Big Data Visual Analytics (CS 661)

Instructor: Soumya Dutta

Department of Computer Science and Engineering

Indian Institute of Technology Kanpur (IITK)

email: soumyad@cse.iitk.ac.in

Acknowledgements

- Some of the following slides are adapted from the excellent course materials and tutorials made available by:
  - Prof. Han-Wei Shen (The Ohio State University)
  - Prof. Klaus Mueller (State University of New York at Stony Brook)
  - David DeMarle (Intel)

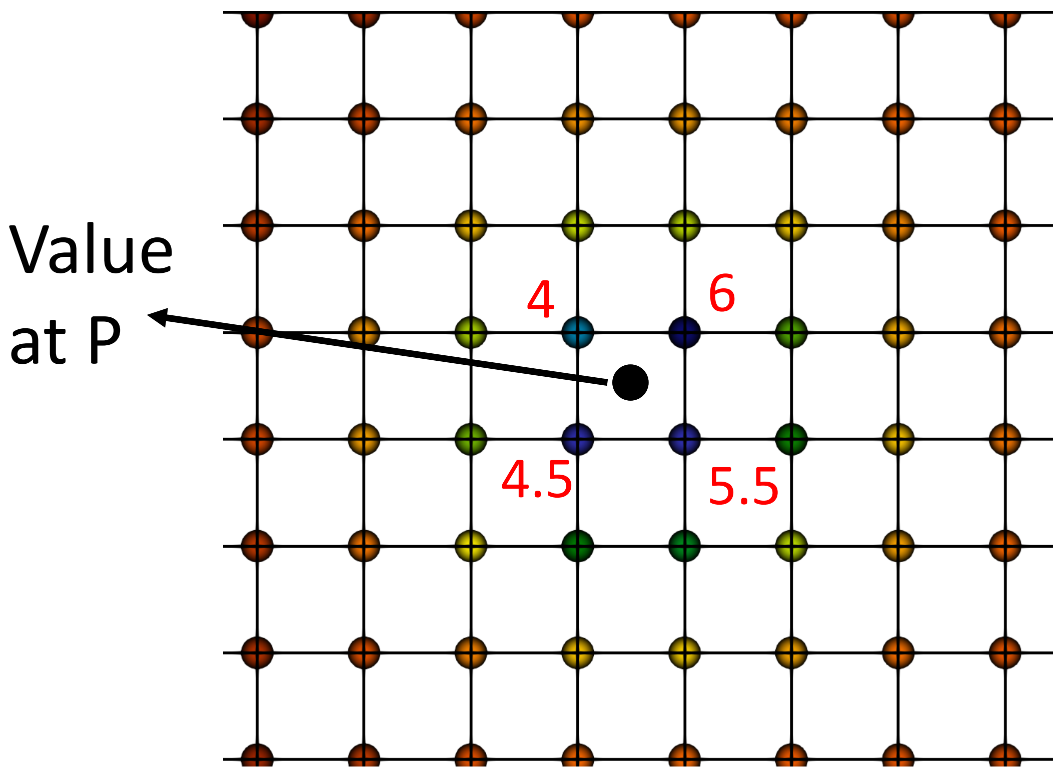
Study Materials

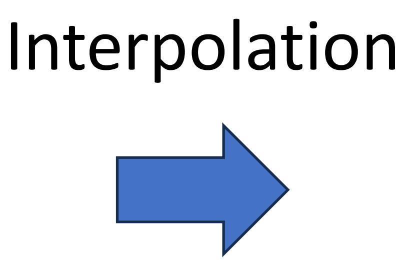
- The Visualization Toolkit by Will Schroeder, Ken Martin, Bill Lorensen
  - Chapter 1, Chapter 4, Chapter 5
  - Get the pdf: https://vtk.org/vtk-textbook/
- Reference for learning VTK:
  - VTK User's Guide
  - Get the pdf: https://vtk.org/vtk-users-guide/
  - Examples: https://kitware.github.io/vtk-examples/site/Python/
- ParaView Tutorial:
  - https://www.youtube.com/watch?v=sXY72e3Ce4g&list=PLGj2a3KTwhRZ7XupM7f36czTGlJvqq7_N&index=3

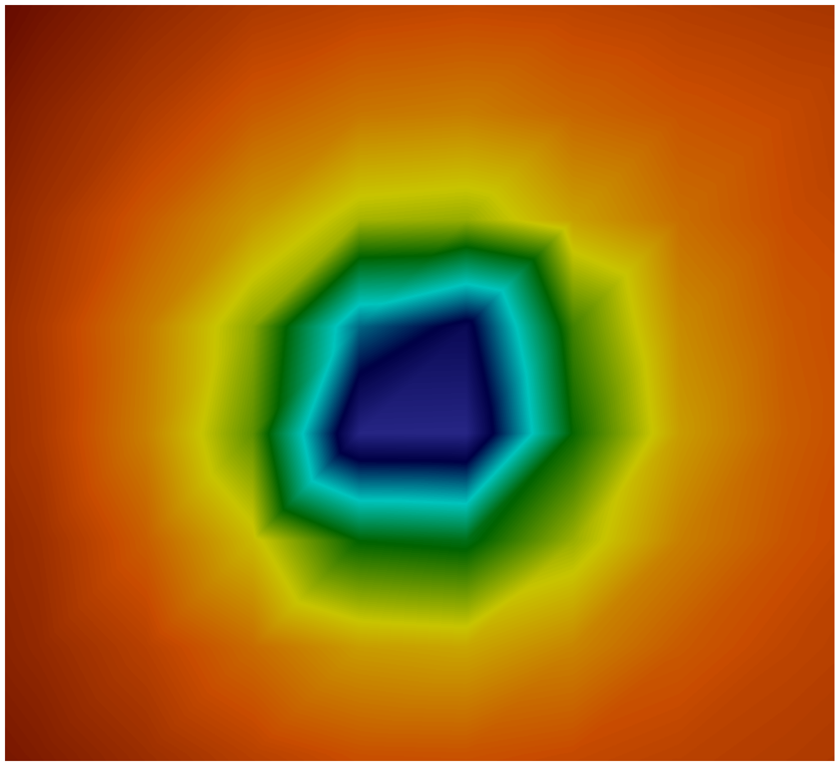
Linear Interpolation for Scientific Data

Why Interpolation?

- All visualization algorithms must deal with discrete data
- What if an algorithm require value at a point which is inside a cell?

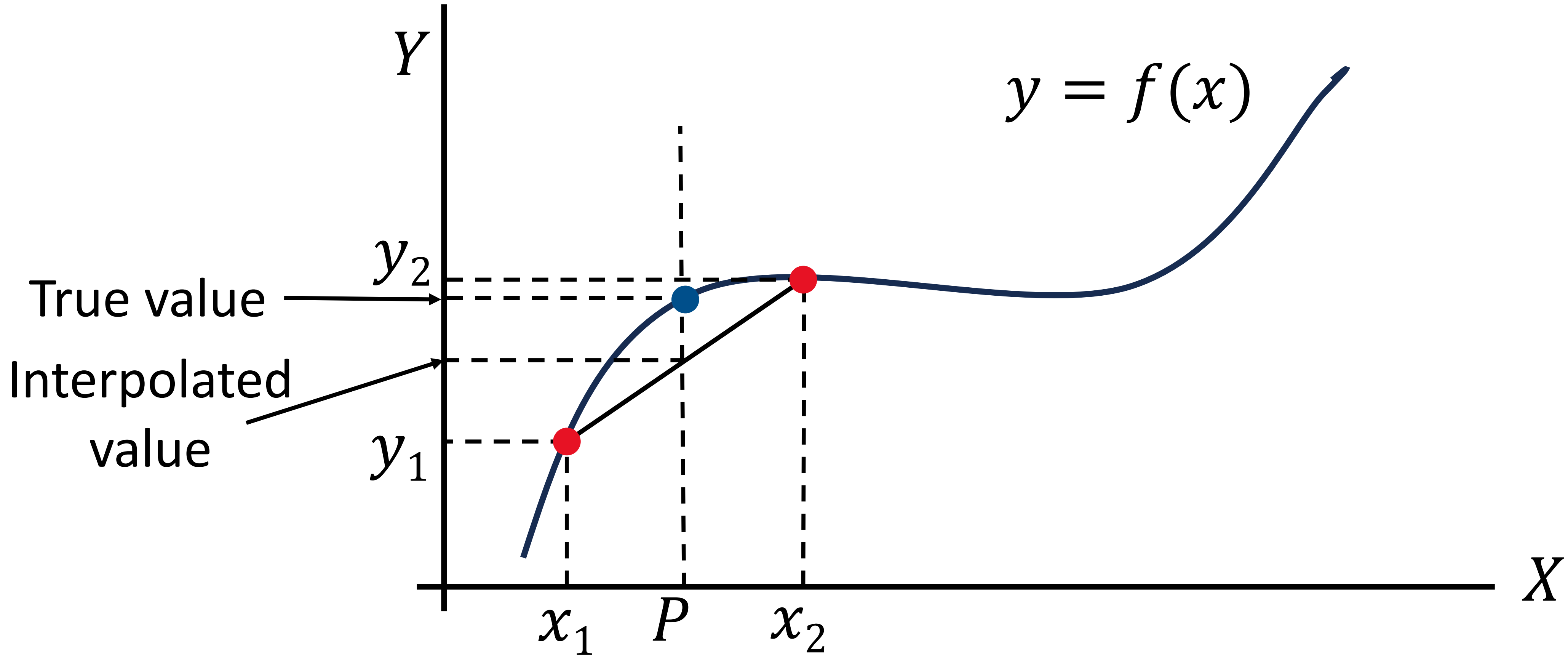






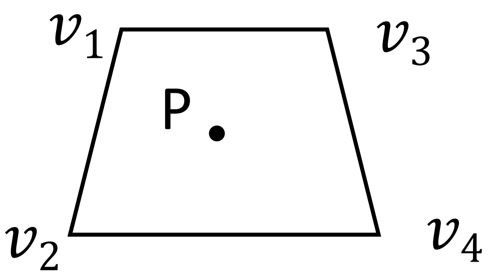
Linear Interpolation (LERP)

- Linear Interpolation (lerp): connecting two points with a straight line in the function plot



Linear Interpolation (LERP)

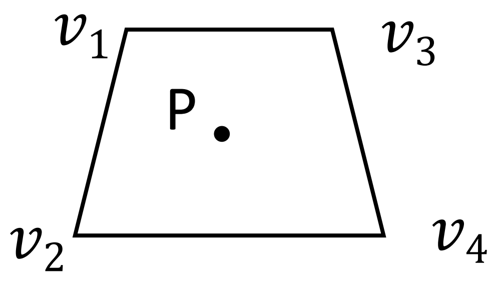
- General form: $V_p = \sum w_i * vi$ (weighted sum)



v_i : value at vertex i
w_i : weight for v_i

Linear Interpolation (LERP)

