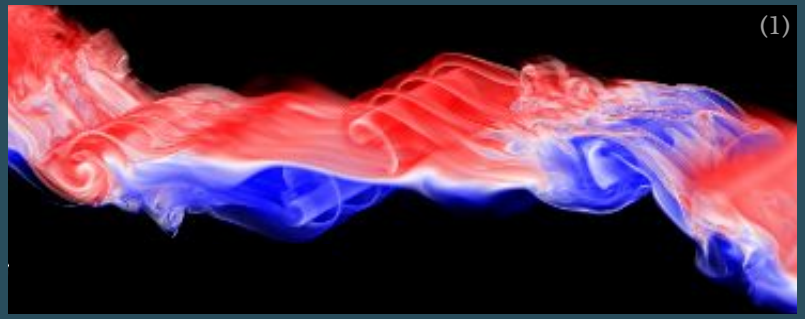
Three-Dimensional Graphics

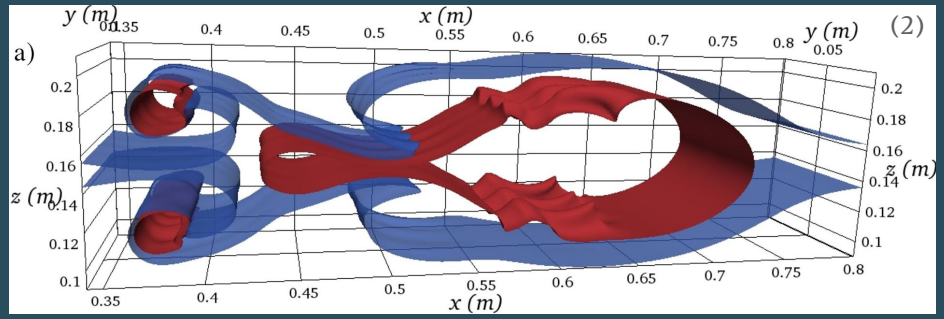
as a Tool for Studying Dynamics

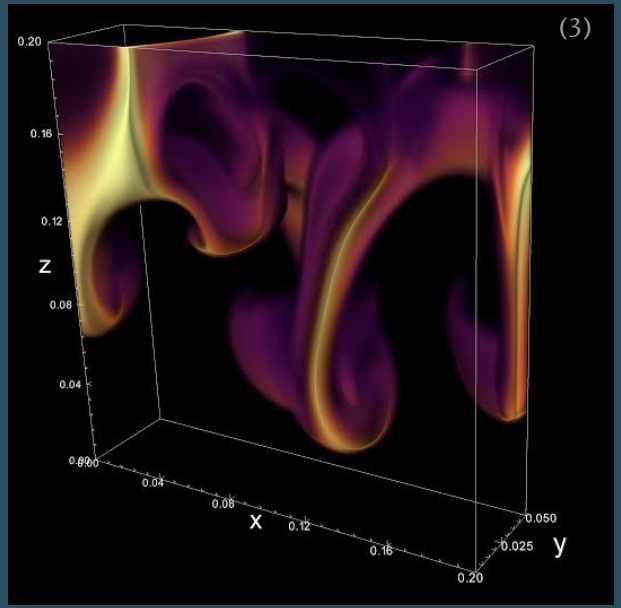
Laura Chandler, Marek Stastna

Contributions from: Aaron Coutino, David Deepwell, Andrew Grace, Jared Penney

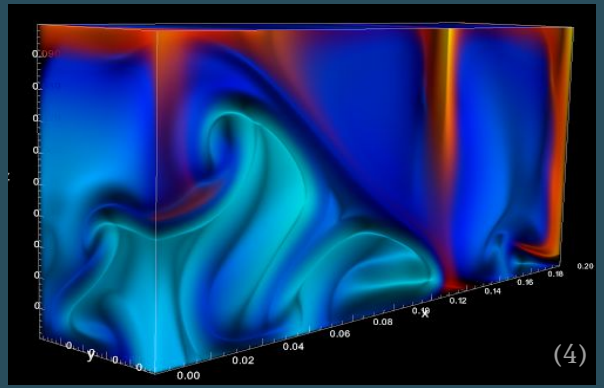
Department of Applied Mathematics, University of Waterloo, Waterloo ON

