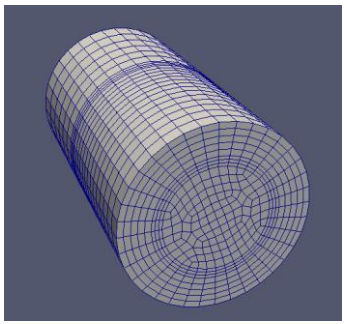
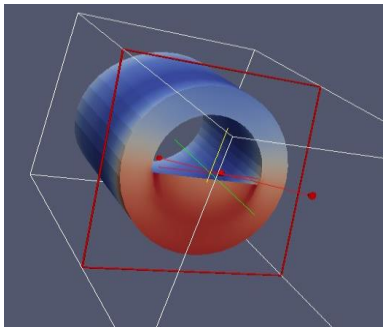
Basic Visualization Techniques & ParaView

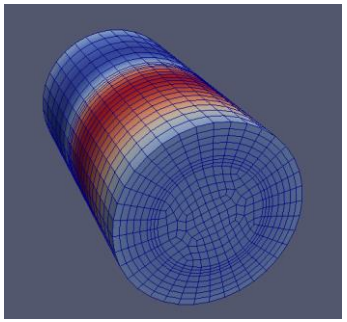
Courtesy of Prof. Han-Wei Shen
(CSE 5544, The Ohio State University)

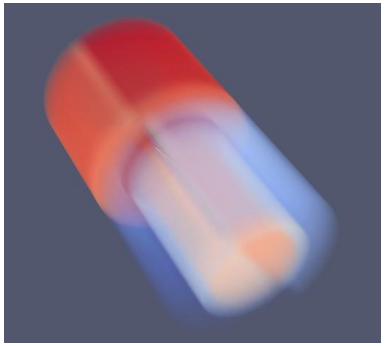
Visualizing Scientific Data

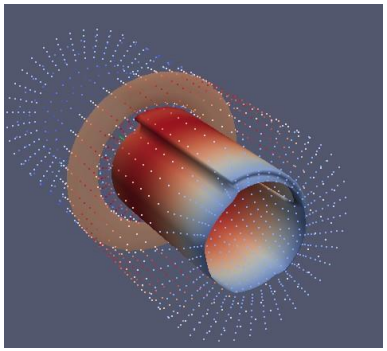
- Common Visualization Techniques
  - Mesh view
  - Outer surface with attributes
  - Slicing
  - Glyphing
  - Contouring
  - Volume rendering