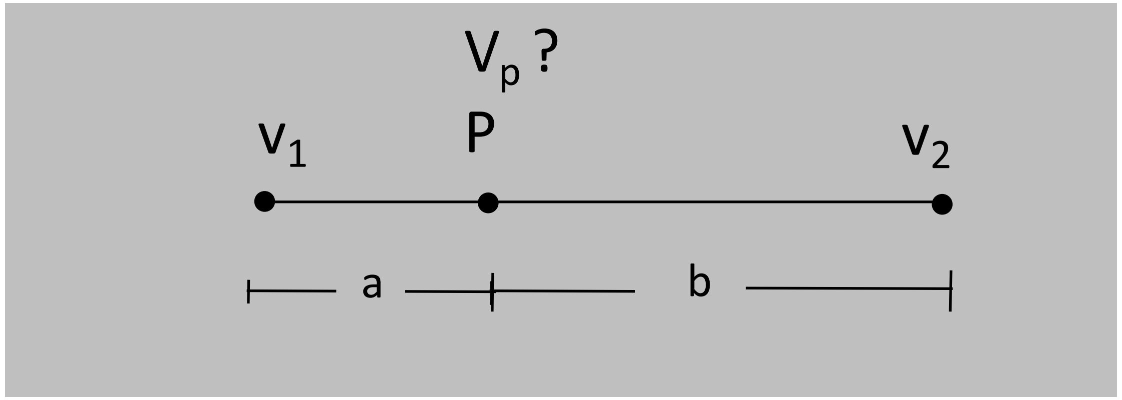
- General form: $V_p = \sum w_i * vi$ (weighted sum)



v_i : value at vertex i
w_i : weight for v_i

- Essential information needed:
  - Cell type
  - Data value at cell corners
  - Parametric coordinates of the point in question (P)
    - Related to the position of point P in the cell

LERP in Line

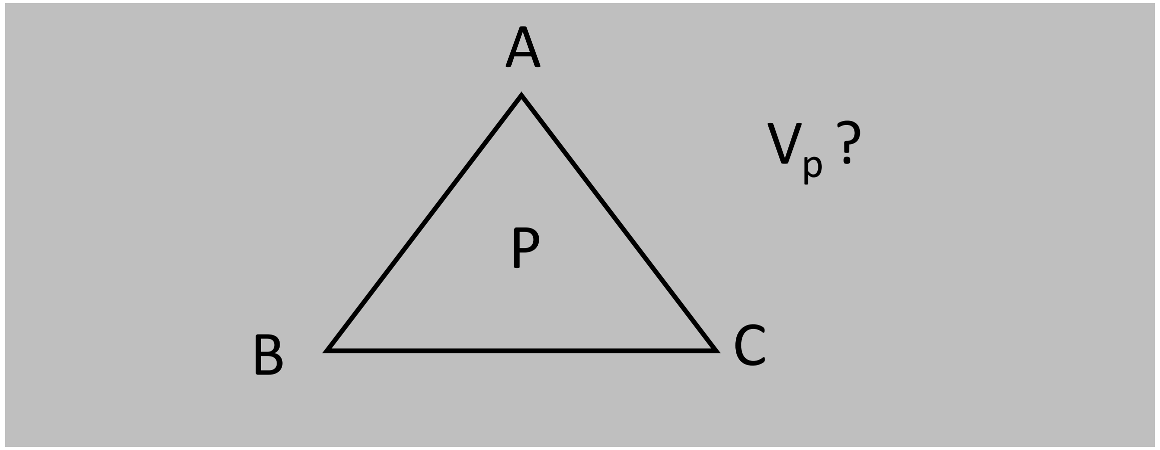


- Parametric coordinate of P: $\alpha = a / (a+b)$
- Linearly interpolated value of P:

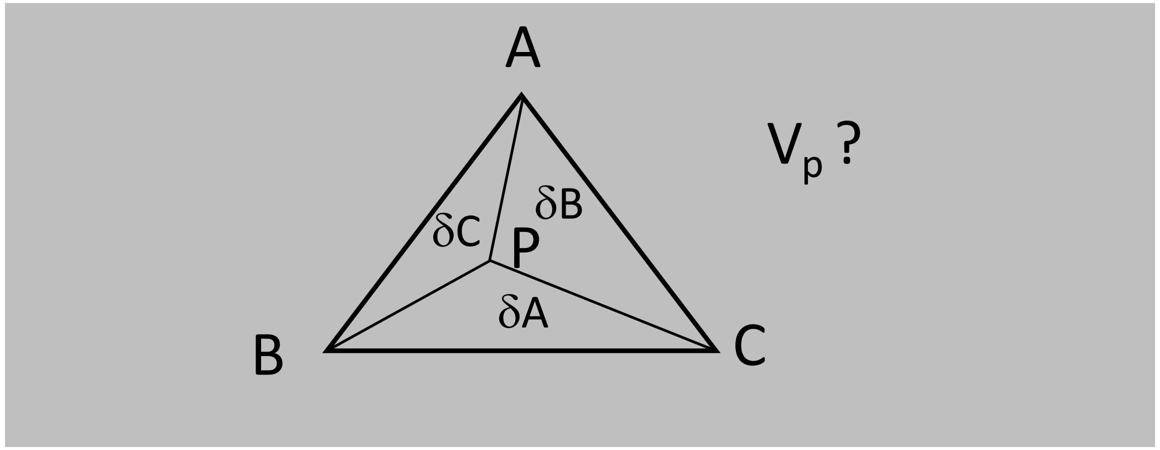
lerp(v1,v2, α)

$$\boxed{V_p = (1-\alpha) * V_1 + \alpha * V_2}$$

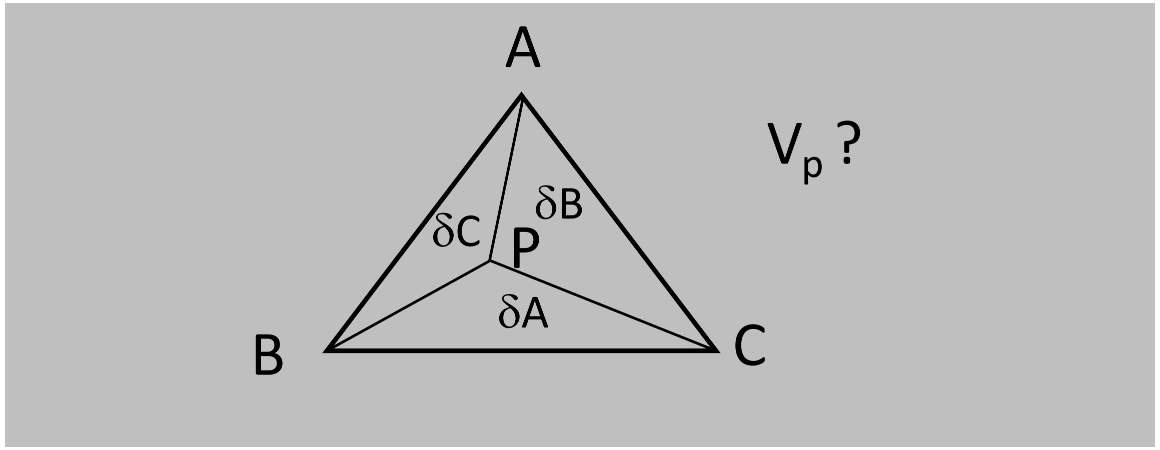
Lerp in Triangle



Lerp in Triangle



Lerp in Triangle



- Parametric coordinates of P: (α,β,γ)

$\alpha = \delta A \,/\, (\delta A + \delta B + \delta C)$

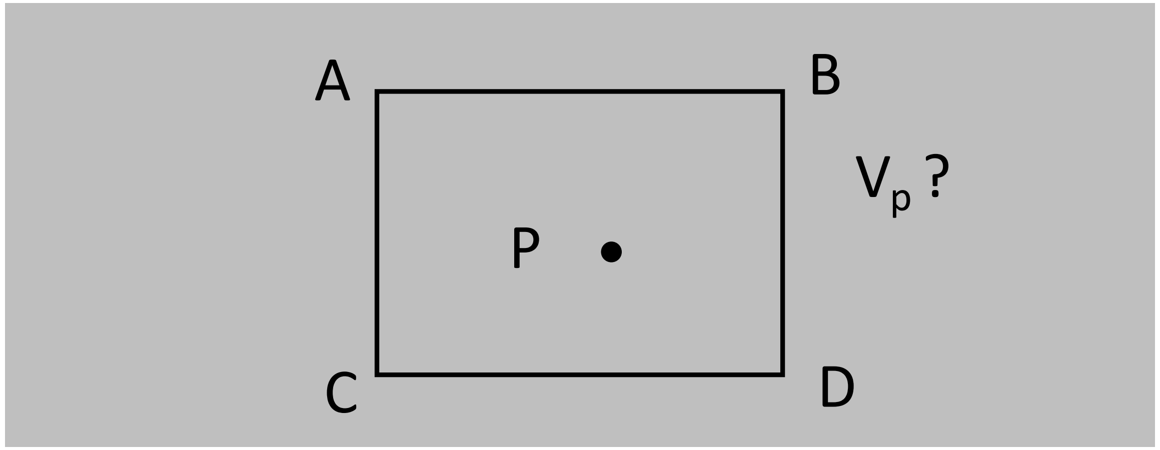
$\beta = \delta B \,/\, (\delta A + \delta B + \delta C)$

$\gamma = \delta C \,/\, (\delta A + \delta B + \delta C)$

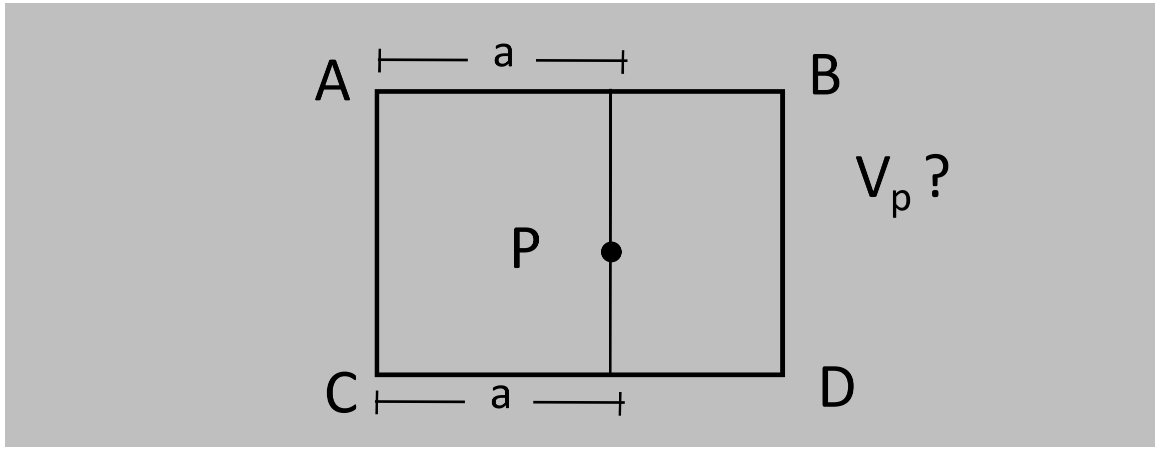
Baricentric Coordinates

- Linearly interpolated value of P: $\boxed{V_A * \alpha + V_B * \beta + V_C * \gamma}$

