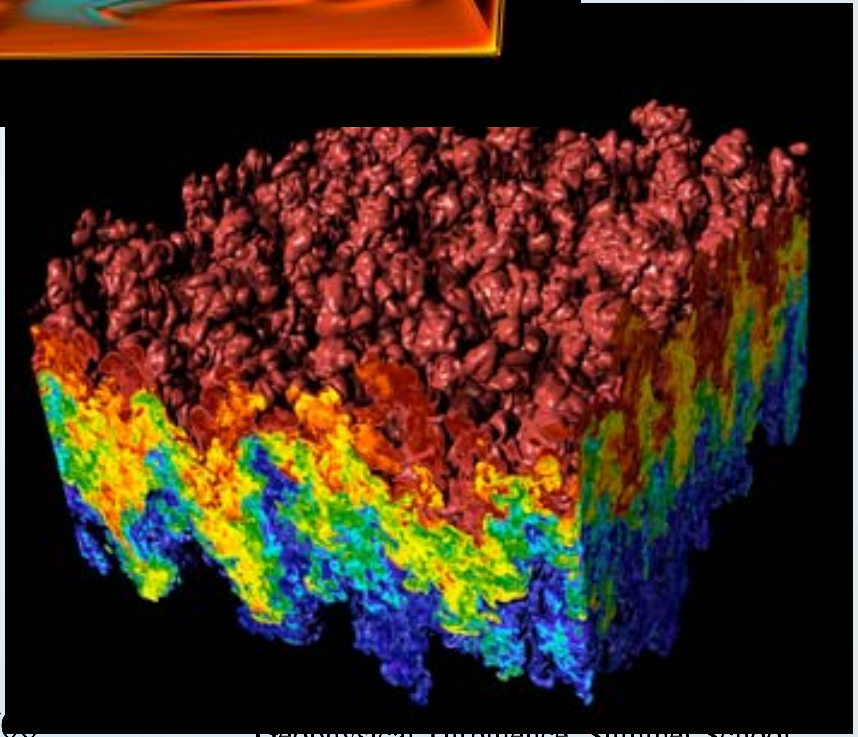
# Examples: Weather





## Examples: turbulence





## Questions for you to ponder over the next two weeks

1. What is the role of Scientific Visualization?

2. How effective is Scientific Visualization in that role?

## Definition: *Scientific Visualization (SciVis)*

NCAR

*"… method of computing. It transforms the symbolic into the geometric, enabling researchers to observe their simulations and computations. It enriches the process of scientific discovery and fosters profound and unexpected insights."*

1987 NSF Report: Visualization in Scientific Computing

*"… the process of exploring, transforming, and viewing data as images to gain understanding and insight into the data"*

The Visualization Toolkit, 2nd Edition

*"an interdisciplinary branch of science, primarily concerned with the visualization of three dimensional phenomena, such as architectural, meteorological, medical, biological systems. The emphasis is on realistic rendering of volumes, surfaces, illumination sources, and so forth, perhaps with a dynamic (time) component."*

Wikipedia

- A process for transforming numbers into pictures for the purpose of communication and/or analysis
- SciVis is ***NOT*** computer graphics!!
  - Computer graphics is a tool
  - SciVis is a process that combines computer graphics, HCI, signal processing, perceptual science, and a number of other disciplines and technologies

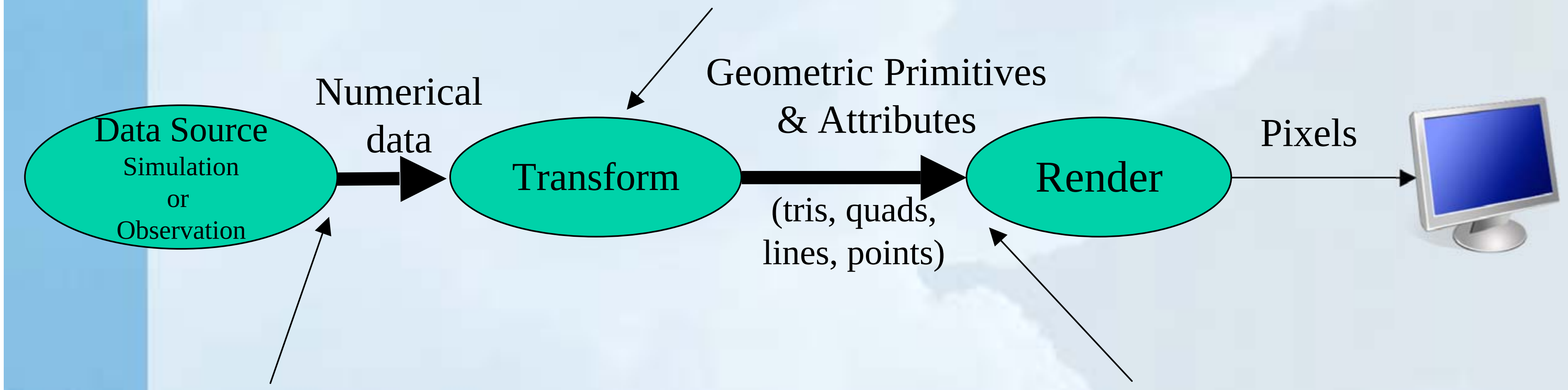
## Why Bother?

- Human beings are visual creatures… Approximately 50% of the brain's neurons are associated with vision
- SciVis is a powerful aid for filtering important details from the flood of data generated by today's computers and data acquisition systems
- SciVis enables communication of complex results to layman and experts alike
- Numerical simulations and experiments produce extremely large datasets
- The size of these datasets are increasing exponentially fast
- Statistics may not tell the entire story
- A picture is worth a thousand words….

## Visualization Pipeline

Visualization practitioners role: map (transform) numerical data to rendering primitives and associated attributes with these constraints:

- No standardization for input data format and numerous grid types
- Well-defined (but limited) computer graphics API



No canonical API

(Lots of data formats to choose from)

Reasonably well-defined API (e.g. OpenGL, Direct3D, X11)

## Recent paradigm shifts in SciVis

1. <u>How</u> SciVis is used
   - From: Post Processing of:
     - Pretty Pictures
     - Videos
   - To: Modern Data Analysis
     - Integrated Visual Data Exploration
     - Tools for Discovery and Analysis
2. <u>Where</u> visualization transforms are performed
   - From: CPU
   - To: GPU (Graphics Processing Unit)

## Two key requirements for moving from *pretty pictures* to data analysis

1. Interactive processing
   - Interactive
     - Better suited for analysis and exploration where visualization parameters not known a priori (e.g. view point, color maps, etc.)
     - May limit data size or algorithm complexity
     - Rendering (and more recently, transformation) typically relies on hardware graphics accelerators (GPU)
   - Batch
     - Requires a priori knowledge of what you want to see
     - Allow complex representations not possible in real time
     - Algorithms typically executed on the CPU
       - Better lends itself to in situ execution on the supercomputer
2. Quantitative information

## Modern day graphics processor (GPU)

- Capabilities
  - Processing cores: ~*256*
  - Memory bandwidth: ~*100 GBs*
  - Floating point performance: ~*1000* GFLOPS
  - Video (data) memory: ~1GB
  - Max number of instructions: ~2048
- Cost: ~$500
- Bus interface: PCIe x16 (4-8 GBs)
- Programming model: streaming vector processing (data viewed as streams, computation as kernels)
  - Graphics APIs (OpenGL, Direct3D) + shader language
    - Specialized 4GL languages for graphics, not general computing
  - More general purpose languages (e.g. CUDA) starting to appear
- Primary market: computer gaming
  - If the gaming industry doesn't need it, don't expect to see it on a GPU