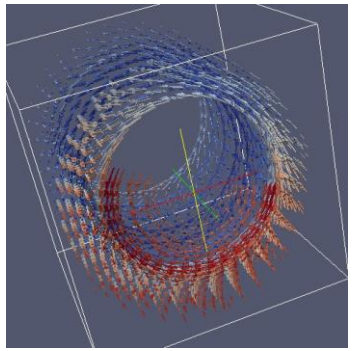






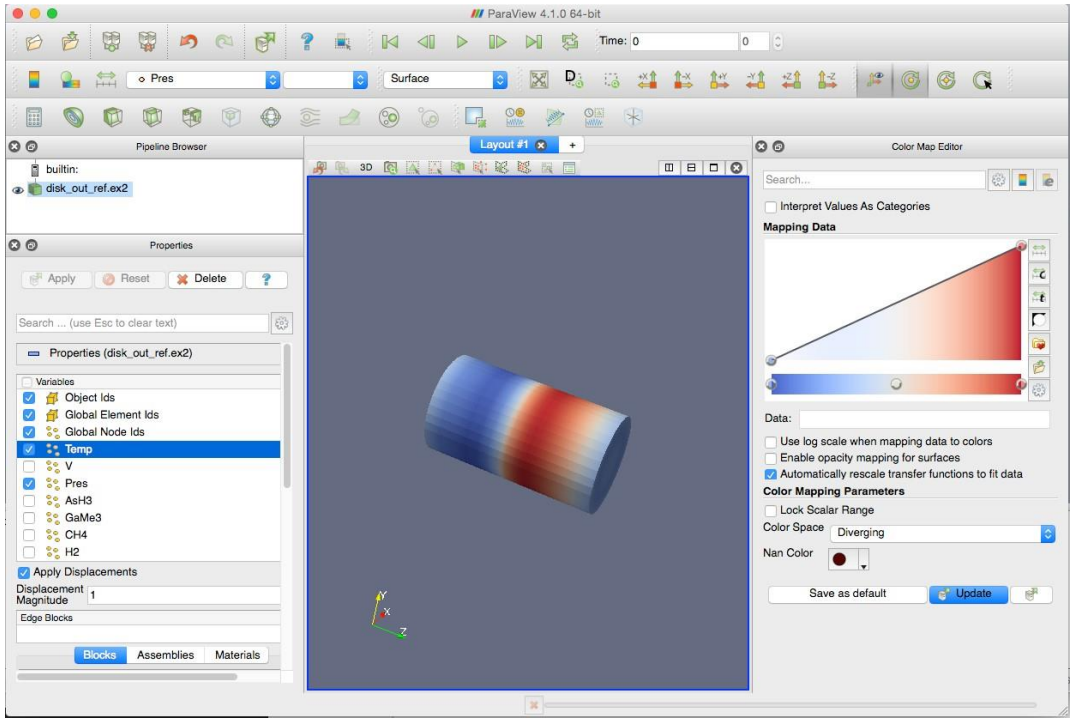






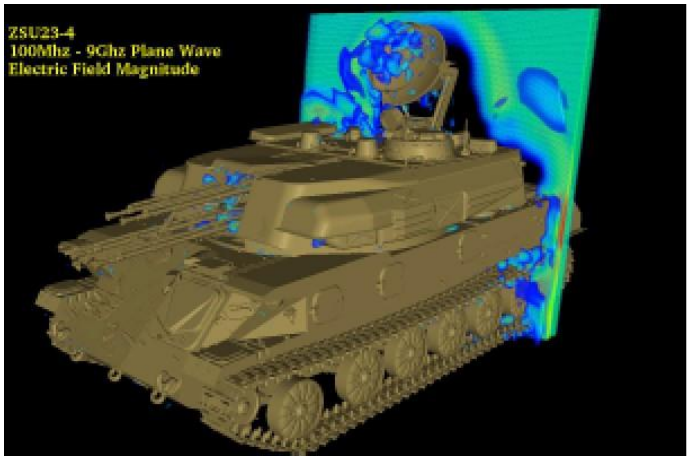
Visualization Software

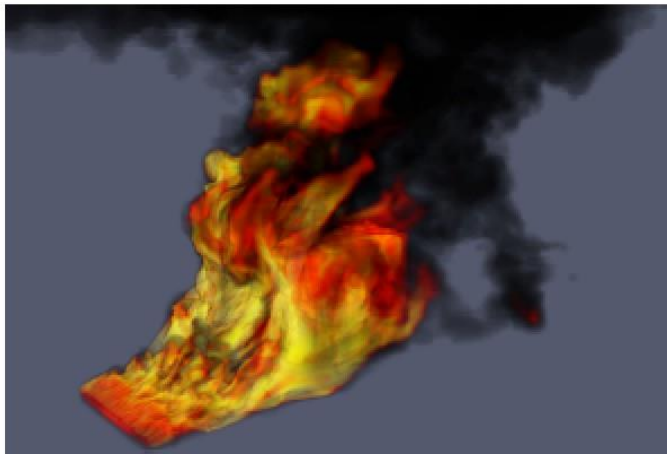
- We will explain how to generate the common visualizations using the ParaView visualization software



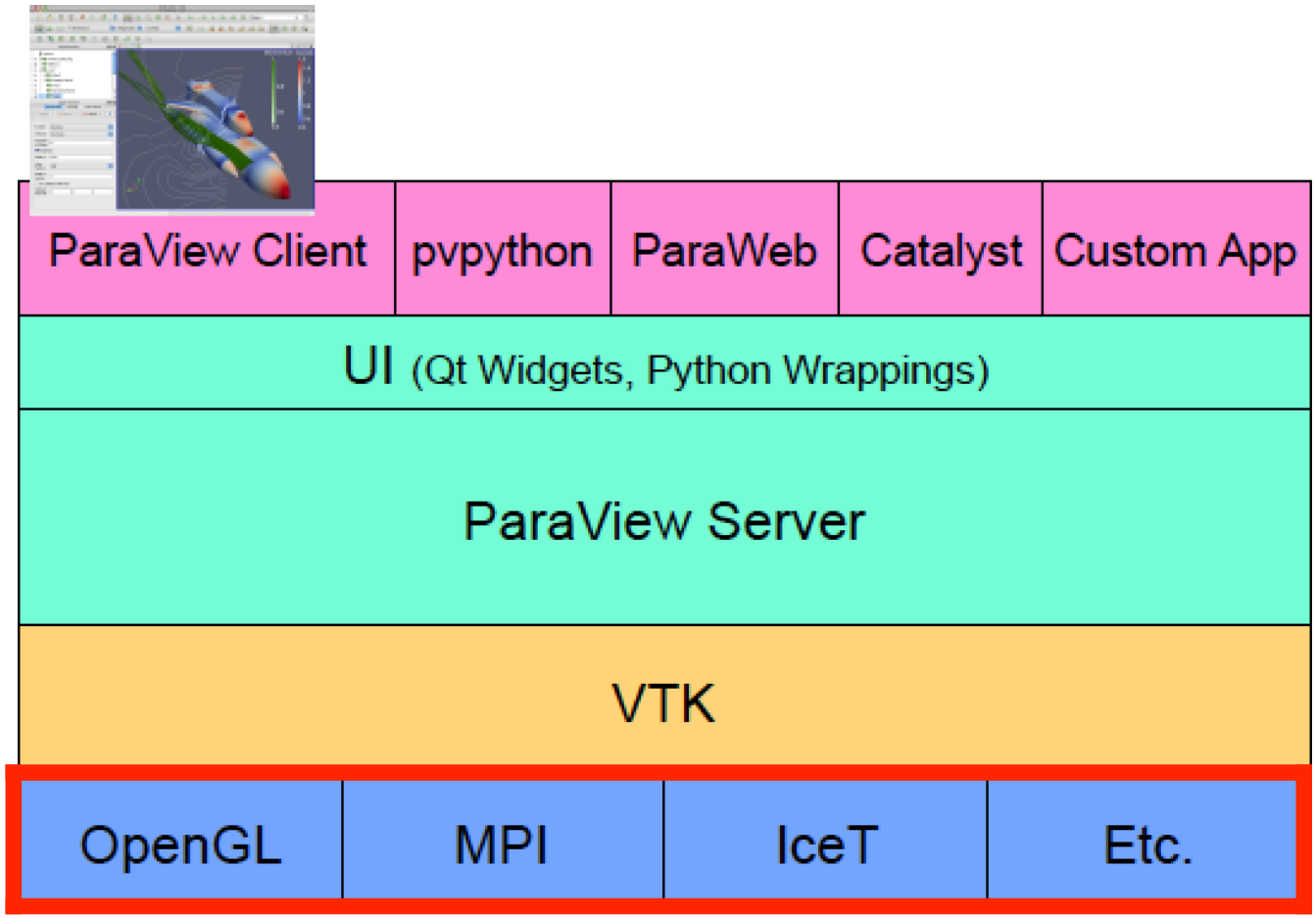
What is ParaView

- An open-source application for visualizing scientific data sets
- Supports a wide range of platforms, from laptop to supercomputers with 100,000 cores
- Built on top of VTK, the visualization toolkit, but with intuitive graphical user interface
- Modular design, can be controlled using scripting language such as python
- Can run on distributed memory parallel computers to process large data sets