Lerp in Rectangle



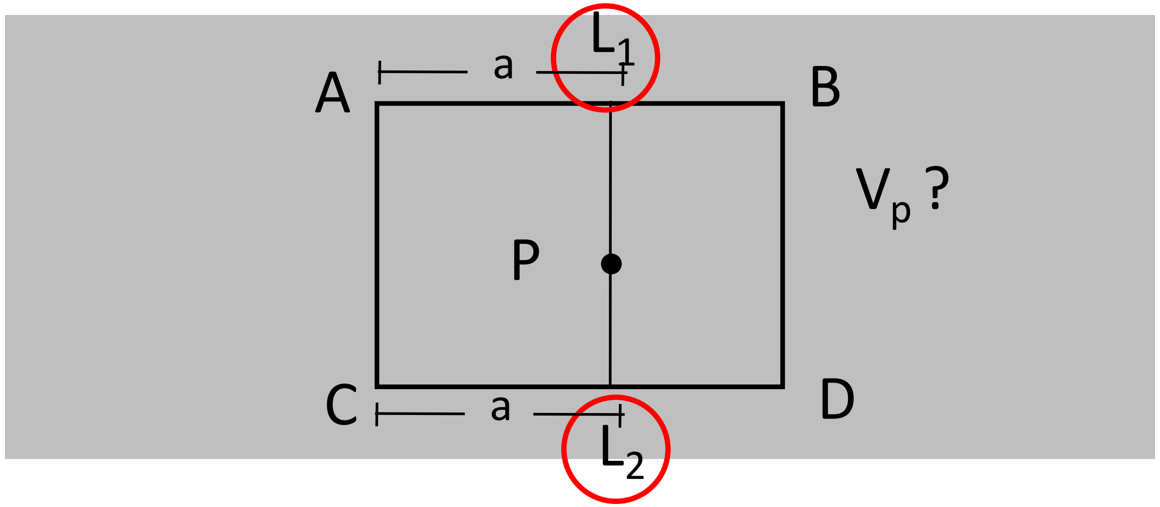
Lerp in Rectangle



- Parametric coordinates of P: (α,β)

$$\alpha = a / \text{width};$$

Lerp in Rectangle

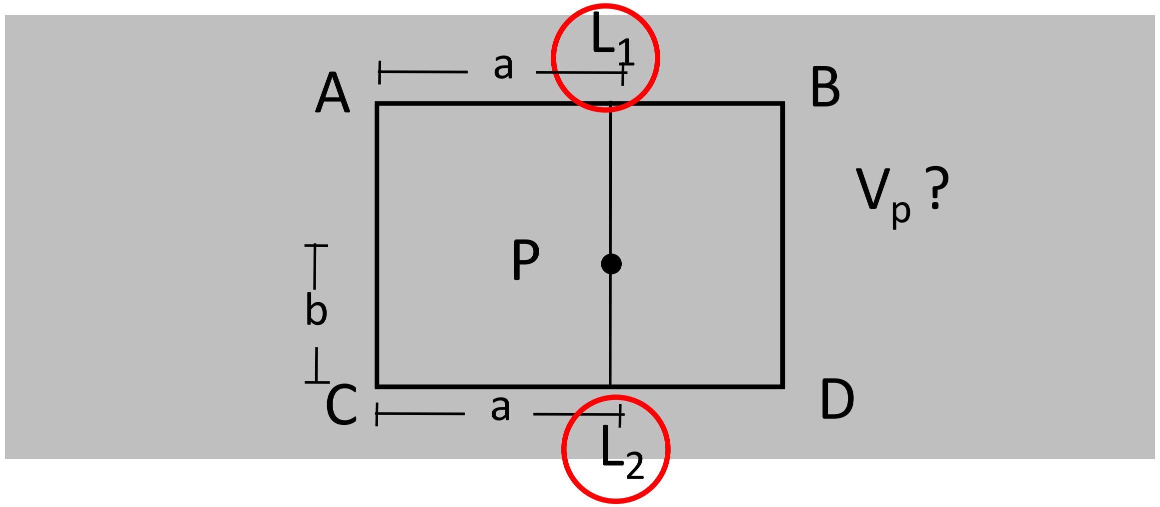


- Parametric coordinates of P: (α,β)

$$\alpha = a / \text{width};$$

- Value at L_1 = $\text{Lerp}(V_A, V_B, \alpha)$;
- Value at L_2 = $\text{Lerp}(V_C, V_D, \alpha)$;

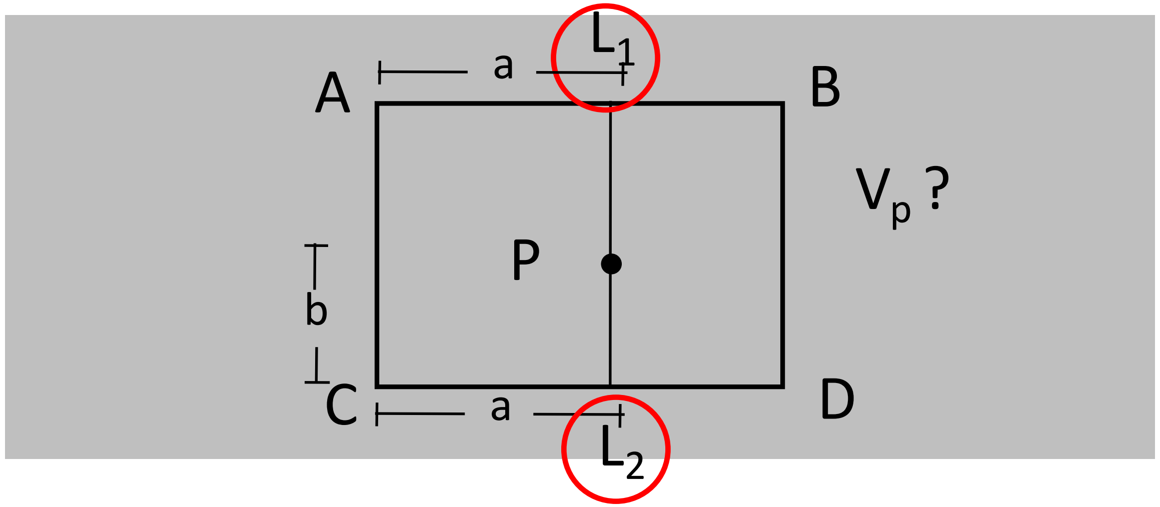
Lerp in Rectangle



- Parametric coordinates of P: (α,β)

$$\alpha = a \,/\, \text{width};$$

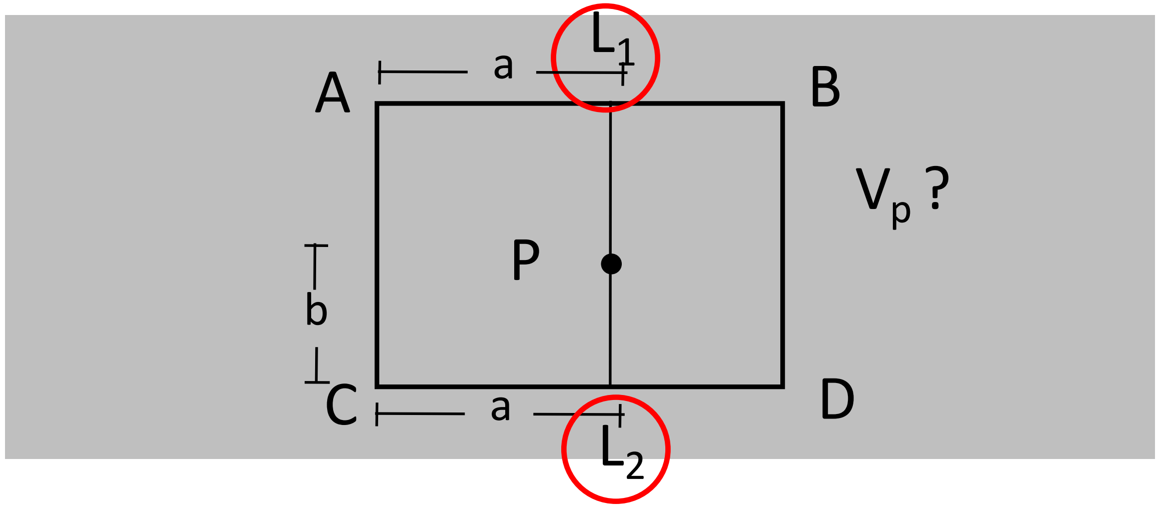
Lerp in Rectangle



- Parametric coordinates of P: (α, β)

$$\alpha = a / \text{width}; \; \beta = b / \text{height}$$

Lerp in Rectangle

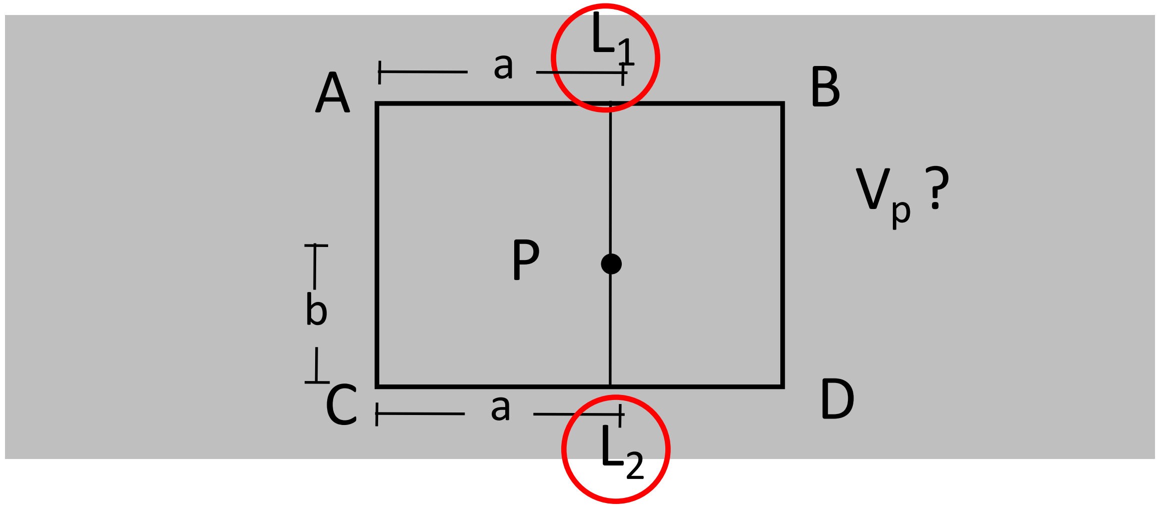


- Parametric coordinates of P: (α, β)

  $\alpha =$ a / width; β = b / height

- Linearly interpolated value of P: $\boxed{\text{Lerp}(V_{L2}, V_{L1}, \beta)}$

Lerp in Rectangle



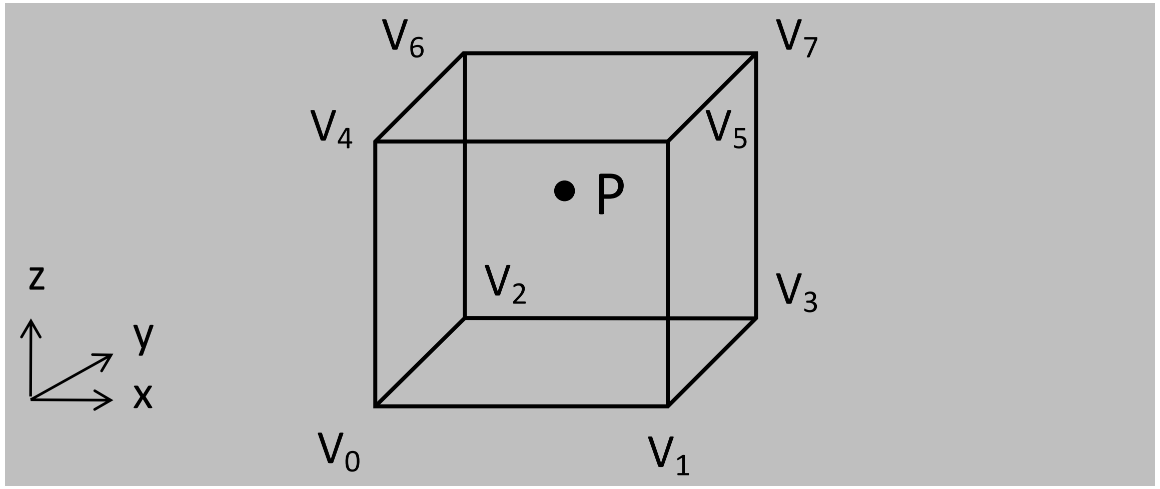
- Parametric coordinates of P: (α, β)

$$\alpha = a / \text{width}; \ \beta = b / \text{height}$$

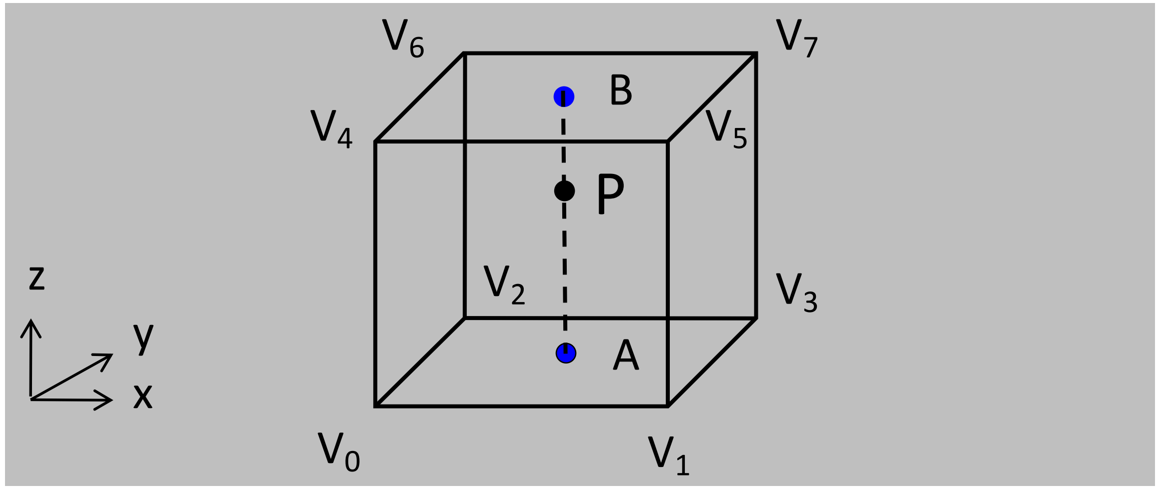
Bi-linear interpolation
Bi-Lerp(V_A,V_B, V_C,V_D)