Many times faster than today's CPUs (10-40x), but offer a limited and complex programming mode

## Characteristics of geophysical turbulence simulation data

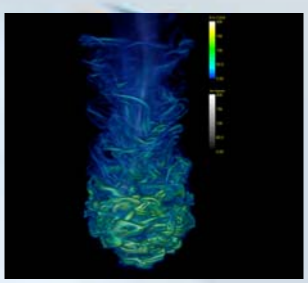
- Multidimensional
  - 2D and 3D, plus time
- Data volumes may be large, occupying Gigabytes or Terabytes of disk space
  - High spatial resolution

  and/or

  - High temporal resolution
- Discretely sampled on grids of varying geometric and topological regularity
- Multivariate
  - ~3-5
- Multi-type
  - Scalar, vector quantities

# Geophysical Turbulence Example #1

## Solar thermal starting plume



Mark Rast, NCAR/CU, 2003

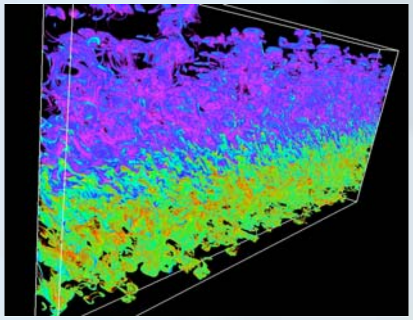
- 2003 - Simulation
  - 6 months run time
  - 504x504x2048 *stretched* grid
    - Rectilinear with non-uniform sampling
  - 5 variables
    - Vector: u,v,w
    - Scalar: temp, rho
  - ~500 time steps saved
  - 9 TBs storage (4GBs/var/timestep)
  - 112 IBM SP RS/6000 processors
- 2004 - Post-processing
  - 3 months
  - 3 derived variables (vorticity)
- 2004 - Analysis
  - On-going

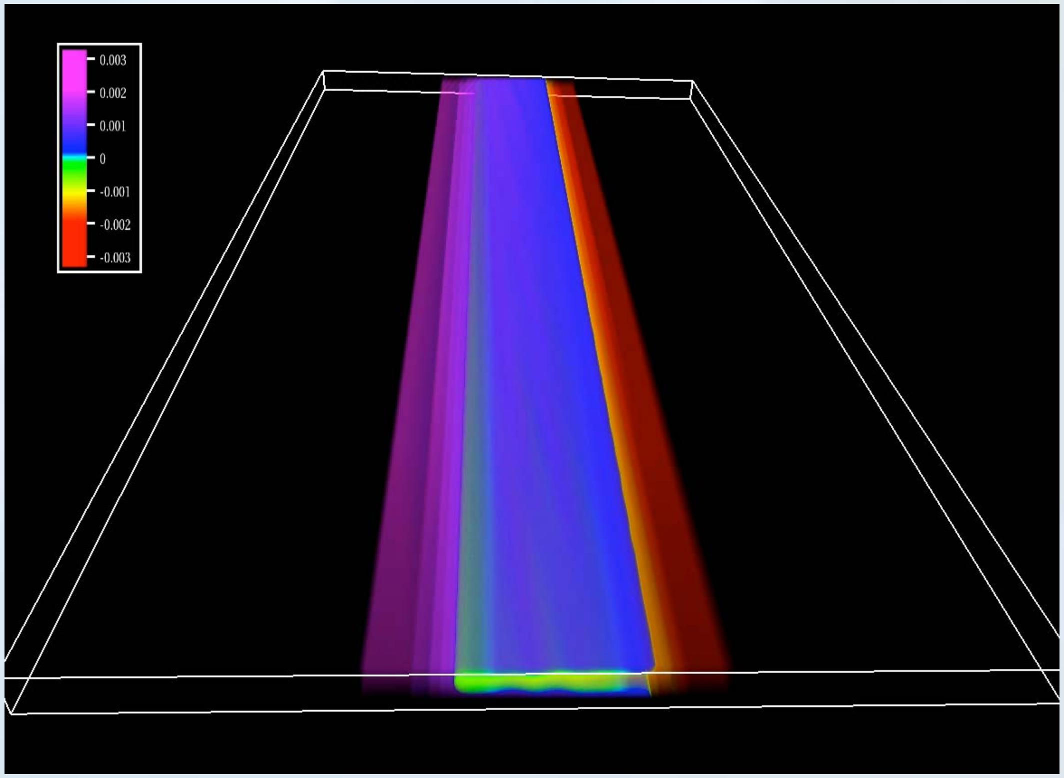
## Geophysical Turbulence Example #2

### Double-diffusive convective instability in sheared ocean flow

- 2007 Simulation
  - 10 days run time
  - 384 IBM Power5 processors
  - 6144x144x3072 Cartesian grid
  - 3 Variables
    - Vector: velocity
    - Scalar: temp., salinity
  - 12 GBs/field
  - 1 TB storage

Bill Smyth, U. Oregon, 2007



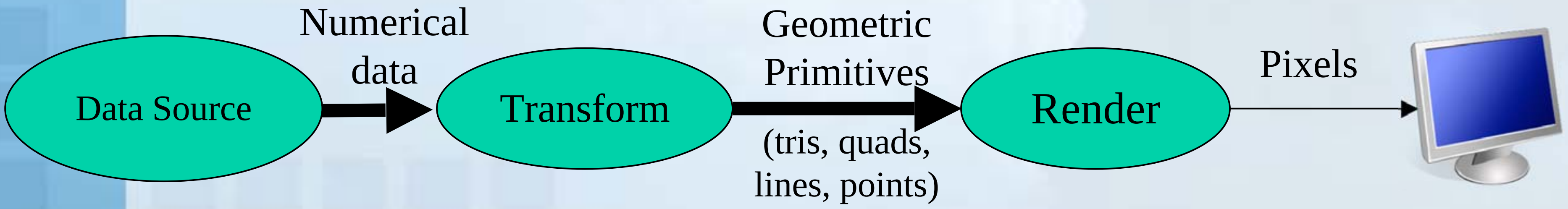


## How is Scientific Visualization of geophysical turbulence data impacted given the following:

- Desire to use SciVis to aid analysis
- Interactive processing requirements
- Characteristics of data (e.g. large, multivariate, time-varying, etc.)