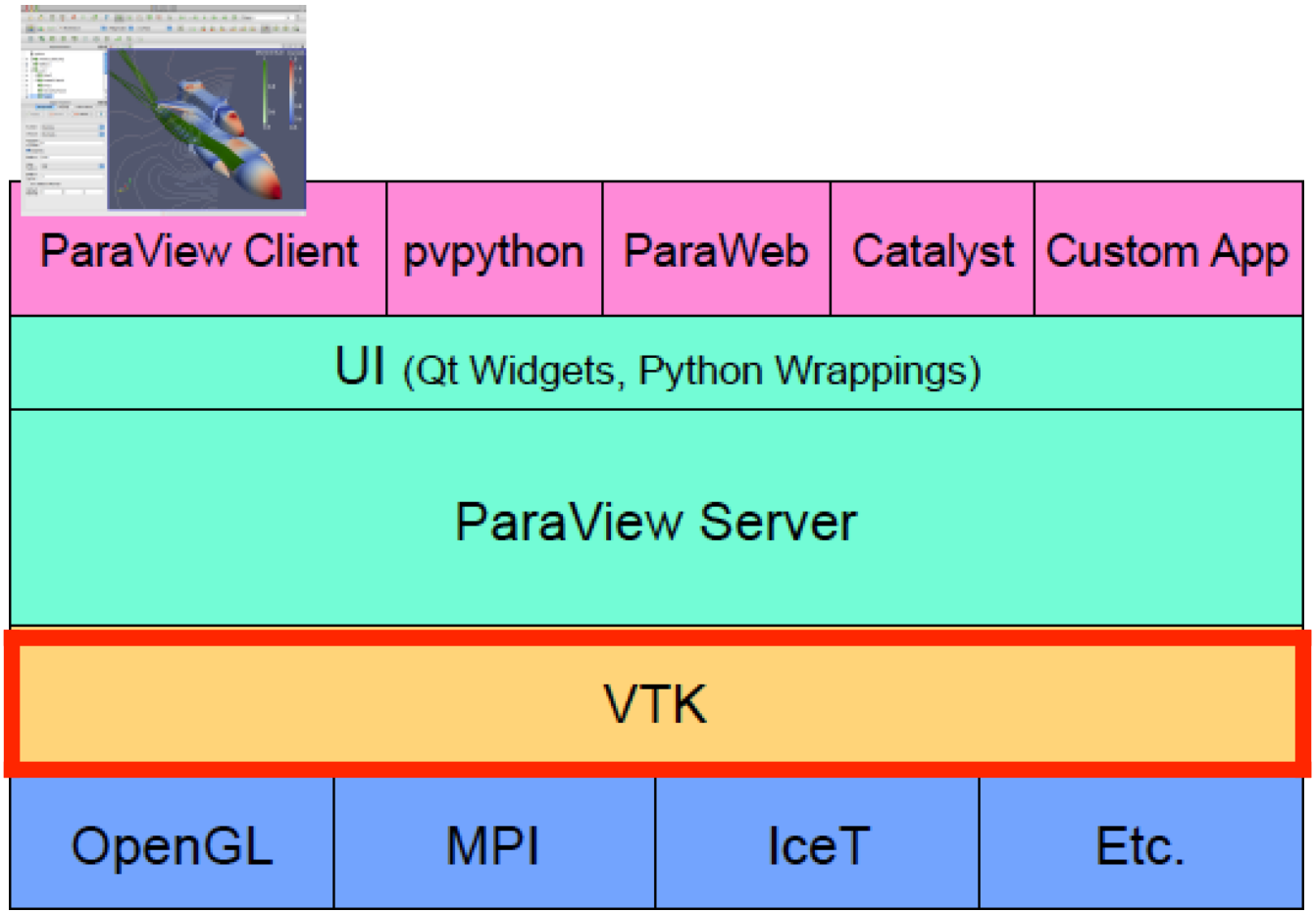




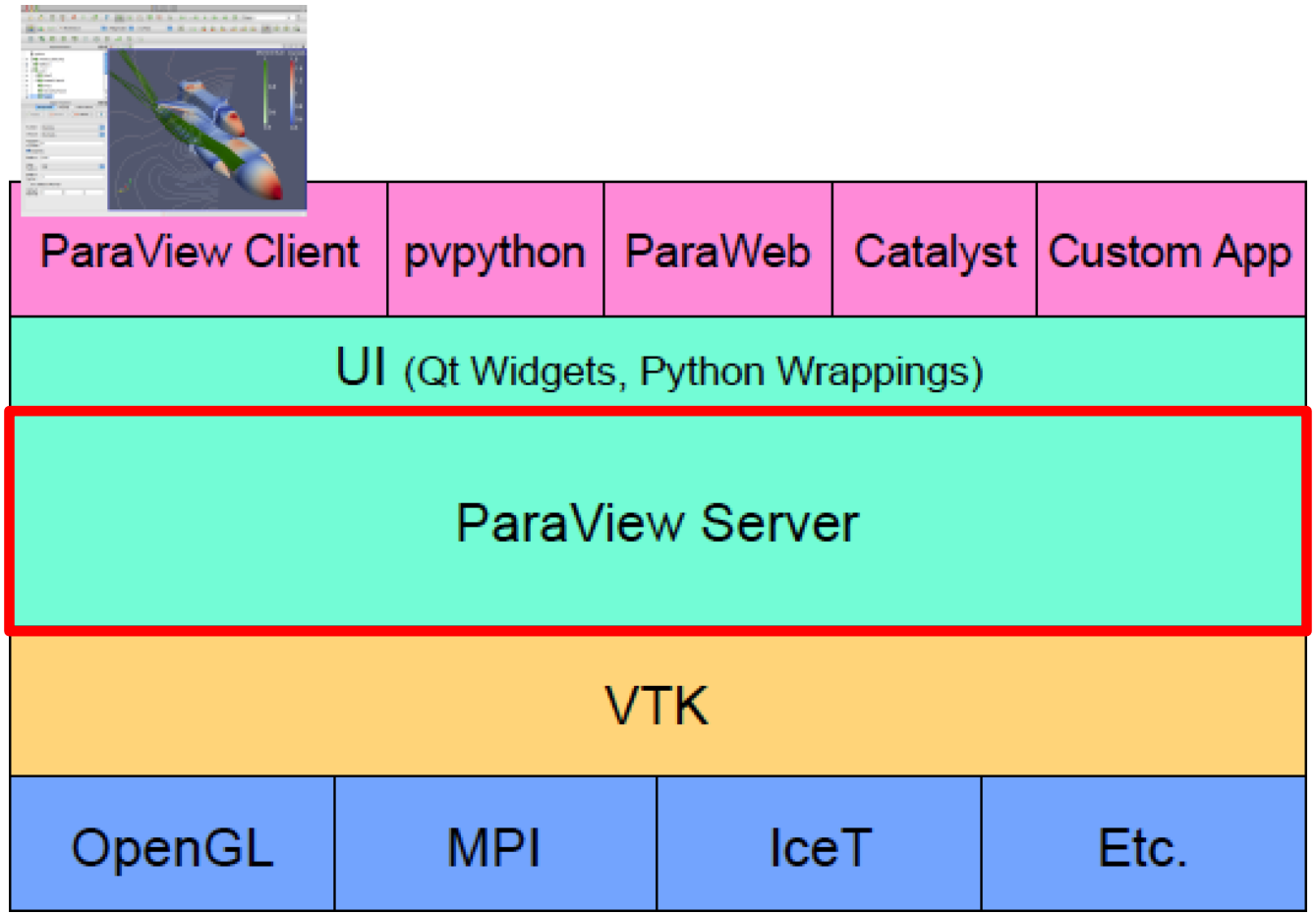
ParaView Software Stack



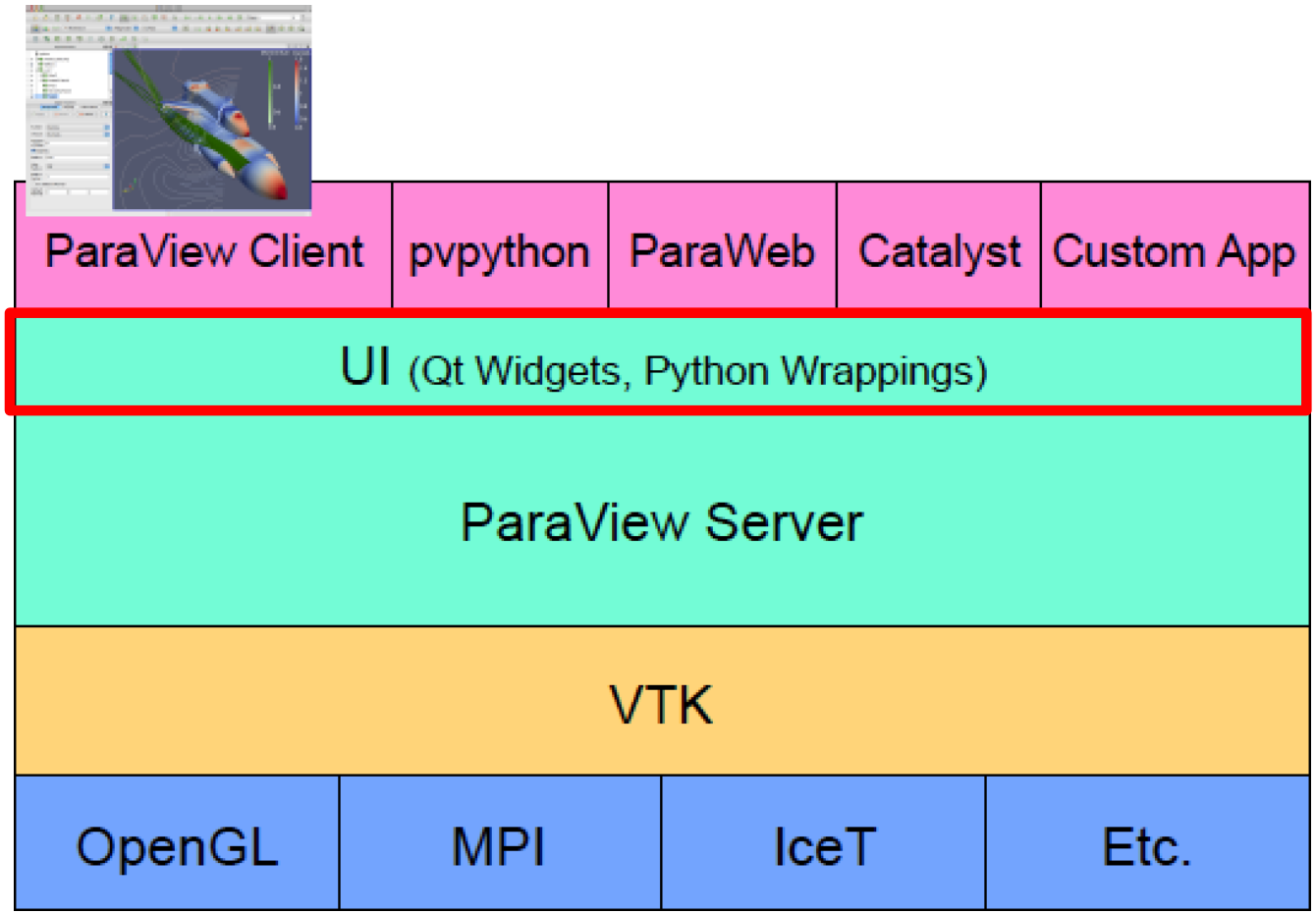
ParaView Software Stack