- Linearly interpolated value of P: $\boxed{\text{Lerp}(V_{L2}, V_{L1}, \beta)}$

Lerp in Cube

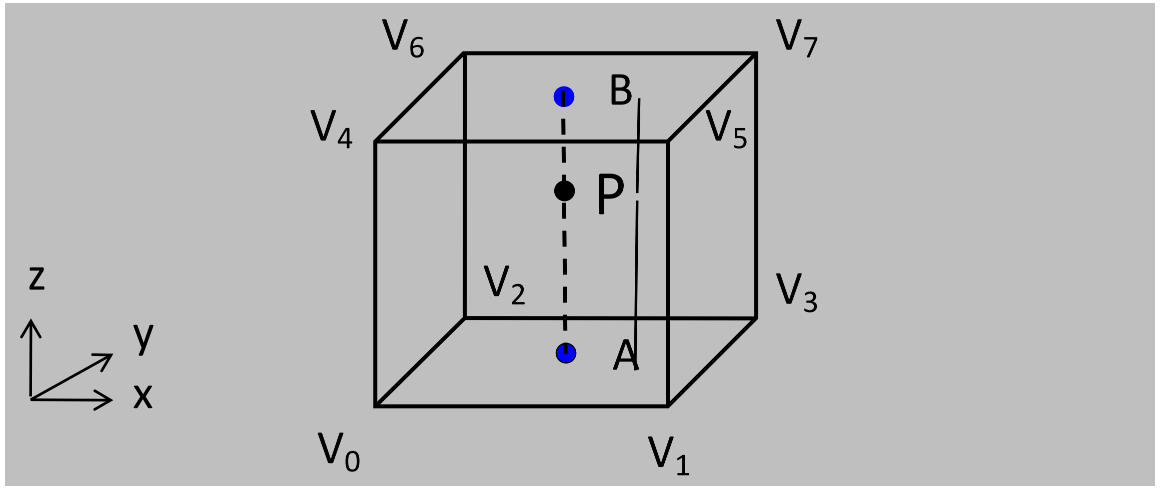


Lerp in Cube



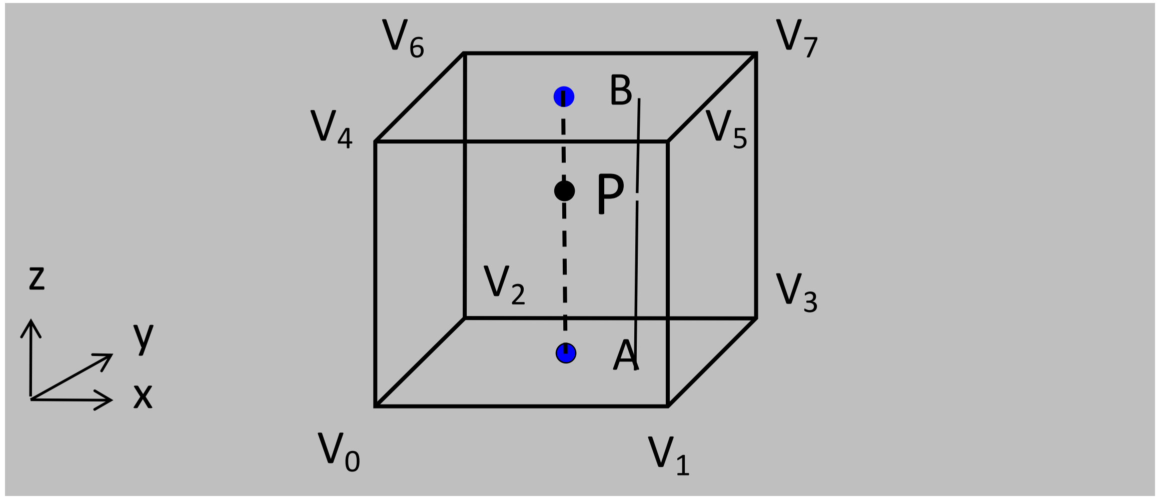
- Value at A = Bi-Lerp(V_0,V_1,V_2,V_3) ;
- Value at B = Bi-Lerp(V_4,V_5,V_6,V_7) ;

Lerp in Cube



- Value at A = Bi-Lerp(V_0,V_1,V_2,V_3) ;
- Value at B = Bi-Lerp(V_4,V_5,V_6,V_7) ;

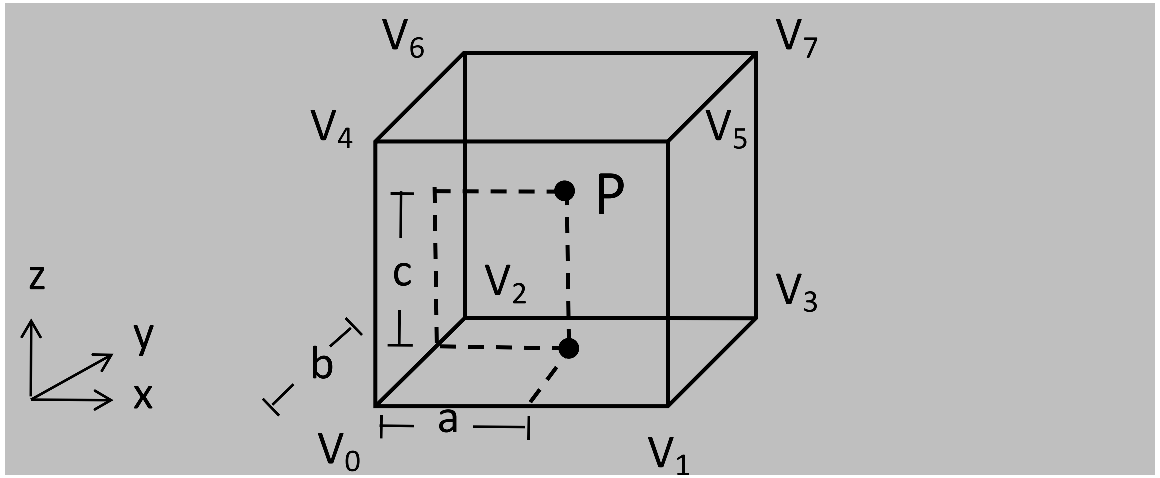
Lerp in Cube



- Value at A = Bi-Lerp(V_0,V_1,V_2,V_3) ;
- Value at B = Bi-Lerp(V_4,V_5,V_6,V_7) ;
- Value at P = Lerp(A,B, PA/AB); ⟵

Tri-linear interpolation

Lerp in Cube: Another Way



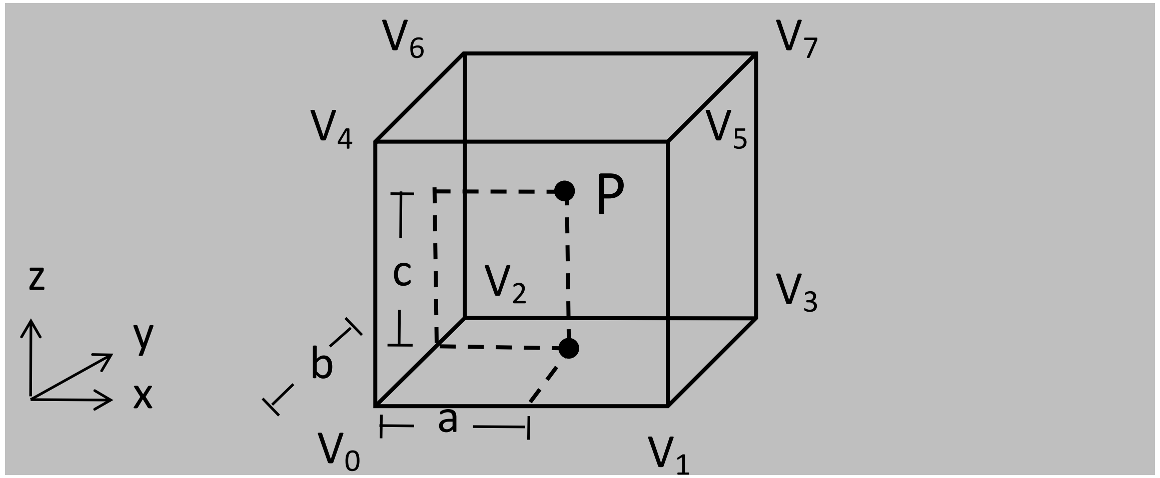
- Parametric coordinates of P: (α,β,γ)

  $\alpha =$ a / width

  $\beta =$ b / depth

  $\gamma =$ c / height

Lerp in Cube: Another Way



- Parametric coordinates of P: (α,β,γ)

  $\alpha =$ a / width

  $\beta =$ b / depth

  $\gamma =$ c / height

- Value at P =

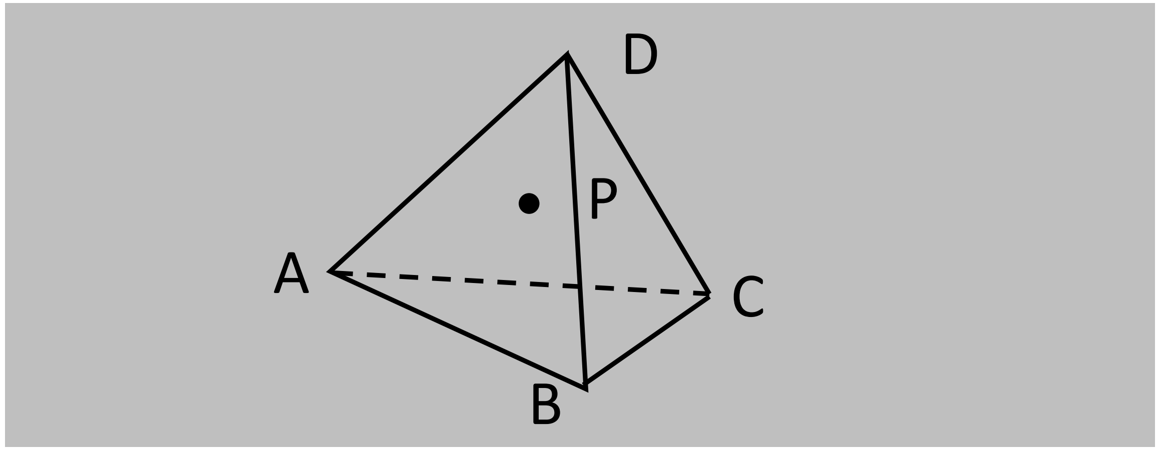
  $(1-\alpha)(1-\beta)(1-\gamma)V_0 + \alpha(1-\beta)(1-\gamma)V_1 +$