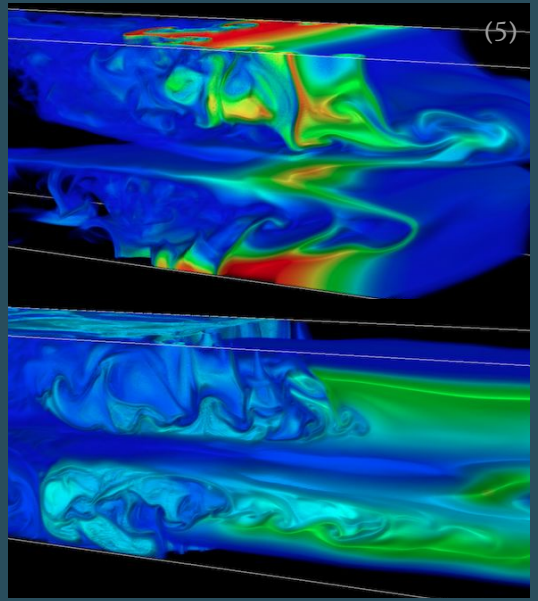






3D Graphics: Where to begin?





Images: Andrew Grace (1), David Deepwell (2), Laura Chandler (3-4), Aaron Coutino (5)

3D Graphics: Where to begin?