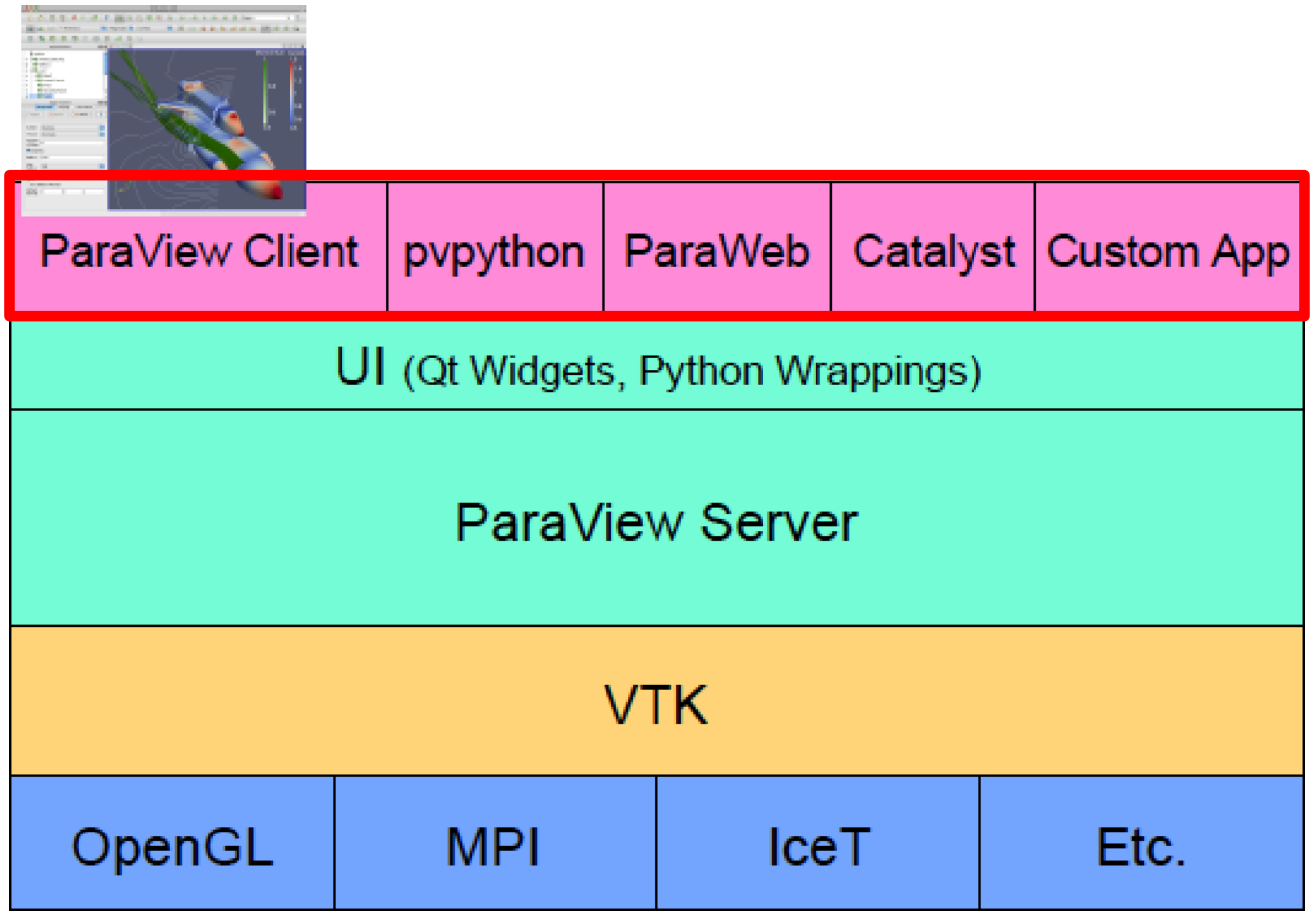
ParaView Software Stack



ParaView Software Stack

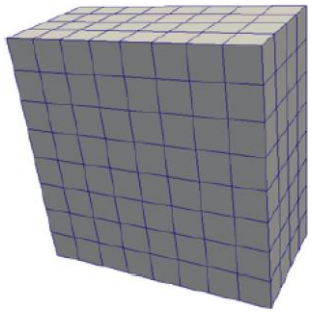


ParaView Software Stack

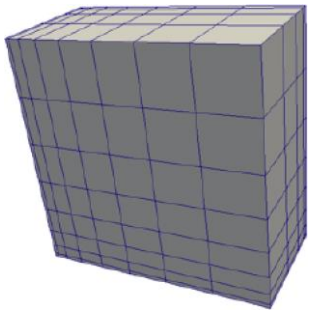


ParaView Data Model

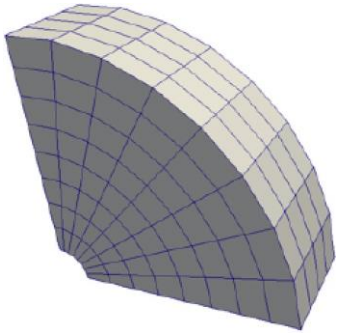
- ParaView can process the following types of spatial data