## Visualization Pipeline
## Multidimensional Data

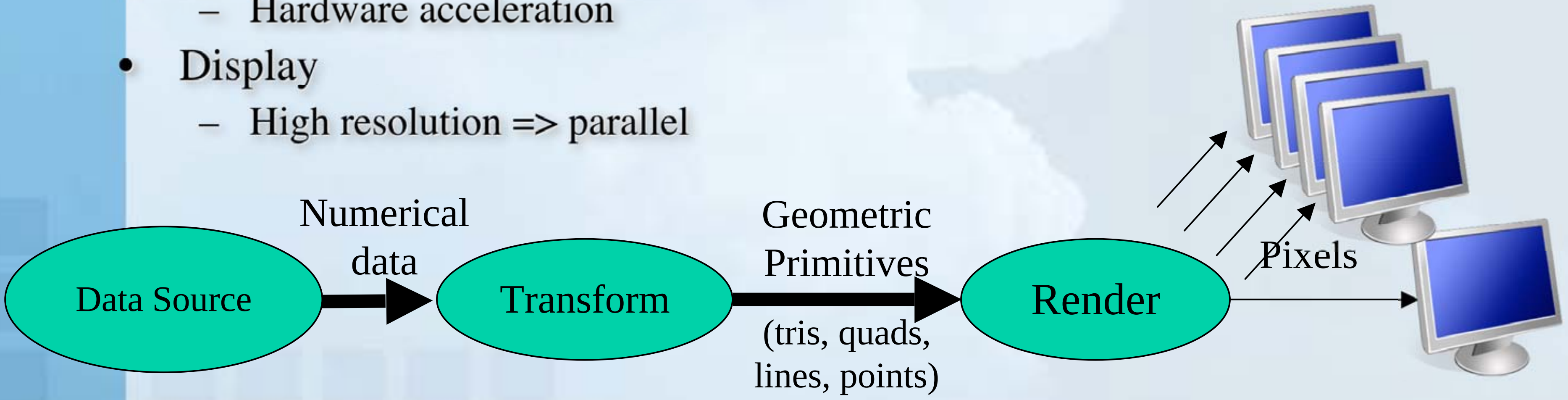
- Transform
  - Algorithms for handling 2D, 3D and 4D data
- Render
  - 3D rendering API (and typically supporting hardware)
- User Interface
  - Command line insufficient for 3D view manipulation



## Visualization Pipeline

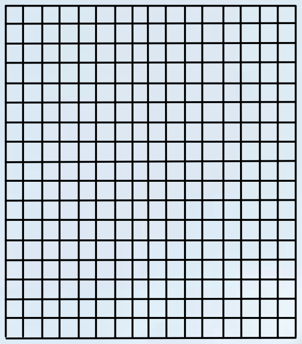
## Large (high resolution) Data & Interactivity

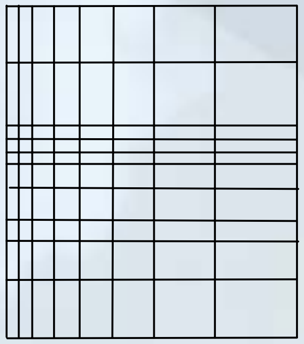
- Data Source
  - High performance IO
  - Data reduction
- Transform
  - Out of core algorithms
  - Parallel implementation
- Render
  - Parallel
  - Hardware acceleration
- Display
  - High resolution => parallel

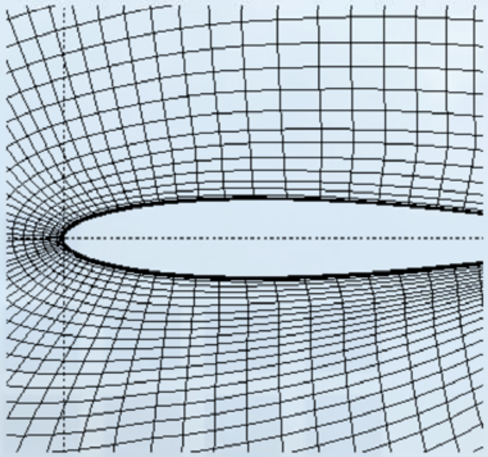


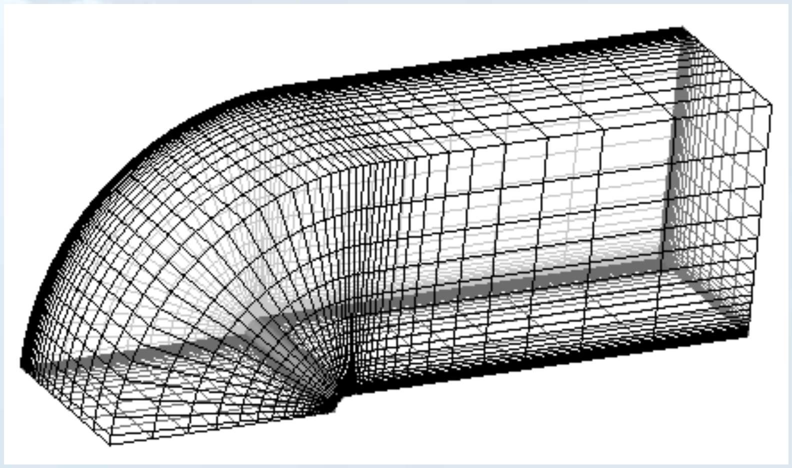
# Structured grids

- Regular topology
- Regular or irregular geometry
- Simple transformations
  - Computationally efficient
  - Broadly supported
- Difficult to fit to complex domains
- May over sample space leading to unnecessary computation and storage