  $(1-\alpha)\beta(1-\gamma)V_2 + \alpha\beta(1-\gamma)V_3 +$

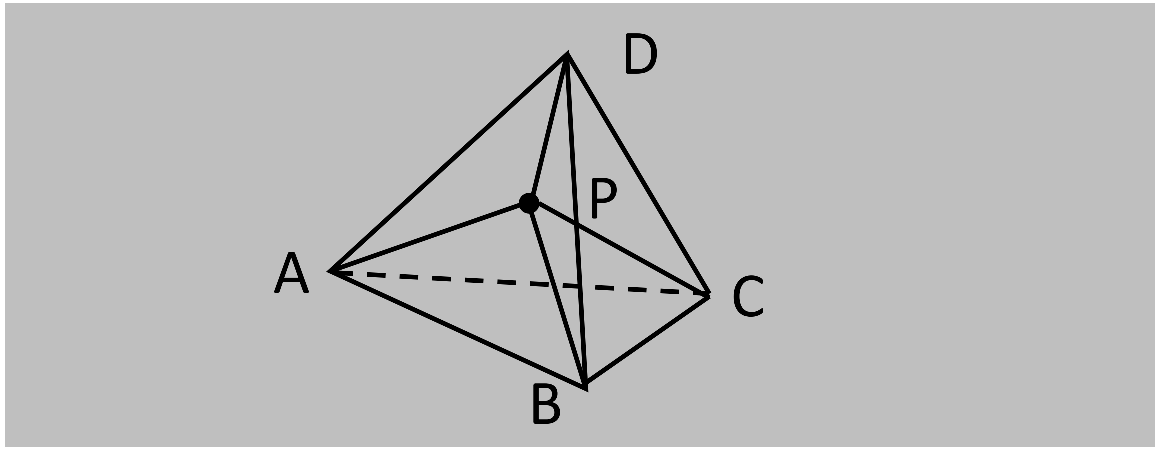
  $(1-\alpha)(1-\beta)\gamma V_4 + \alpha(1-\beta)\gamma V_5 +$

  $(1-\alpha)\beta\gamma V_6 + \alpha\beta\gamma V_7$

Lerp in Tetrahedron

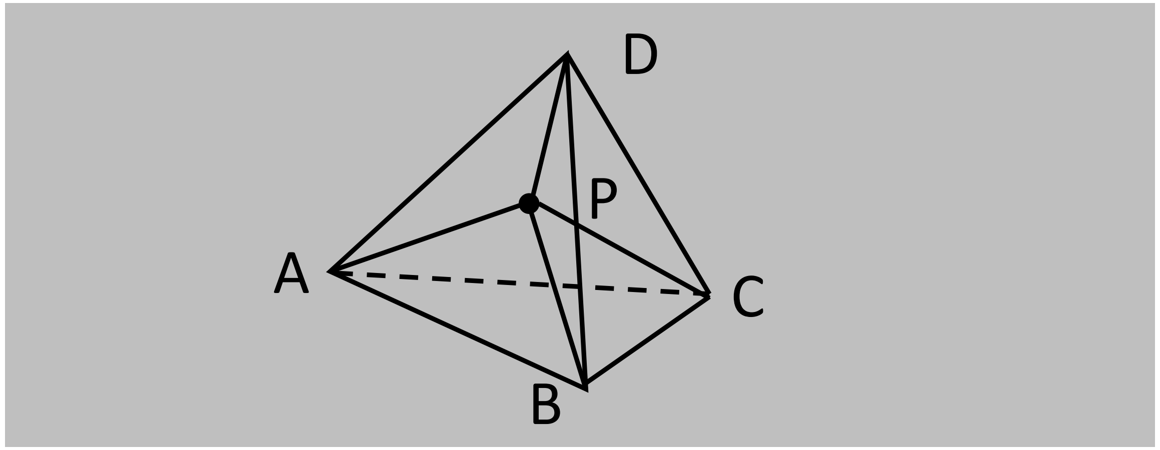


Lerp in Tetrahedron



- Break the tetrahedron ABCD into four sub tetrahedra: ABCP, BDCP, ACDP, ADBP
- Calculate the volume of each small tetrahedra
- Calculate P's parametric (tetrahedral) coordinates based on the ratios of the sub-volumes

Lerp in Tetrahedron



- Tetrahedral coordinates of P: $(\alpha,\beta,\gamma,\delta)$

$$\alpha = V_{BDCP} / V_{ABCD}$$

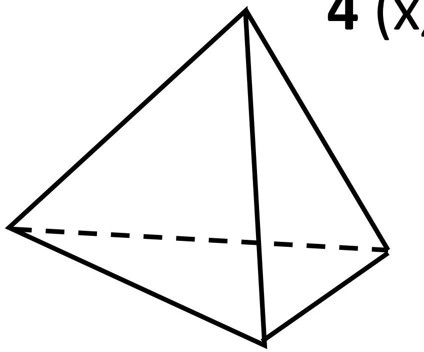
$$\beta = V_{ACDP} / V_{ABCD}$$

$$\gamma = V_{ADBP} / V_{ABCD}$$

$$\delta = V_{ABCP} / V_{ABCD}$$

- Linearly interpolated value of P: $\boxed{V_A * \alpha + V_B * \beta + V_C * \gamma + V_D * \delta}$

Volume of Tetrahedron



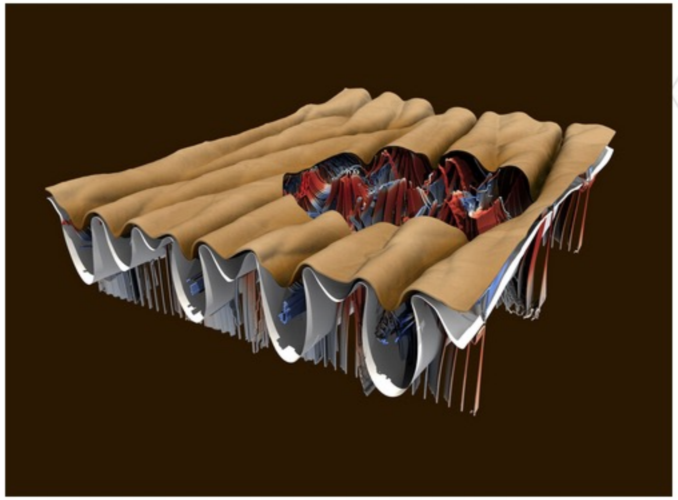
$$V = \frac{1}{6}\det\begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{bmatrix} = \frac{1}{6}\det(\mathbf{J})$$

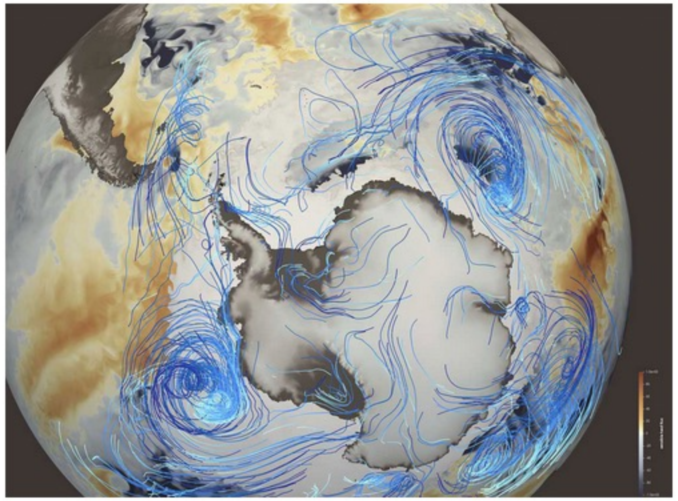
V will be positive if when you look at the triangle 123 from vertex 4, vertex 1 2 3 are in a counterclockwise order

Data Visualization with ParaView

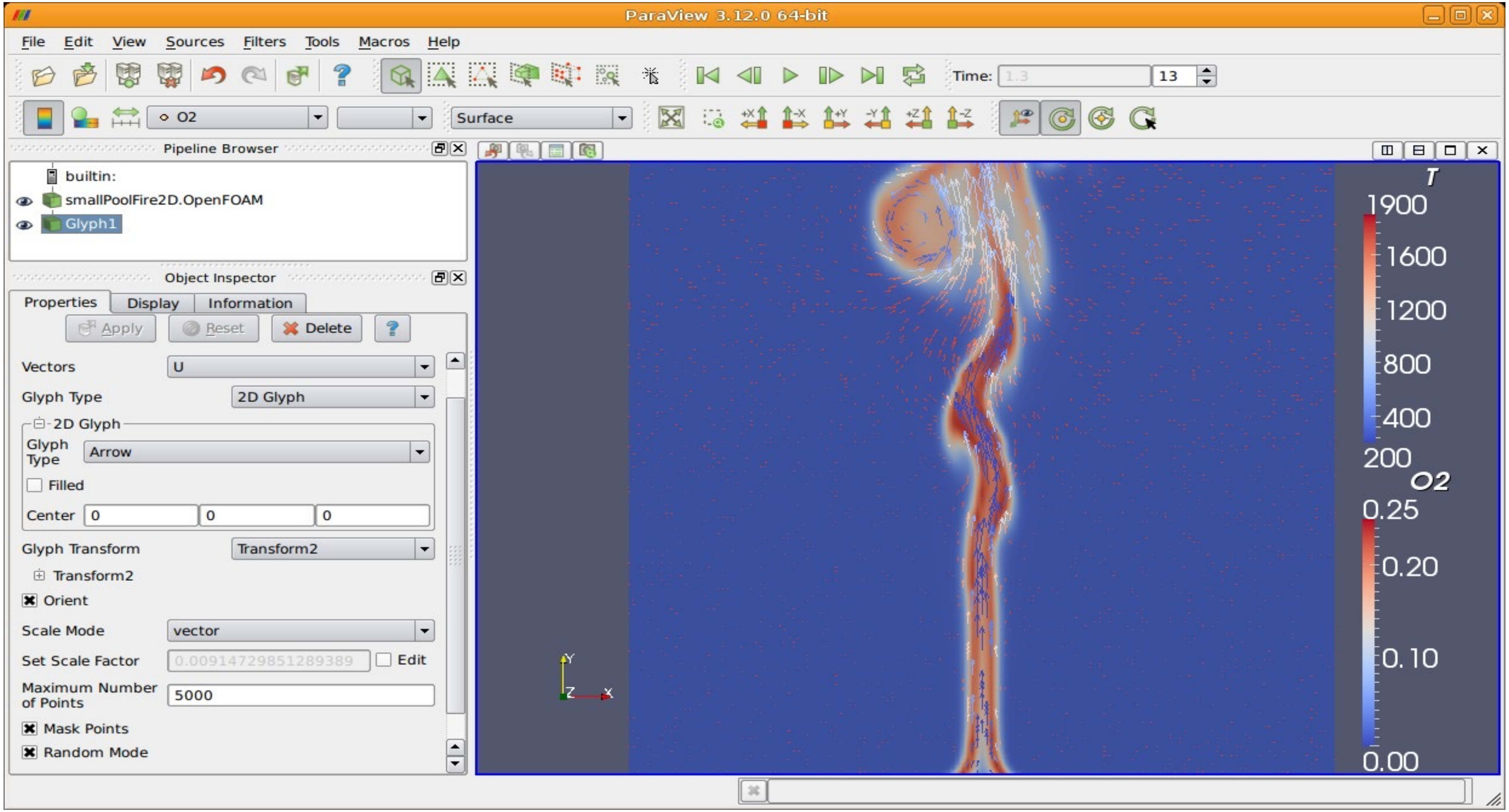
What is ParaView

- An open-source, scalable, multi-platform visualization application
- Support for distributed computation models to process large data sets
- An open, flexible, and intuitive user interface
- An extensible, modular architecture based on open standards
- A flexible BSD 3 Clause license
- Commercial maintenance and support