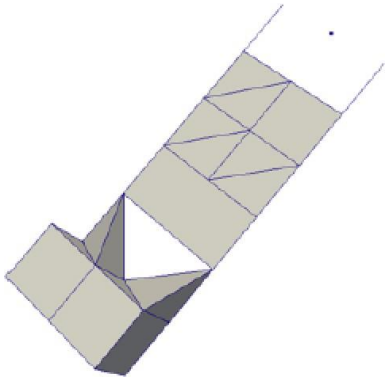
Uniform Rectilinear



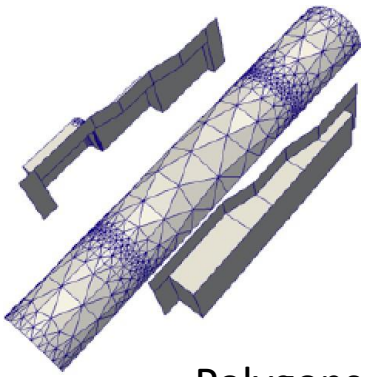
Non-uniform Rectilinear



Curvilinear

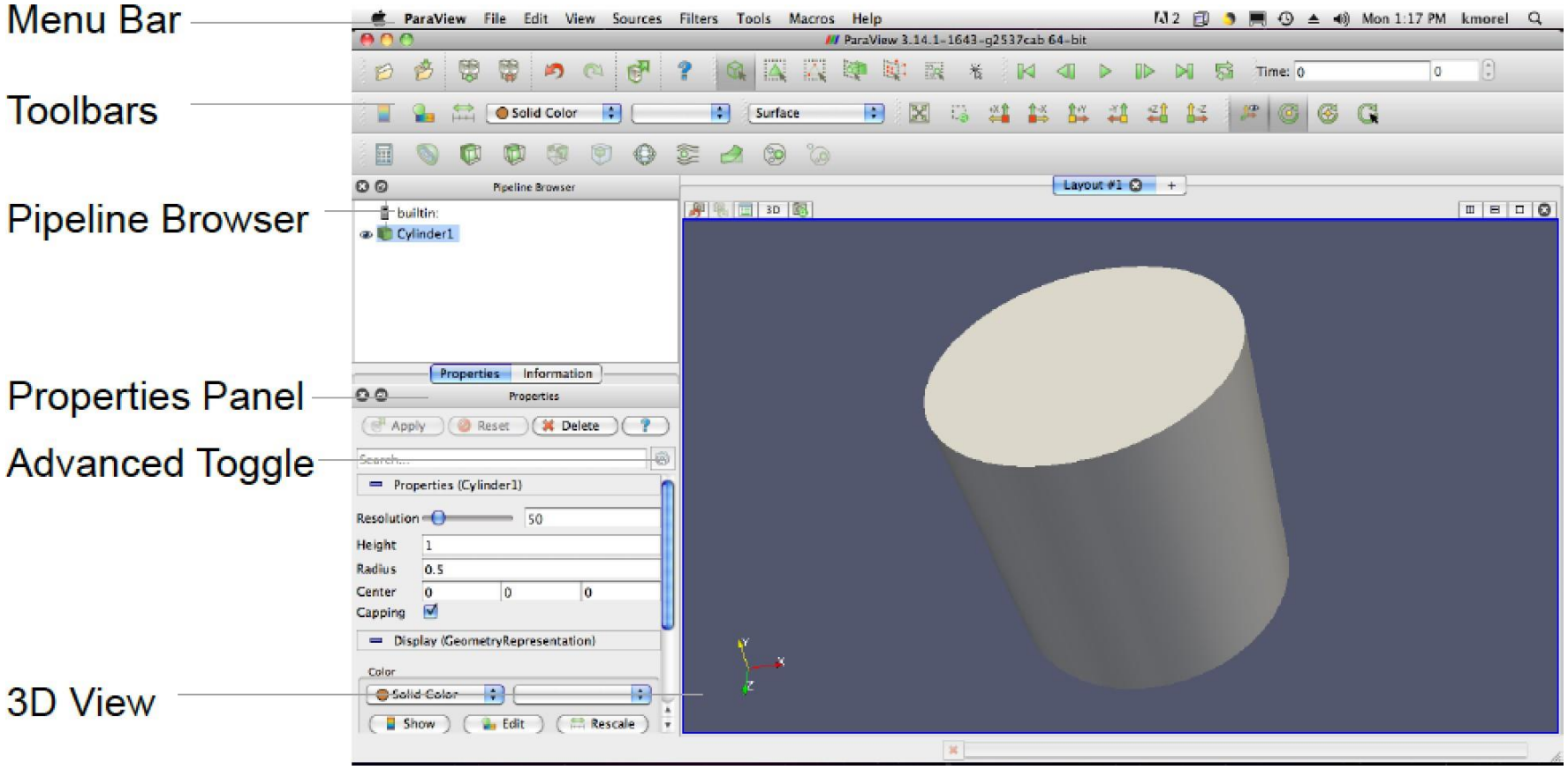


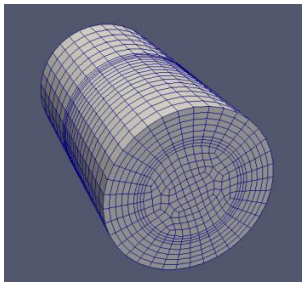
Unstructured Grid



Polygons

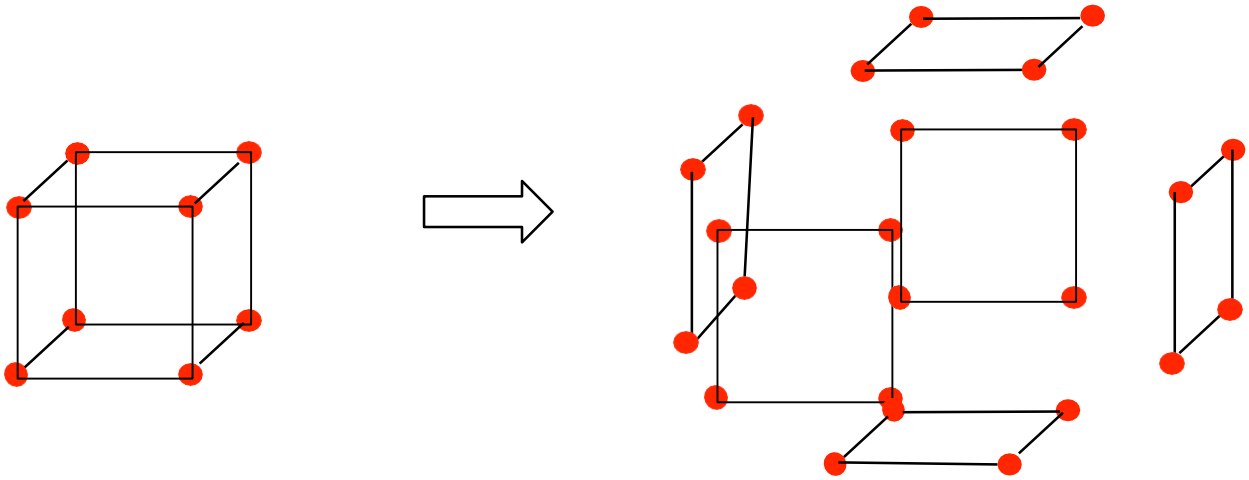
ParaView User Interface

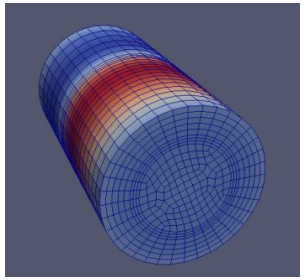




Mesh View

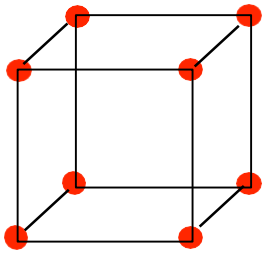
- Convert the faces of each cell in the data set into polygons
- Draw the face either in wireframe or surface (or both) mode using a preferred graphics library (such as OpenGL)



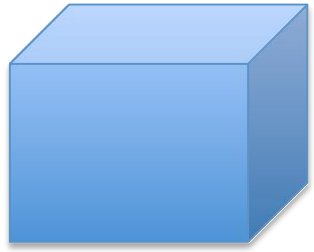


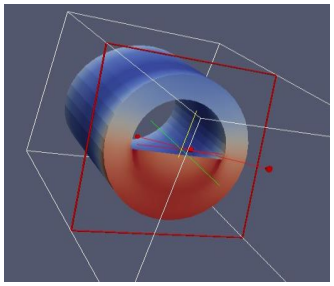
Mesh Surface with Colors