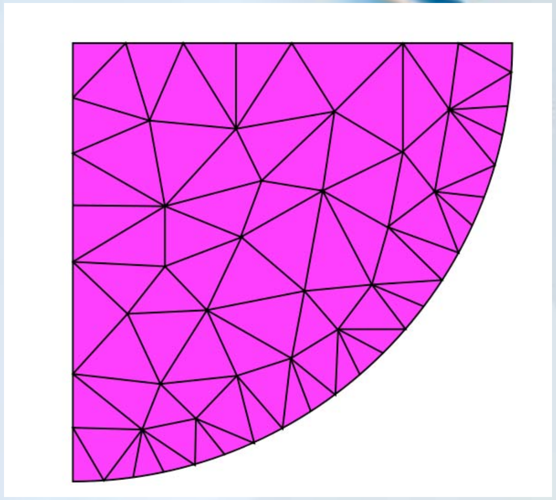




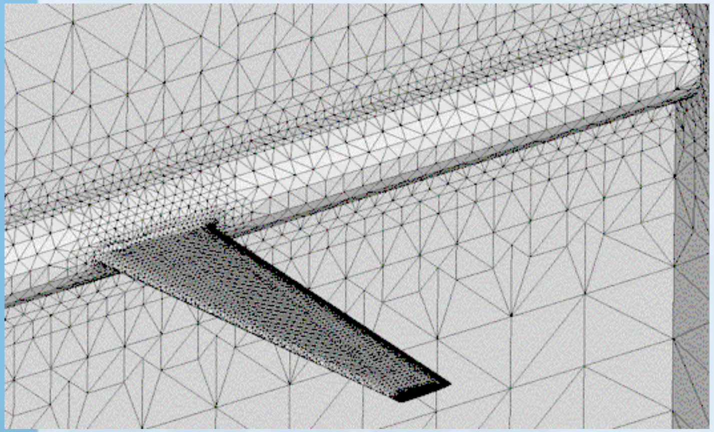


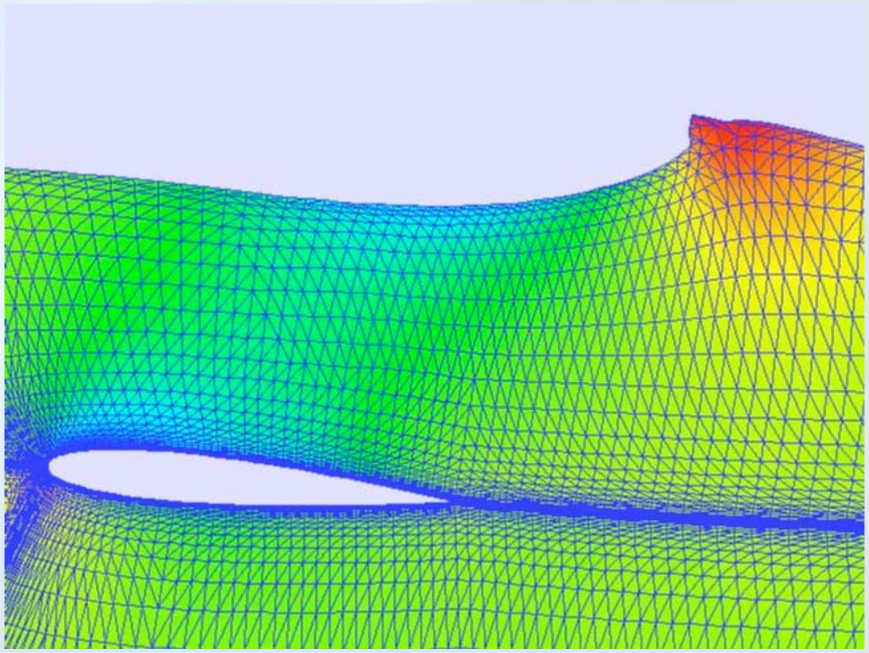


## Unstructured grids



- Irregular topology and geometry
- Adaptable to complex domains
- Sample space only where needed
- More complex transformations
  - Limited support (few tools)
  - Computationally expensive

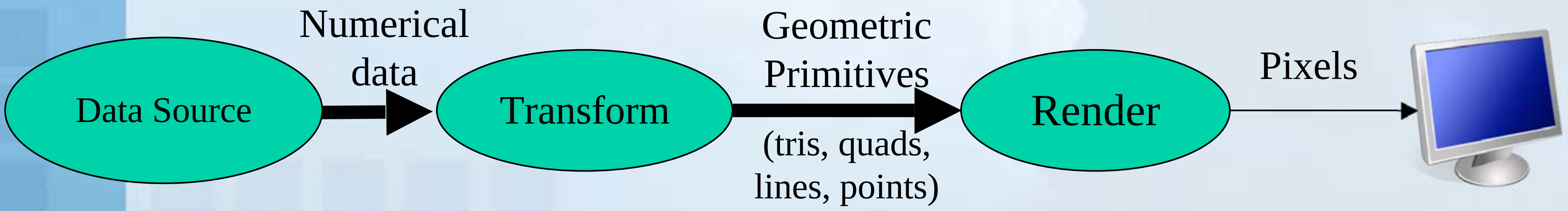




## Visualization Pipeline

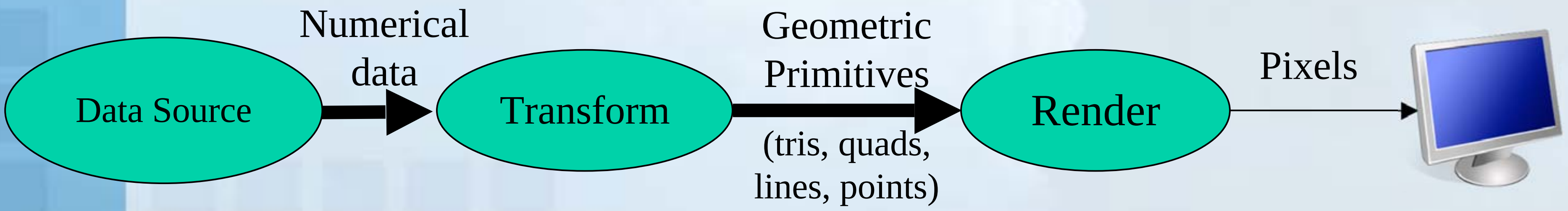
Numerous grids, multiple variables, scalar and vector types

- Transform
  - Flexible algorithms
    - May impact performance
    - Increases complexity
  - Specialized algorithms
    - Increased developer cost



## Visualization Pipeline
## Visual data analysis

- Render
  - Pick & Query
  - Annotation
- User Interface
  - Retrieve and present quantitative information
  - Support traditional (non visual) analysis operations



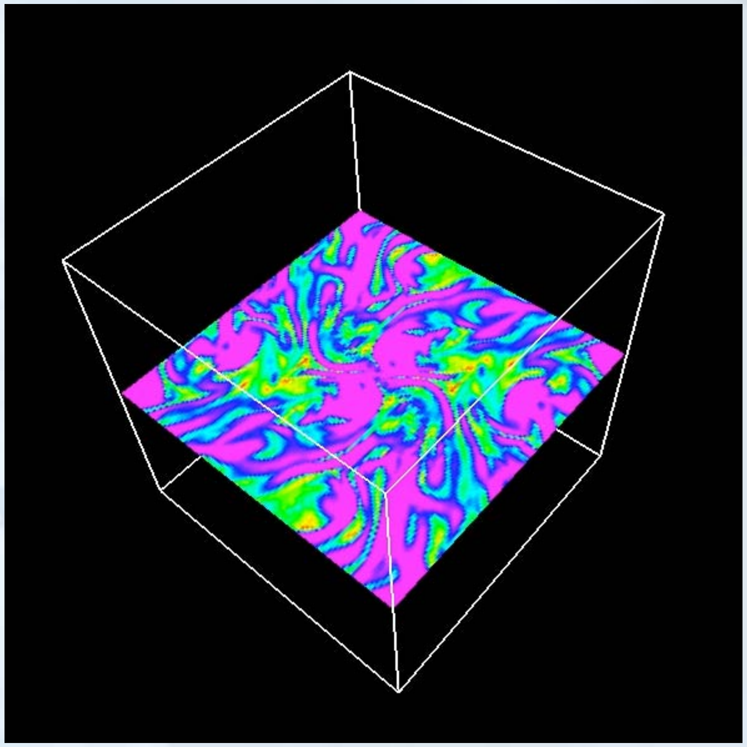
## Fundamental Transforms

## Fundamental Transformation Algorithms - Scalars

- Scalar (e.g. temp, pressure)
  - 2D
    - Pseudo Coloring
    - Contouring
    - Carpet plots
  - 3D
    - Cutting planes
    - Surface Fitting (Isosurfaces)
    - Direct Volume Rendering

## Cutting planes

- Intersect plane with data and color according to a user-defined data mapping
- Simplest of 3D algorithms
- Computational cost:
  - Transformation: low
  - Rendering: low
- Availability:
  - Most grids supported
- Provide only limited information
  - Difficult to get a sense global spatial domain