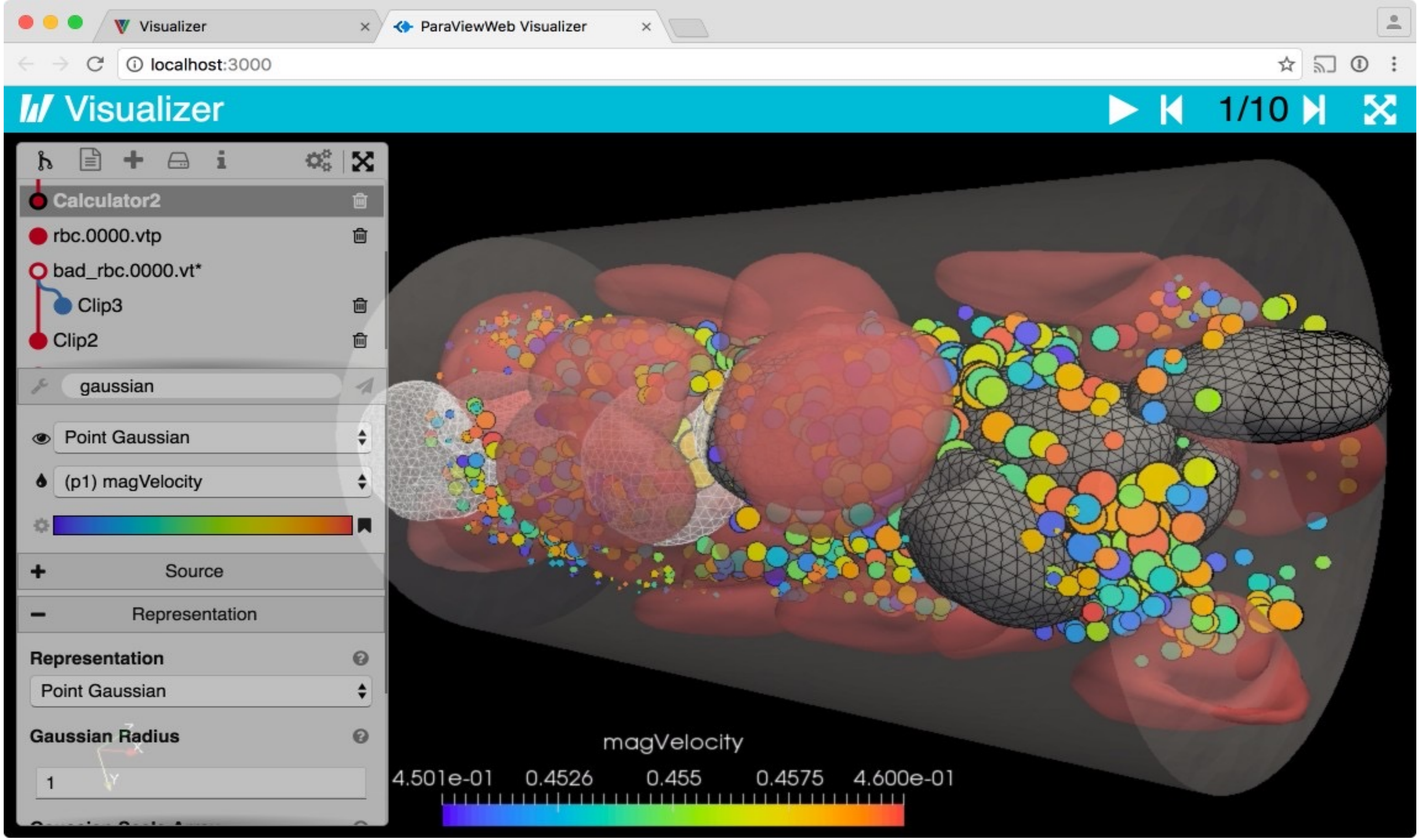




ParaView on the Desktop



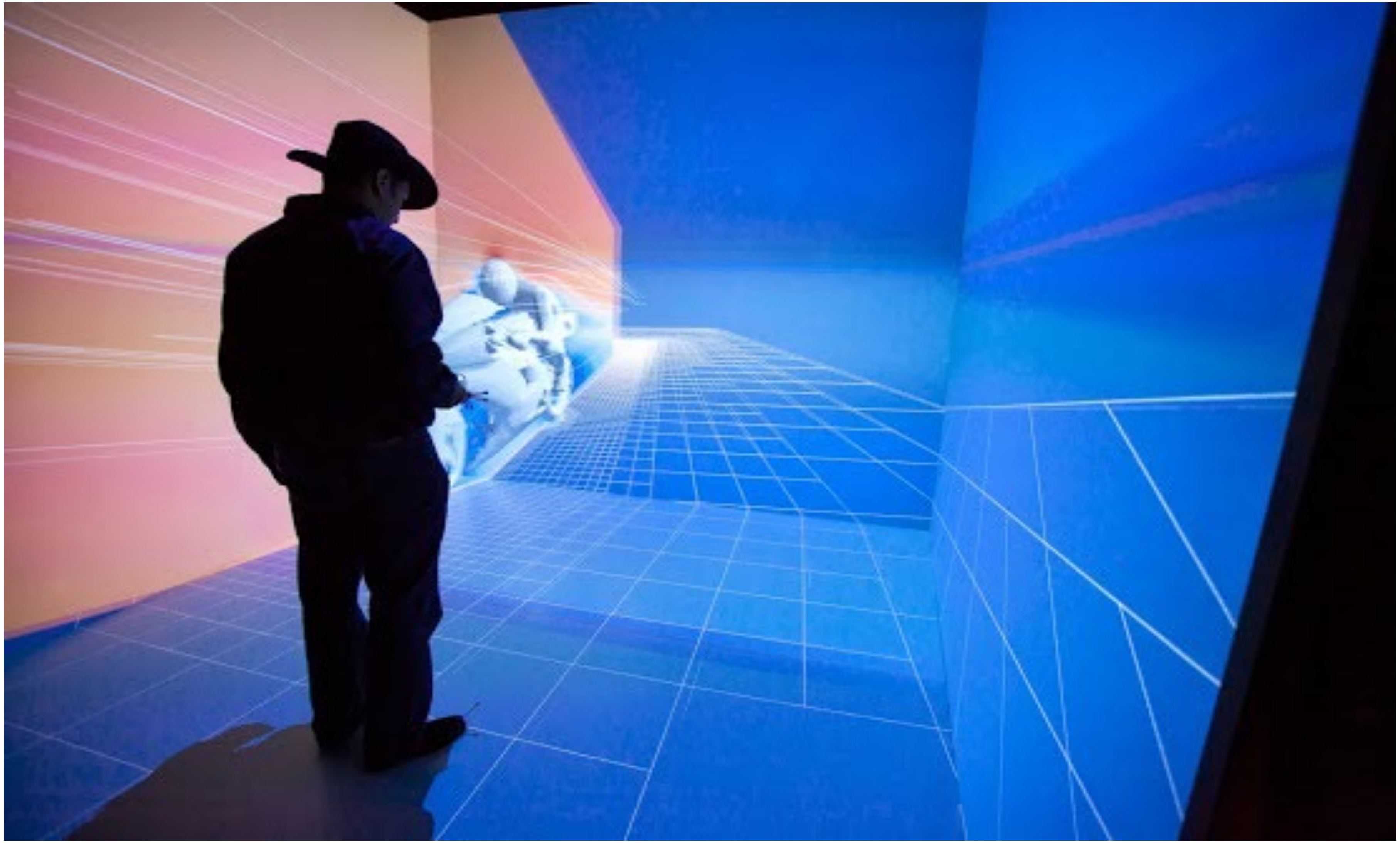
ParaView on the Web



ParaView Scripting - Python

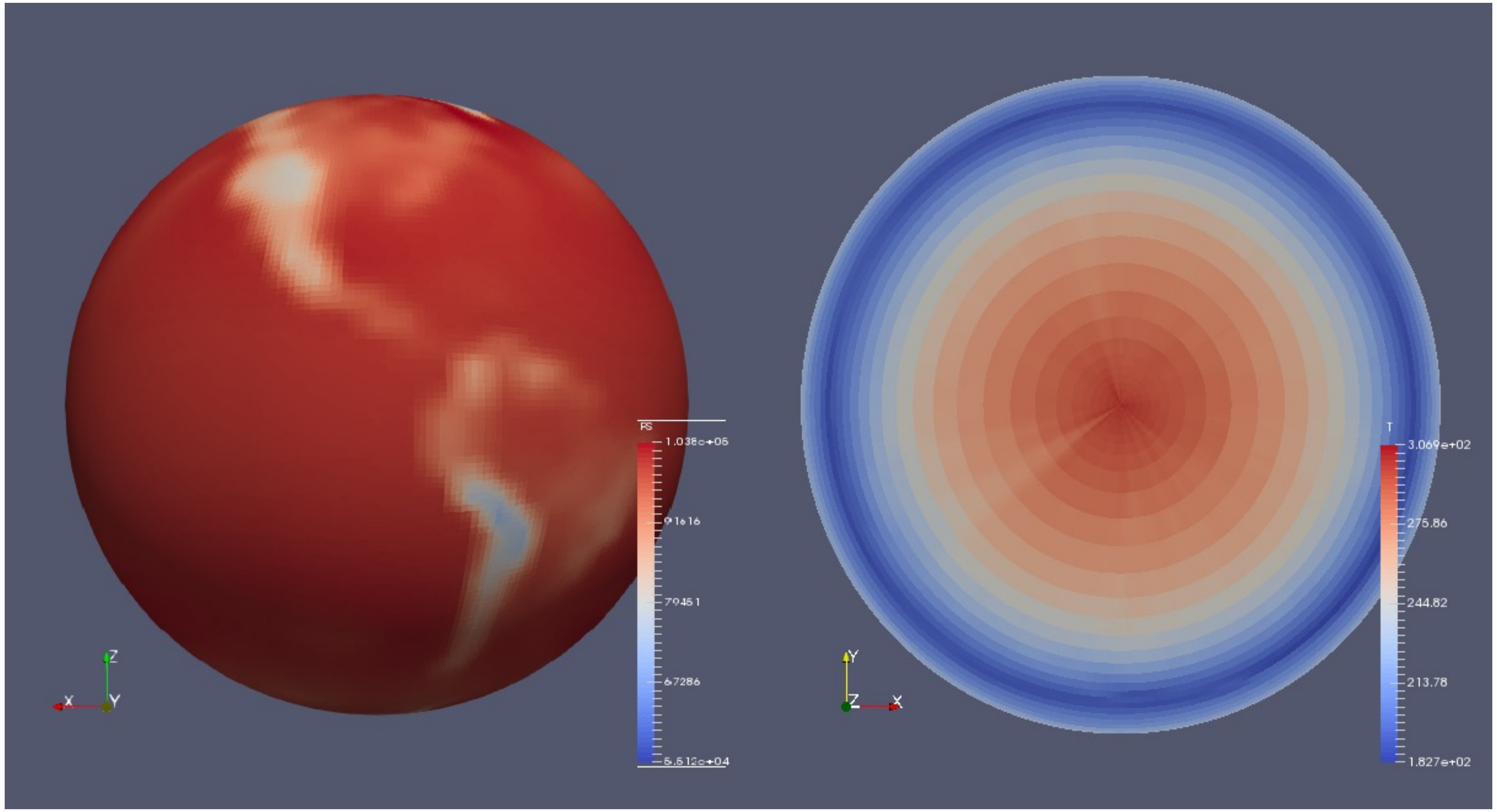
Python scripts can control ParaView with or without the GUI in order to create reproducible and customizable visualizations.