- Map the attribute values at the vertices of each cell to colors by a lookup table
- Draw the faces in surface mode with the color attributes using a preferred graphics library (such as OpenGL)
- Colors are interpolated across the surface



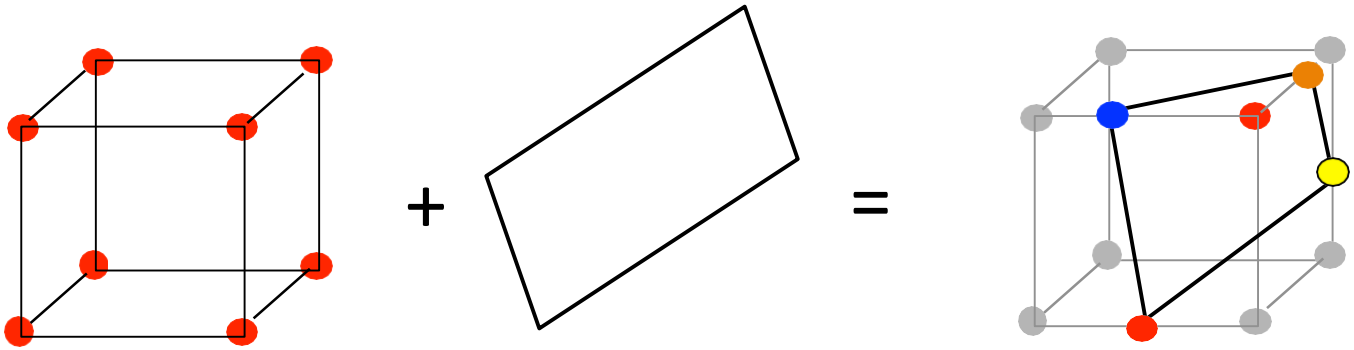
Value	R	G	B
0.00	0	0	1.0
0.05	0	0.1	0.9
0.10	0	0.3	0.7
…	…	…	…

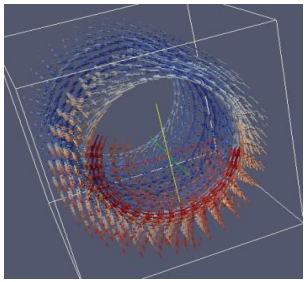




Data Slicing

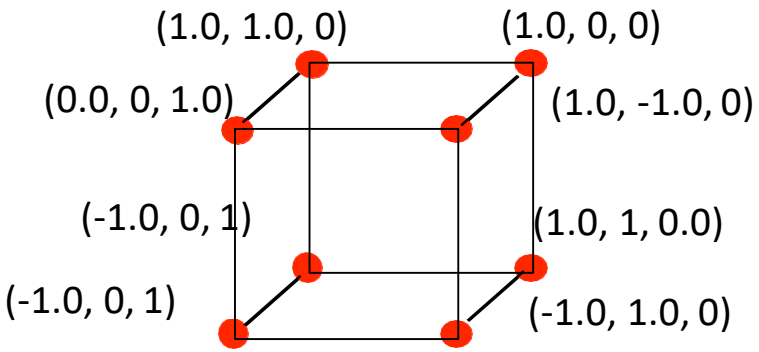
- Intersecting the mesh with a slicing surface (slicer)
- The slicer can be represented as an implicit function f(x,y,z) = 0
- A plane is typically used (Ax + By + Cz+ D = 0), but does not need to be
- Data attributes are sampled at the intersection points between the slicer and the mesh, and the resulting polygonal mesh is rendered