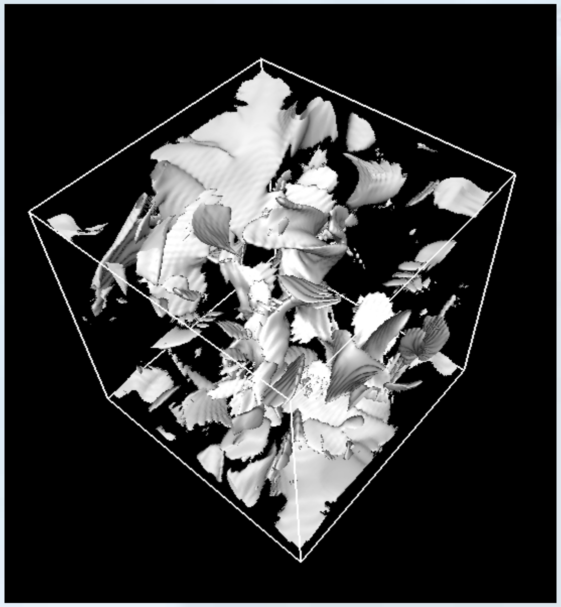


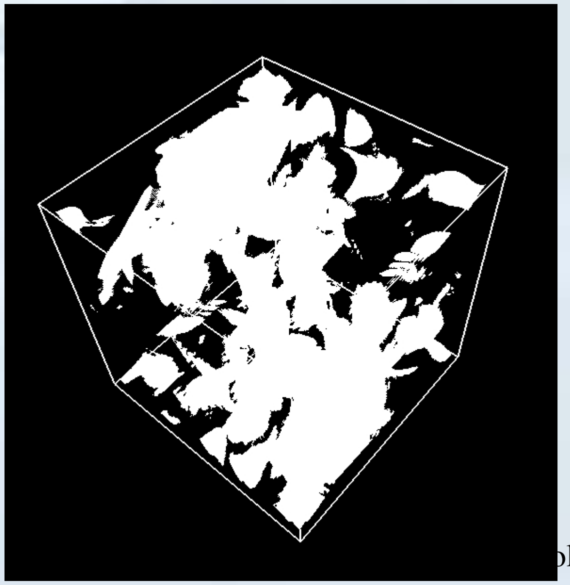
## Isosurfaces

- Surface defined by: *f(x,y,z)* - *c* = 0, where *c* is the isovalue of interest
  - 3D generalization of 2D contour lines
- Cost:
  - Transformation: high
    - Hardware (GPU) support for transform becoming available
  - Rendering: low
- Availability:
  - Most grids supported

- Illumination essential for spatial perceptual cues
- Gradient of field provides surface normal estimate, *N*, for lighting equation:

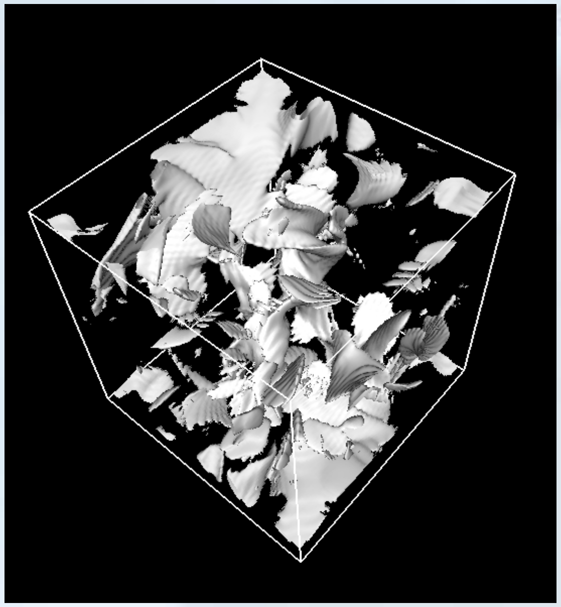
$$N = \nabla f(x, y, z)$$

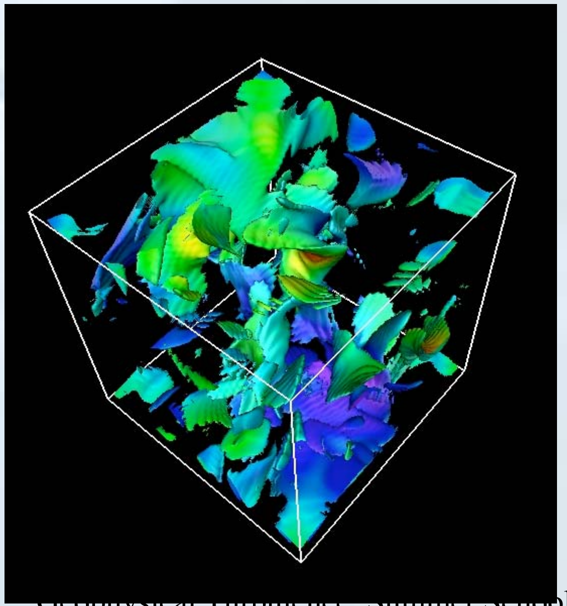




## Pseudo Colored Isosurfaces

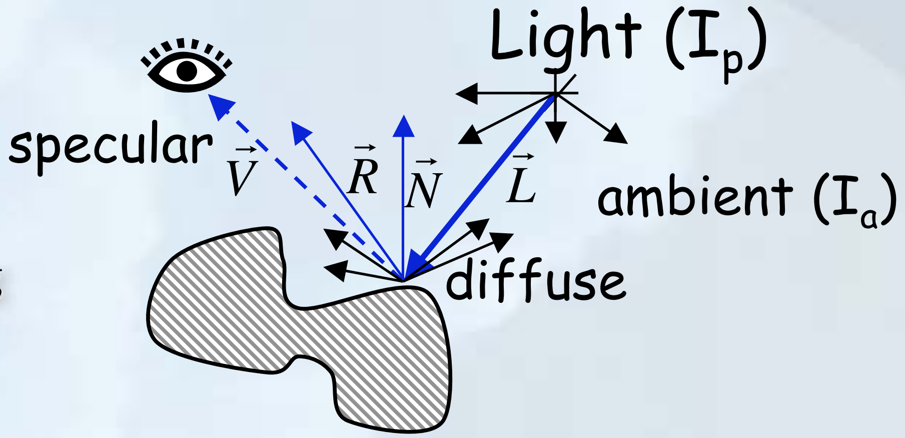
- Color map a second variable to iso field
- Relationship between two quantities
  - E.g. temperature and enstorphy





## A brief digression: illumination models in computer graphics

- Illumination is essential for 3D computer graphics
- Most typical illumination model for SciVis is the Phong model
  - No physical justification for Phong illumination - results just look reasonable
- Phong model (or any other illumination model) will modulate an object's color
- How do you interpret data-mapped color values that have been changed by the illumination model??