ParaView Immersive



ParaView for HPC

ParaView Catalyst

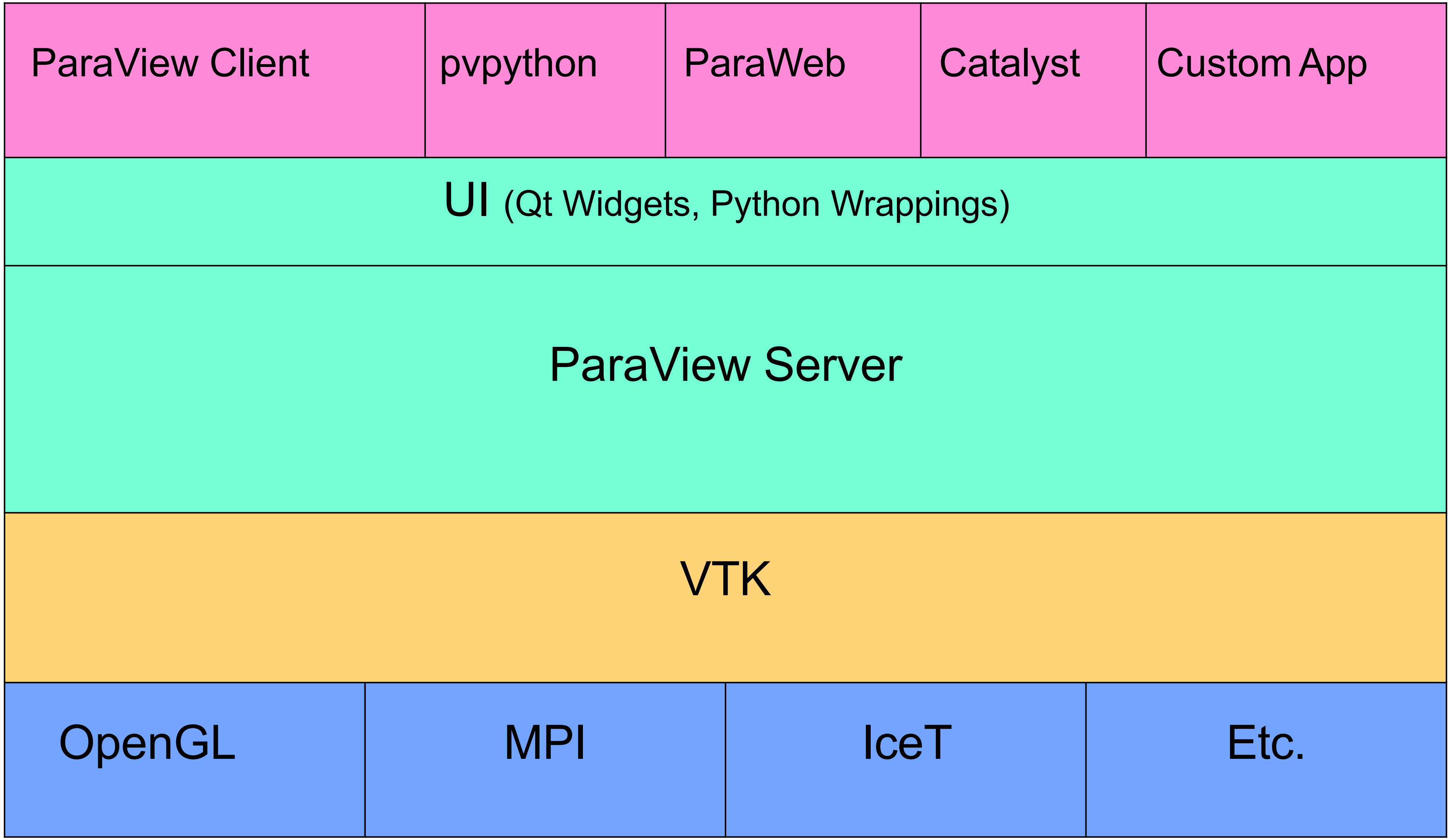


Community Atmosphere Model Data Visualized using ParaView Catalyst's
In Situ Capability

Data Ranges Handled by ParaView

- Used for all ranges of data size
- Landmarks of usage
  - 6 billion structured cells (2005)
  - 250 million unstructured cells (2005)
  - Billions of AMR cells (2008)
  - Scaling test over 1 Trillion cells (2010)
  - 6.33 billion unstructured cells in ParaView Catalyst (2016)
  - 1.1 trillion unstructured cells scaling test (2016)

ParaView Application Architecture



Supported Data Types by ParaView

- ParaView Data (.pvd)
- VTK (.vtp, .vtu, .vti, .vts, .vtr)
- VTK Legacy (.vtk)
- VTK Multi Block (.vtm,.vtmb,.vtmg,.vthd,.vthb)
- Partitioned VTK (.pvtu, .pvti, .pvts, .pvtr)
- ADAPT (.nc, .cdf, .elev, .ncd)
- ANALYZE (.img, .hdr)
- ANSYS (.inp)
- AVS UCD (.inp)
- BOV (.bov)
- BYU (.g)
- CAM NetCDF (.nc, .ncdf)
- CCSM MTSD (.nc, .cdf, .elev, .ncd)
- CCSM STSD (.nc, .cdf, .elev, .ncd)
- CEAucd (.ucd, .inp)
- CGNS (.cgns)
- CMAT (.cmat)
- CML (.cml)
- CTRL (.ctrl)
- Chombo (.hdf5, .h5)
- Claw (.claw)
- Comma Separated Values (.csv)
- Cosmology Files (.cosmo, .gadget2)
- Curve2D (.curve, .ultra, .ult, .u)
- DDCMD (.ddcmd)
- Digital Elevation Map (.dem)
- Dyna3D(.dyn)
- EnSight (.case, .sos)
- Enzo boundary and hierarchy
- ExodusII (.g, .e, .exe, .ex2, .ex2v.., etc)
- ExtrudedVol (.exvol)
- FVCOM (MTMD, MTSD, Particle, STSD)
- Facet Polygonal Data
- Flash multiblock files
- Fluent Case Files (.cas)
- GGCM (.3df, .mer)
- GTC (.h5)
- GULP (.trg)
- Gadget (.gadget)
- Gaussian Cube File (.cube)
- JPEG Image (.jpg, .jpeg)
- LAMPPS Dump (.dump)
- LAMPPS Structure Files
- LODI (.nc, .cdf, .elev, .ncd)
- LODI Particle (.nc, .cdf, .elev, .ncd)
- LS-DYNA (.k, .lsdyna, .d3plot, d3plot)
- M3DCl (.h5)
- MFIX Unstructred Grid (.RES)
- MM5 (.mm5)
- MPAS NetCDF (.nc, .ncdf)
- Meta Image (.mhd, .mha)
- Miranda (.mir, .raw)
- Multilevel 3d Plasma (.m3d, .h5)
- NASTRAN (.nas, .f06)
- Nek5000 Files
- Nrrd Raw Image (.nrrd, .nhdr)
- OpenFOAM Files (.foam)
- PATRAN (.neu)
- PFLOTRAN (.h5)
- PLOT2D (.p2d)
- PLOT3D (.xyz, .q, .x, .vp3d)
- PLY Polygonal File Format
- PNG Image Files
- POP Ocean Files
- ParaDIS Files
- Phasta Files (.pht)
- Pixie Files (.h5)
- ProSTAR (.cel, .vrt)
- Protein Data Bank (.pdb, .ent, .pdb)
- Raw Image Files
- Raw NRRD image files (.nrrd)
- SAMRAI (.samrai)
- SAR (.SAR, .sar)
- SAS (.sasgeom, .sas, .sasdata)
- SESAME Tables
- SLAC netCDF mesh and mode data
- SLAC netCDF particle data
- Silo (.silo, .pdb)
- Spheral (.spheral, .sv)
- SpyPlot CTH
- SpyPlot (.case)
- SpyPlot History (.hscth)
- Stereo Lithography (.stl)
- TFT Files
- TIFF Image Files
- TSurf Files
- Tecplot ASCII (.tec, .tp)
- Tecplot Binary (.plt)
- Tetrad (.hdf5, .h5)
- UNIC (.h5)
- VASP CHGCA (.CHG)
- VASP OUT (.OUT)
- VASP POSTCAR (.POS)
- VPIC (.vpc)
- VRML (.wrl)
- Velodyne (.vld, .rst)
- VizSchema (.h5, .vsh5)
- Wavefront Polygonal Data
- WindBlade (.wind)
- XDMF and hdf5 (.xmf, .xdmf)
- XMol Molecule