File Type: netCDF

- Represents scientific data using arrays
- Scalable
- Portable
- Appendable

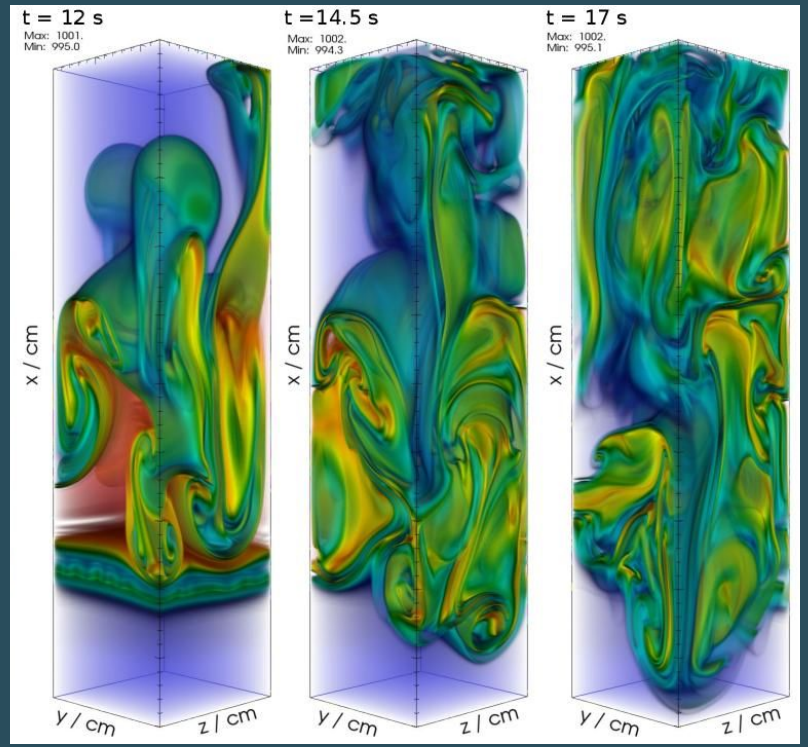
Visualization Tool: VisIt

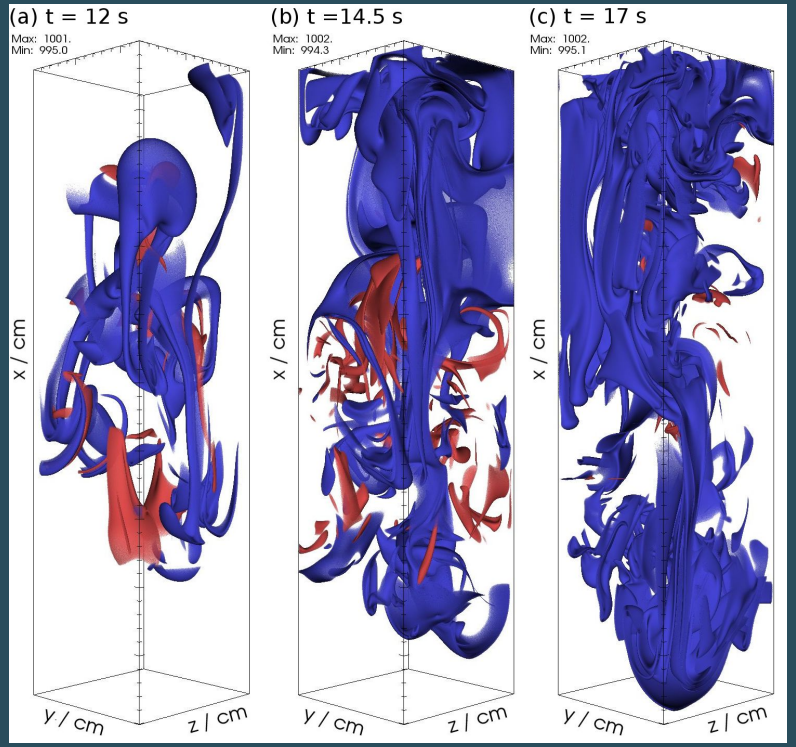
- Interactive
- Scalable
- Parallel
- Supports different meshes
- Plot vector, scalar, tensor fields
- Easy graphical interface

VisIt is compatible with...

- ANSYS
- Boxlib
- CGNS
- Chombo
- CTRL
- Ensight Gold
- Enzo
- Exodus
- FITS
- FLUENT
- FVCOM
- GGCM
- GIS
- H5Nimrod
- H5Part
- Image
- ITAPS
- MFIX
- MM5
- NASTRAN
- Nek5000
- NetCDF
- OpenFOAM
- PATRAN
- Silo
- Spheral
- VTK
- Wavefront OBJ
- Xmdv
- XDMF
- ZeusMP
- And more!