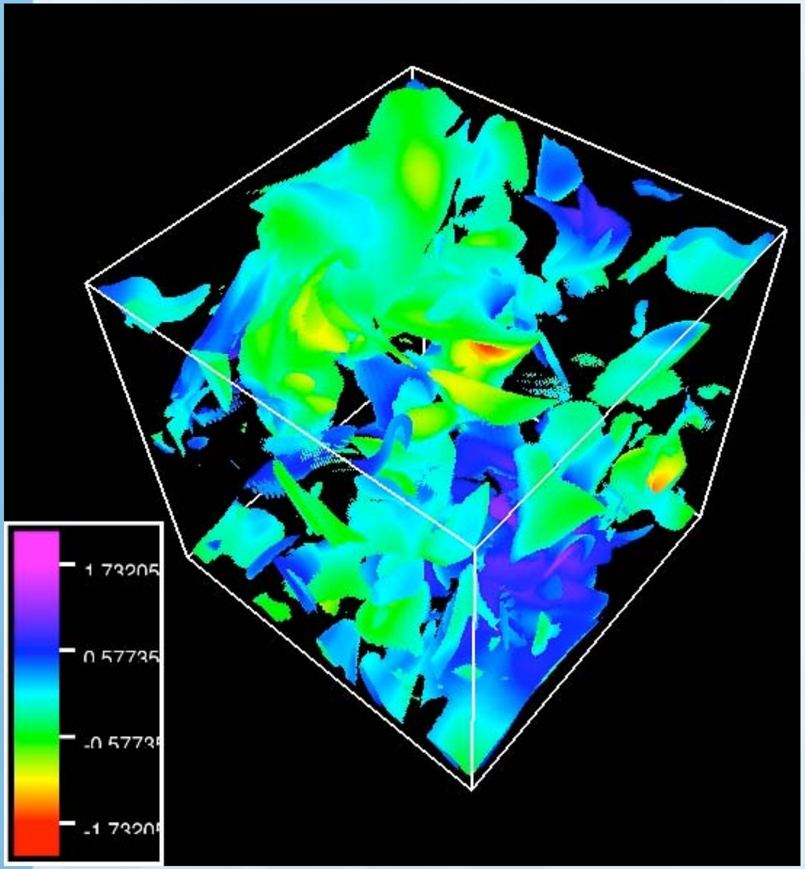
$$I = Ambient + Diffuse + Specular$$

$$Ambient = I_a k_d$$

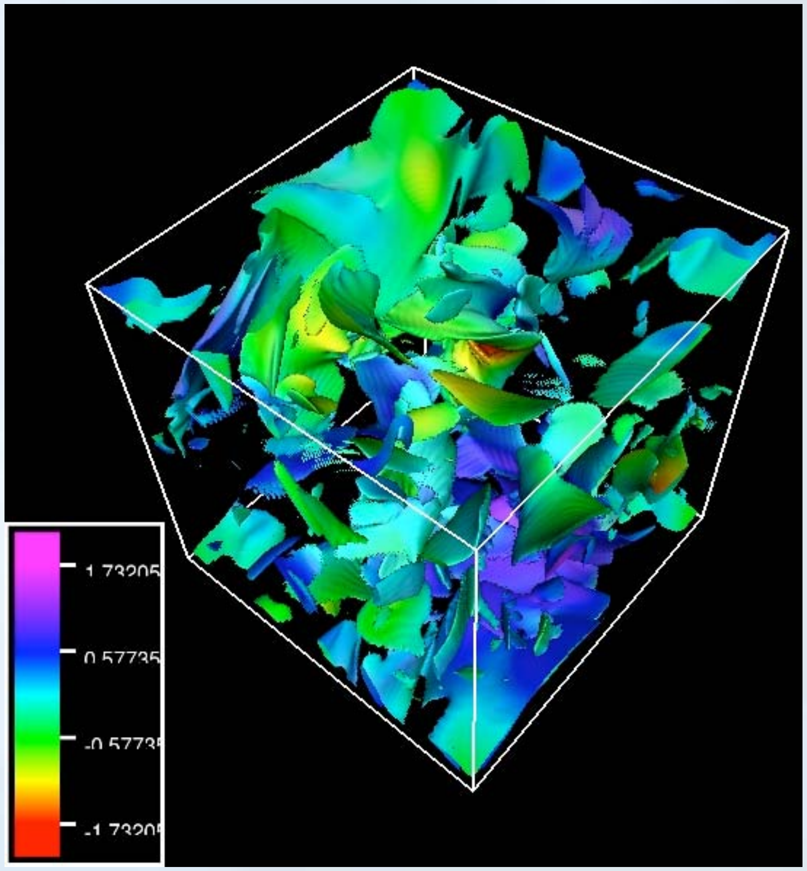
$$Diffuse = I_p k_d \left(\vec{N} \cdot \vec{L}\right)$$