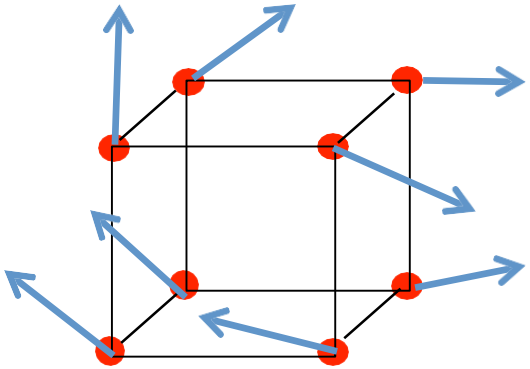


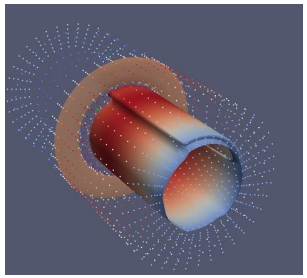


Glyphing

- Graphical objects shown at selected points (e.g. grid points) to display the data
  - Pros: Precise
  - Cons: extremely local, and can cause visual cluttering
- Example: arrows to depict vectors

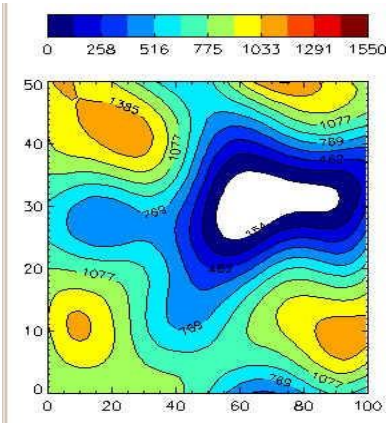


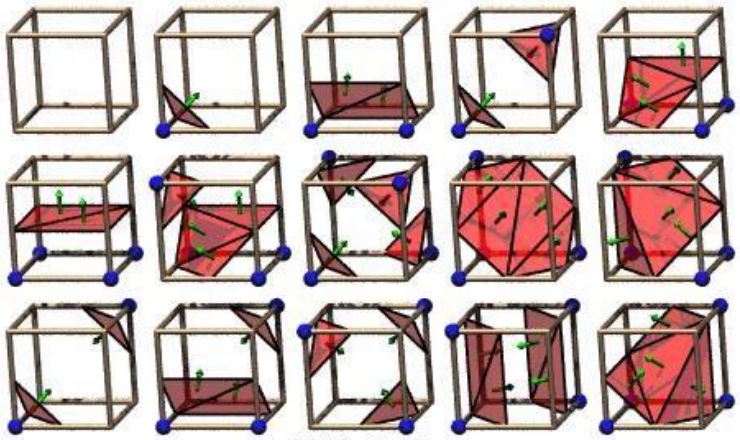


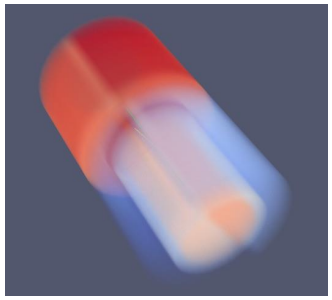


Contouring

- Show all the points whose attribute values equal to a constant; f(x,y,z) = C
- Contouring on a 2D surface: curves
- Contouring in a 3D volume: surfaces
- Discrete algorithms are needed to extract the contours (e.g. Marching Cubes)

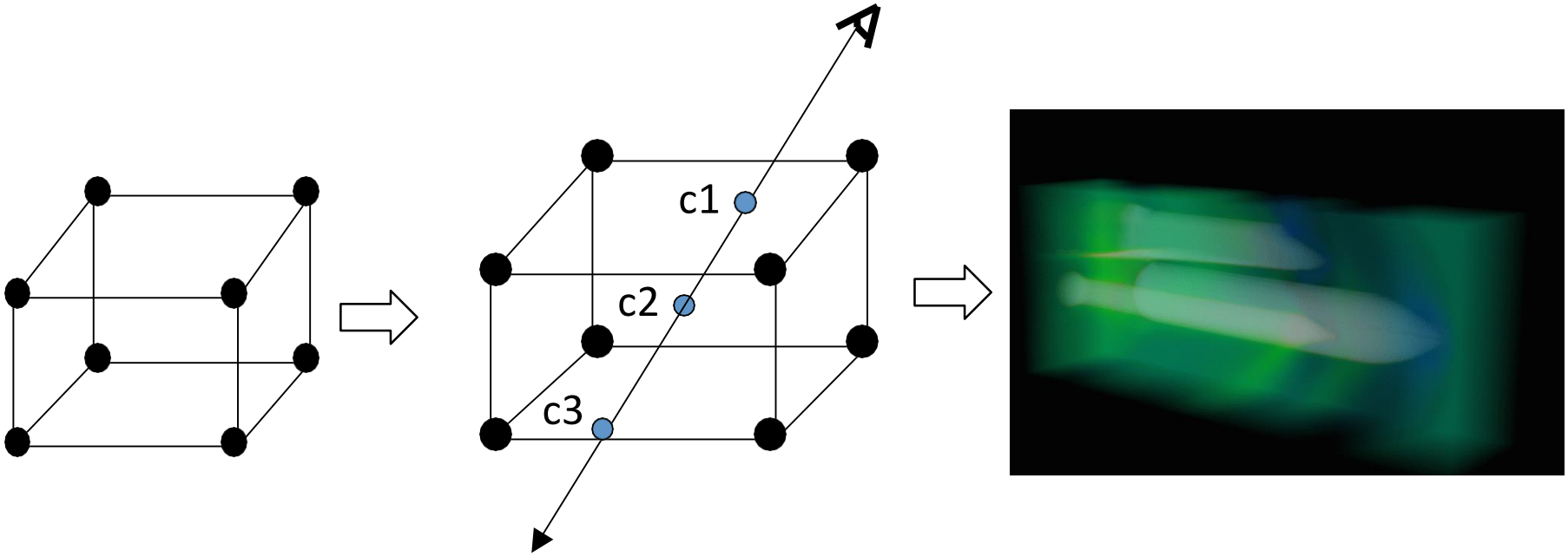



The 15 Cube Combinations



Volume Rendering

- A method to visualize the entire 3D data set by simulating light transport across the volume
- A 2D projection of 3D discrete samples



ParaView Video Demo