VisIt: Both visually pleasing and scientifically relevant

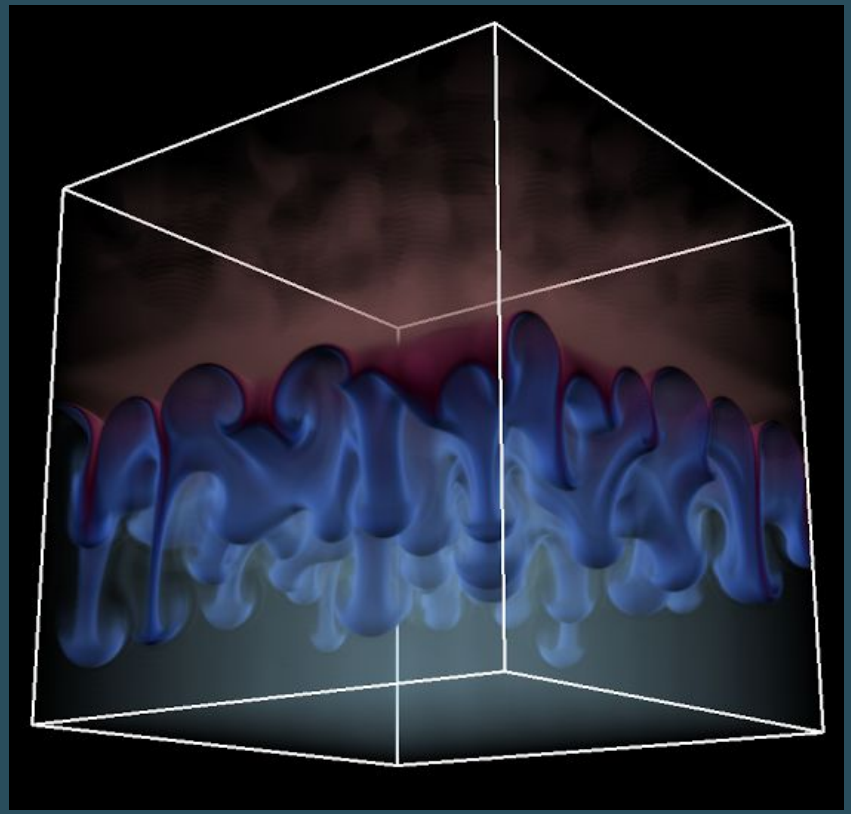




Images: Jared Penney

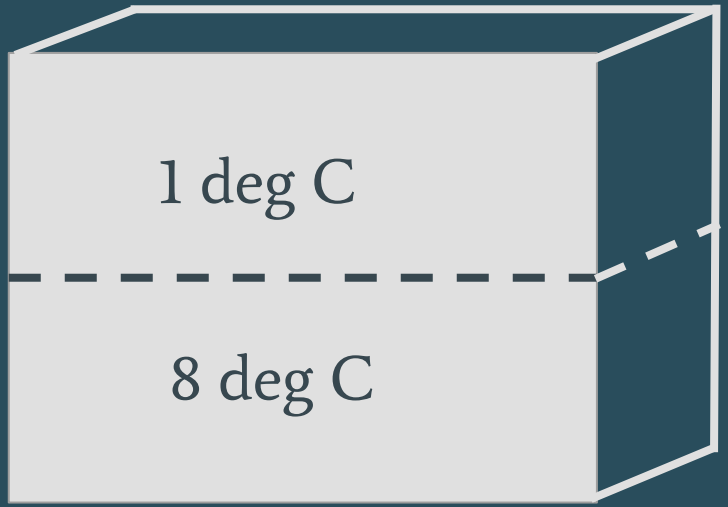
VisIt in 6 steps:

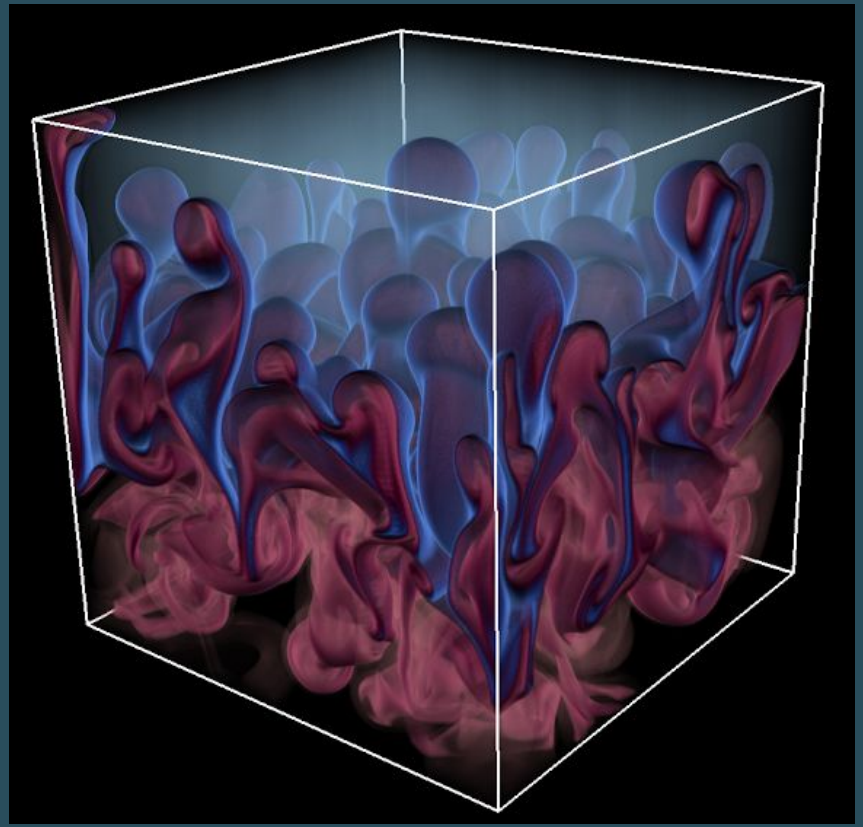
- Expressions
- Plot types
- Colormaps
- Opacity
- Creativity!
  (lighting, movies, etc.)
- Python scripting