User Interface

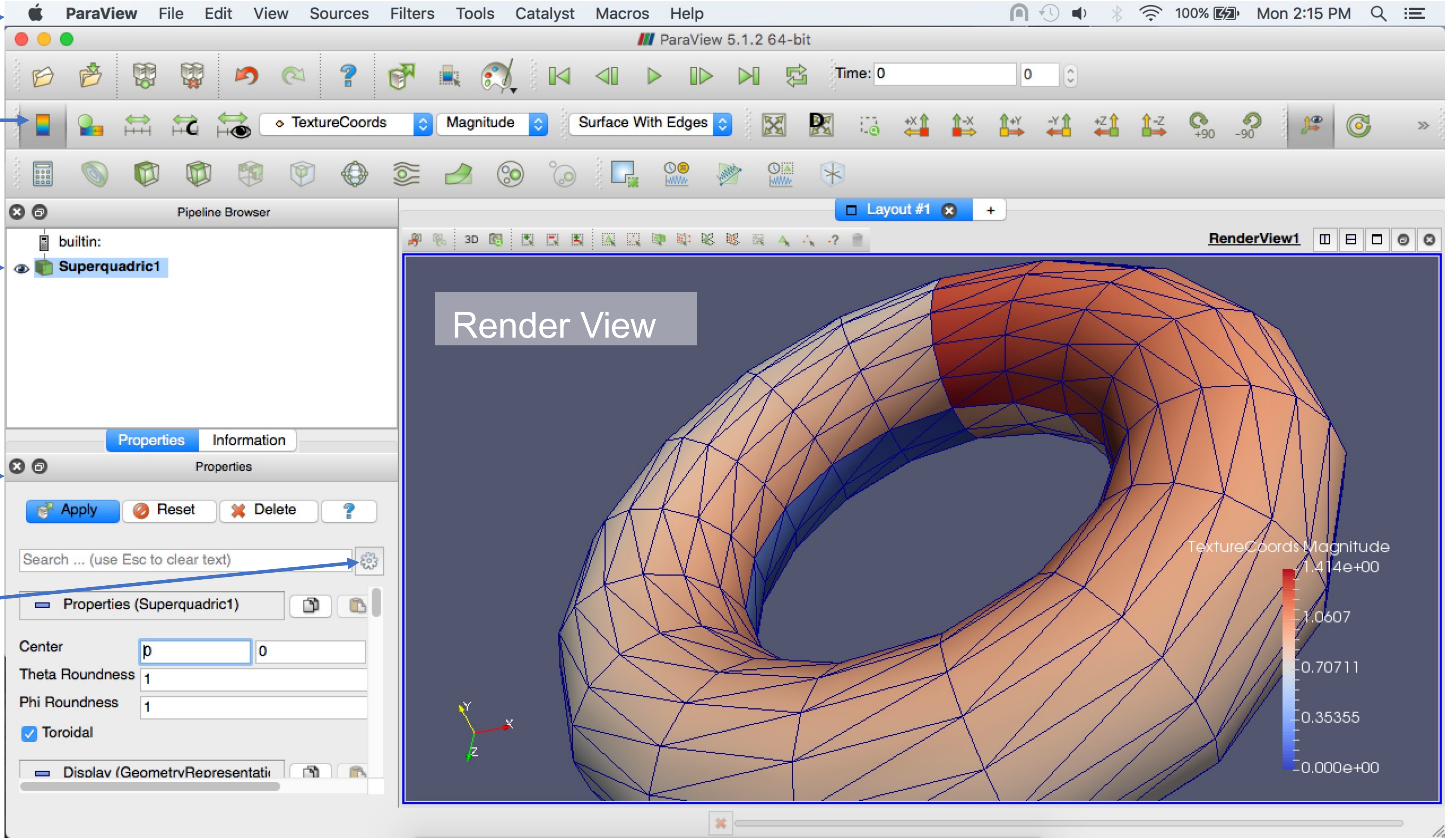
Menu bar

Toolbar

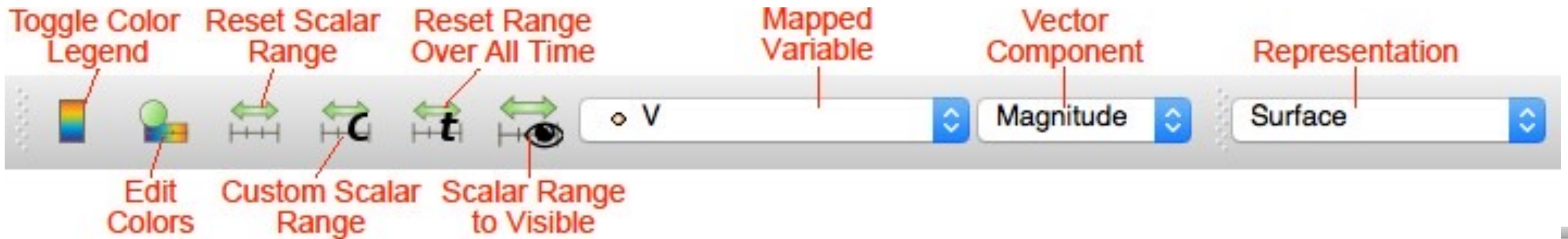
Pipeline Browser

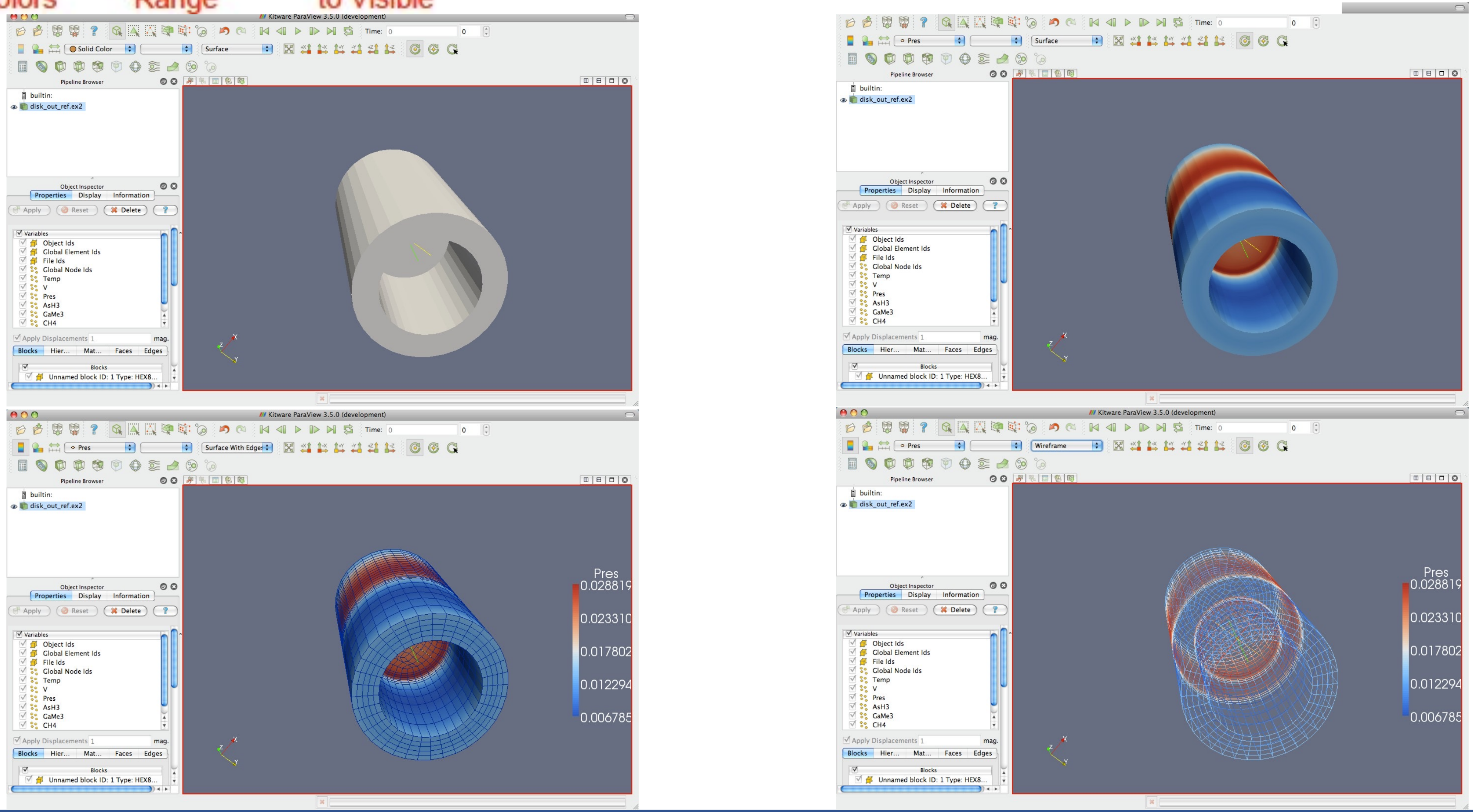
Properties Panel

Advanced Toggle



Data Representation

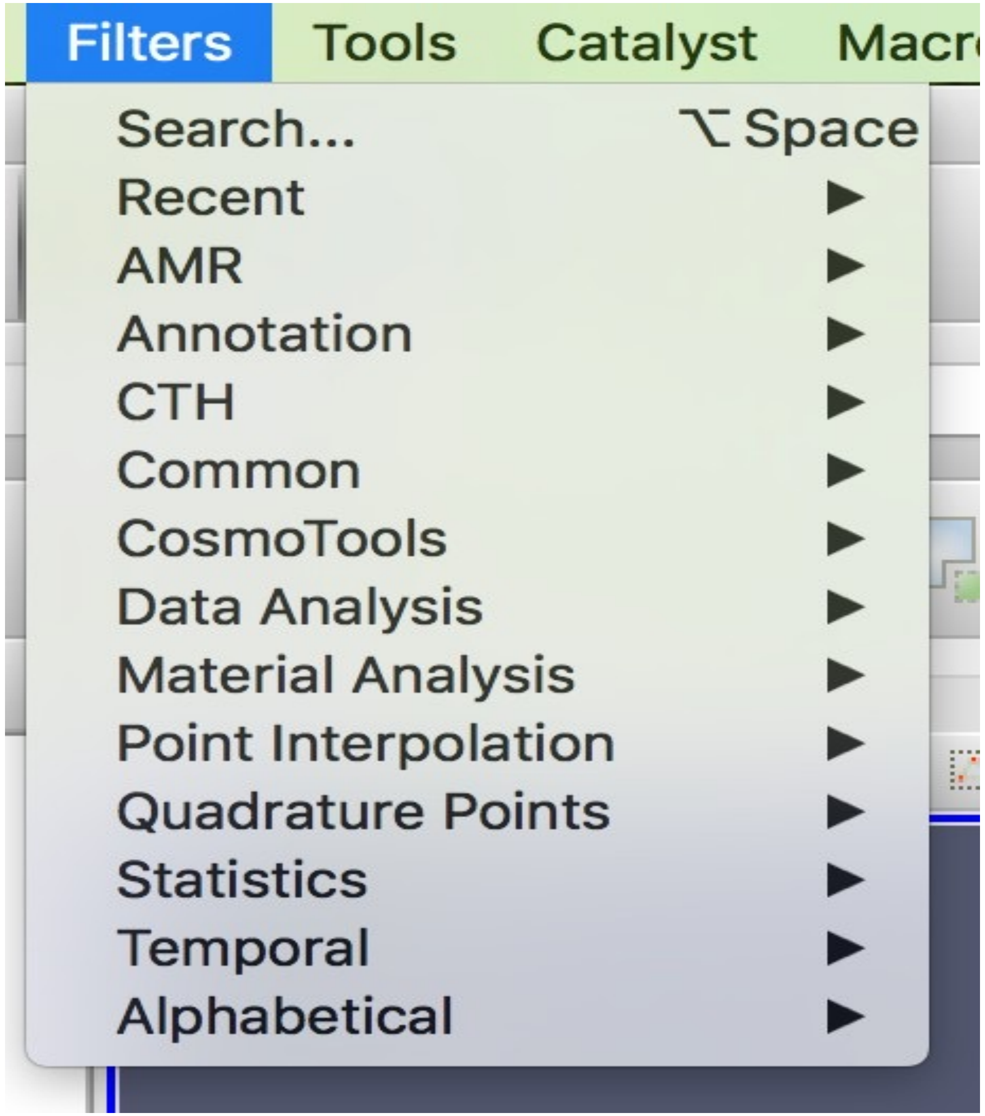




Common Filters

- Calculator
- Contour
- Clip
- Slice
- Threshold
- Extract Subset
- Glyph
- Stream Tracer
- Warp (vector)
- Group Datasets
- Extract Level

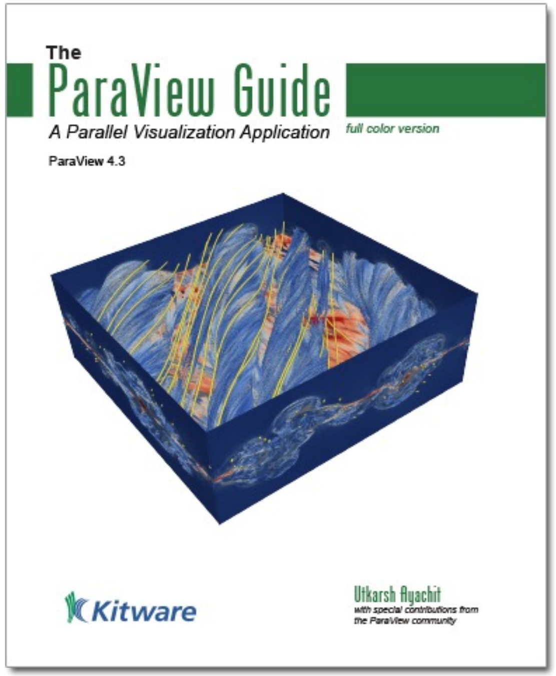
Filters Menu



~ 150 filters
+ C++ plugins
+ python filters

More Information

- Help Menu
  - Getting Started
  - *The ParaView Tutorials*
  - *The ParaView Guide*
    - *aka The Book*
  - The ParaView web page
    - www.paraview.org
  - ParaView discussion forum
    - https://discourse.paraview.org/



Tutorial: https://www.youtube.com/watch?v=sXY72e3Ce4g&list=PLGj2a3KTwhRZ7XupM7f36czTGlJvqq7_N&index=3

ParaView Demo