$$Specular = I_p k_s \left(\vec{R} \cdot \vec{V}\right)^n$$

$$\left(0.0 \le k \le 1.0\right)$$



No lighting



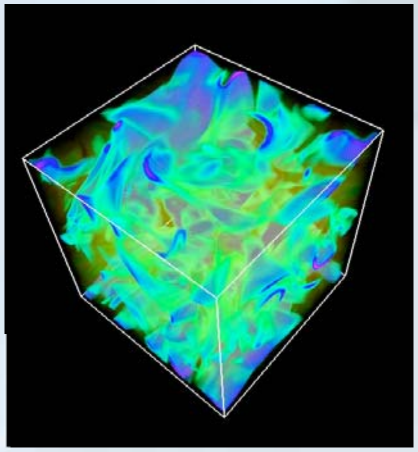
Lighting

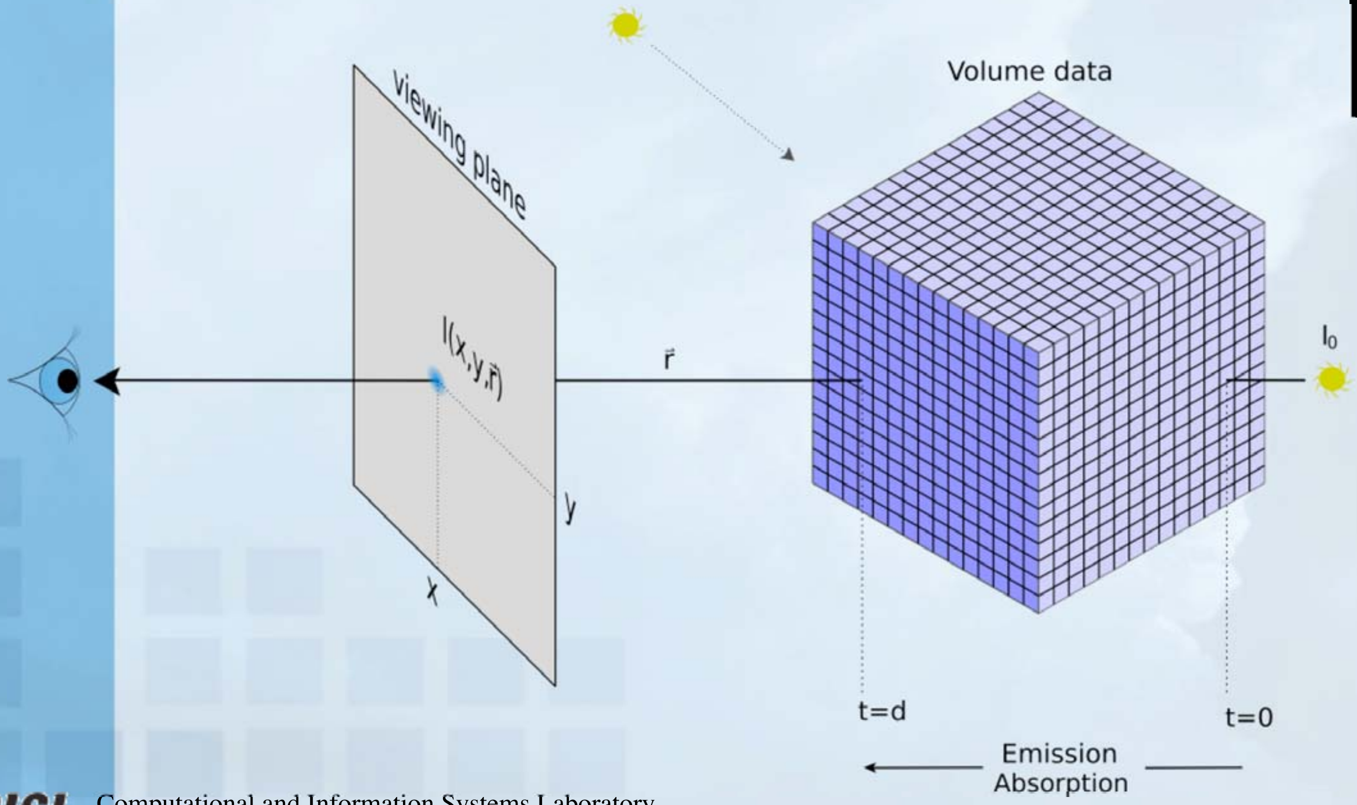
# Direct Volume Rendering

$$I(x,y,\vec{r}) = I_0 e^{-\tau(0,d)} + \int_0^d c(t) e^{-\tau(t,d)} dt$$

$$\tau(d_0,d_1) = \int_{d_0}^{d_1} k(t) dt$$

Emission-absorption optical model





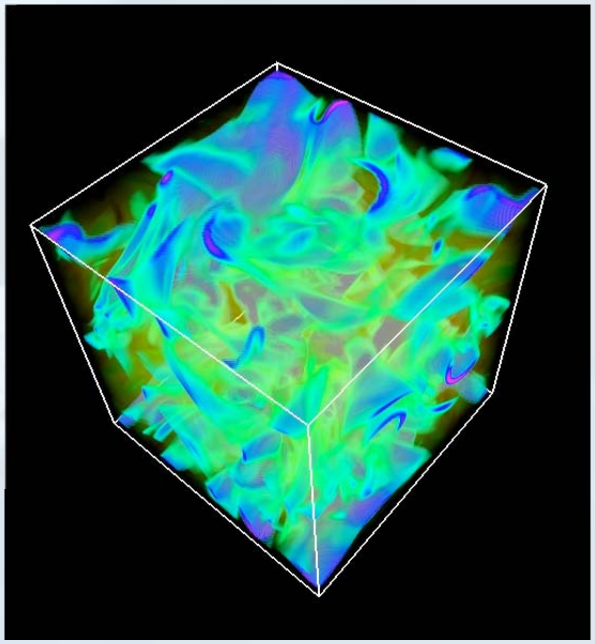
## Direct Volume Rendering

- Display a range of values
  - Map color and opacity:

    $C = T_c(f(x,y,z))$

    $O = T_o(f,x,y,z))$

  Where $T_c$ and $T_o$ are user defined “transfer” functions
- Provides holistic view of data, well suited for amorphous features such as clouds or fog
- No graphical primitives!
- Cost:
  - Transform + render: high
    - GPU implementations common
- Availability:
  - Limited support for complex grids
  - Irregular topology or geometry => CPU

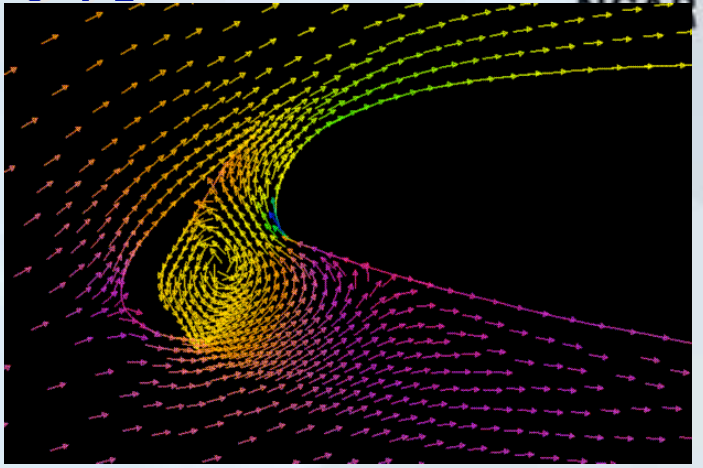


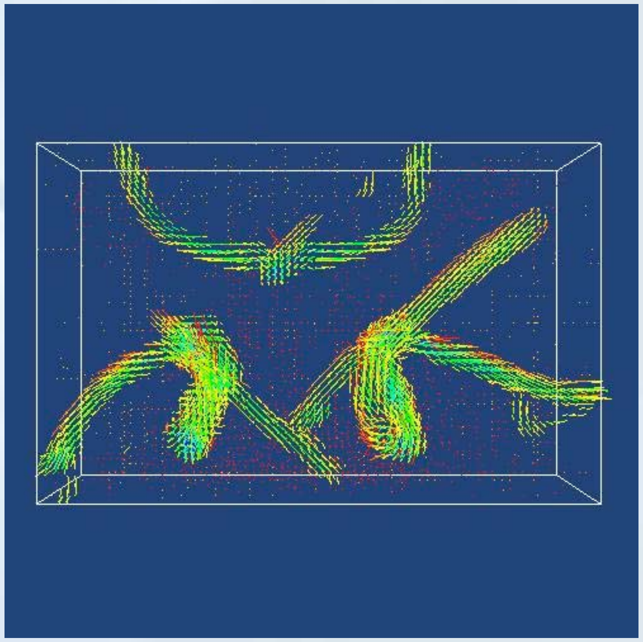
## Fundamental Transformation Algorithms - Vectors

- Vector (e.g. velocity, magnetic field)
  - Hedge Hogs (oriented glyphs)
  - Integral curves
  - Dense texture methods

## Hedge Hogs (oriented glyphs)

- Map a glyph (e.g. arrow) to each sample point in the field, oriented according to the field solution
  - Glyphs may be scaled by the field's magnitude or some other scalar quantity
- Cost
  - Transform: low
  - Render: varies
- Availability
  - Easily implemented on all grid types
- Poorly suited for 3D data
  - Visual clutter
  - Spatial scale ambiguities



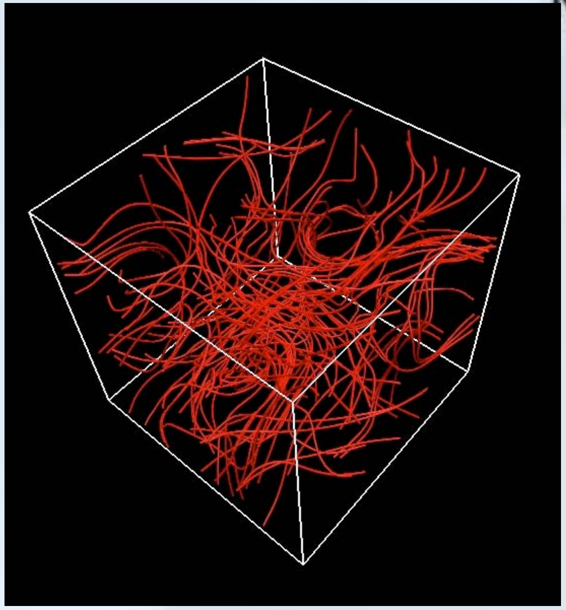


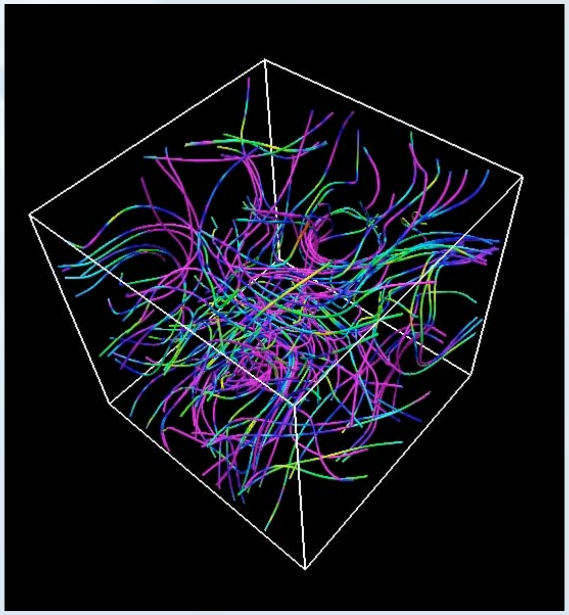
# Integral curves for steady fields (stream lines)

- Curve parameterized by *s* defined by:

$$\frac{d\vec{p}}{ds} = \vec{v}(\vec{p})$$

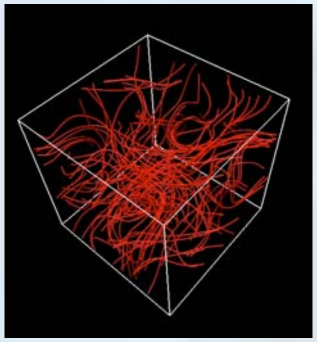
$$\vec{p}(s_1) = \vec{p}_0 + \int_{s_0}^{s_1} \vec{v}(\vec{p})\,ds$$





- Cost:
  - Transform: varies (dependent on number of curves)
  - Rendering: varies
- Availability:
  - Wide grid support
- No options for hardware (GPU) acceleration currently
- Care must be taken in 3D to avoid visual clutter (limits # of lines, reducing cost)

# Integral curves for unsteady fields (streak lines & path lines)



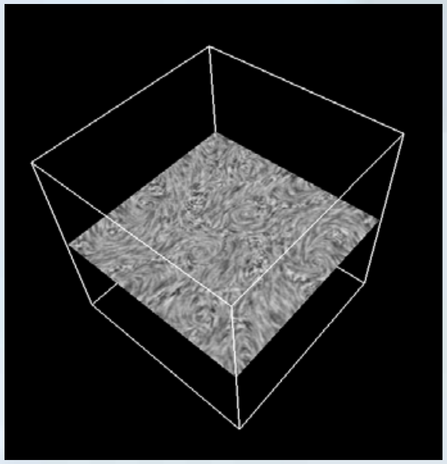
$$\frac{d\vec{p}}{dt} = \vec{v}\left(\vec{p}(t), t\right)$$

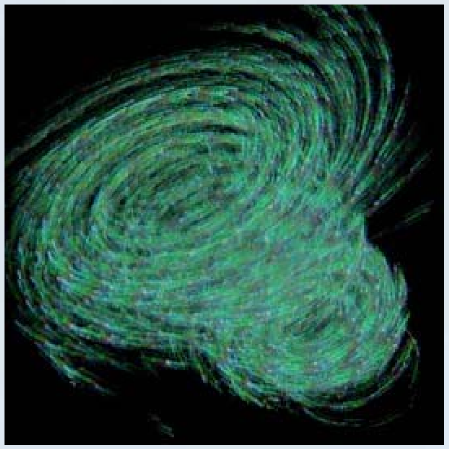
$$\vec{p}(t_1) = \vec{p}(t_0) + \int_{t_0}^{t_1} \vec{v}\left(\vec{p}(t)\right) dt$$

- Streaklines: locus of points of all the fluid particles that have passed through a particular spatial point in the past (dye steadily injected into the fluid at a fixed point extends along a streakline)
- Pathlines: trajectories that individual fluid particles follow
- Cost:
  - Transform: very high
  - Rendering: varies
- Availability
  - Very limited due to low demand
- No options for hardware (GPU) acceleration currently
- Care must be taken in 3D to avoid visual clutter

## Dense texture vector field visualization

- Vector field analog to direct volume rendering
  - Volume render a noise field advected by flow
  - Provide dense spatial representation of vector field
  - No geometric primitives
- Cost:
  - Transform & rendering: high
    - Numerous GPU implementations exist
- Availability:
  - Generally restricted to Cartesian grids





## Questions?

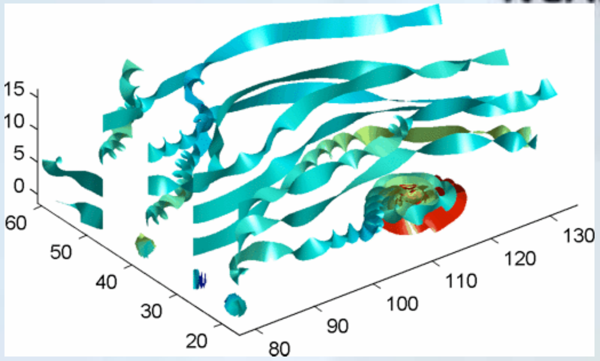
- References

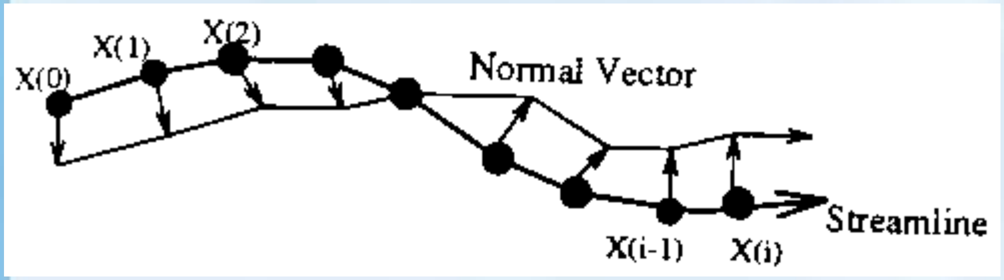
*The Visualization Tookit*, Third Edition. Will Schroeder, Ken Martin, Bill Lorensen

*Visualization Handbook*. Charles Hansen and Chris Johnson, 2004

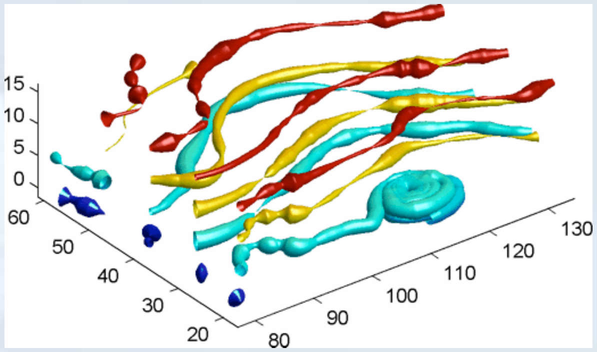
## Stream ribbons and stream tubes

- Stream ribbons
  - Shows helicty in flow $\left(\vec{v} \cdot [\nabla \times \vec{v}]\right)$

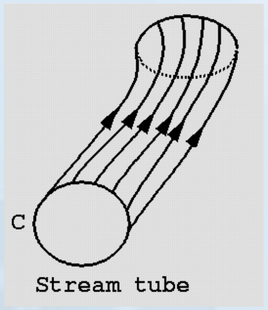




- Stream tubes
  - Closed curve bounded by streamlines
  - Illustrates convergence and divergence



15
10
5
0
60
50
40
30
20
80
90
100
110
120
130



C
Stream tube