Model Setup

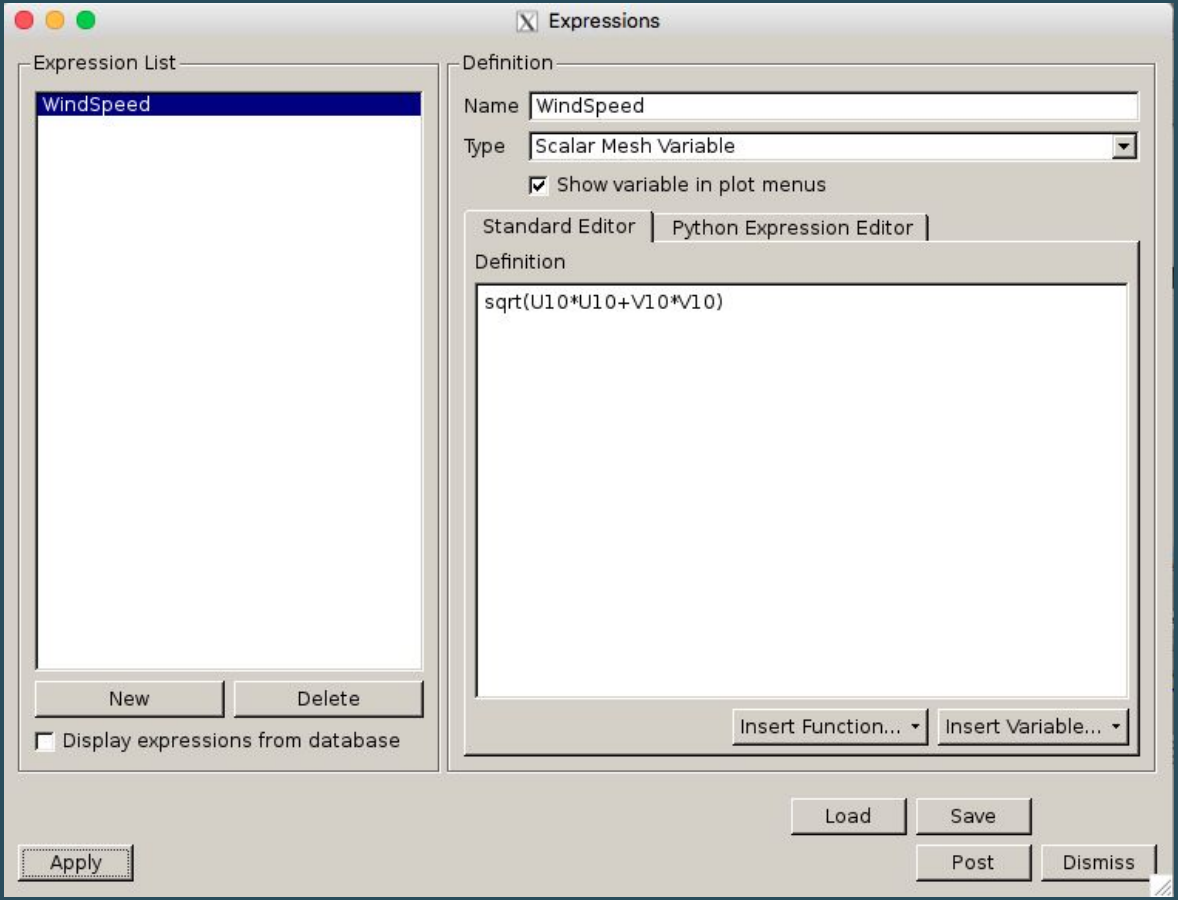
Rayleigh Taylor setup
near density maximum





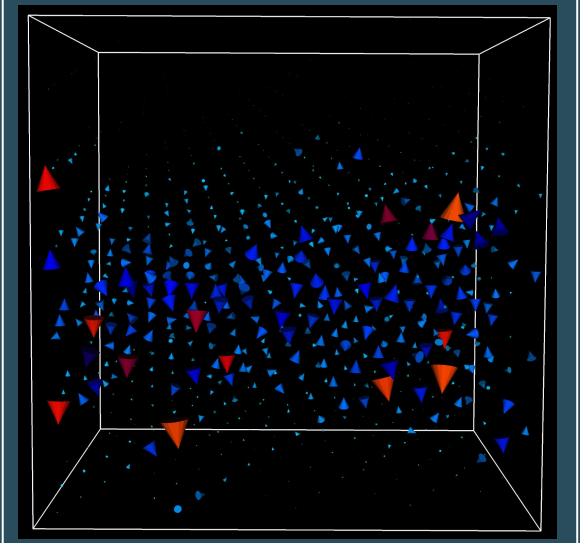
Expressions

Calculate new fields with given info

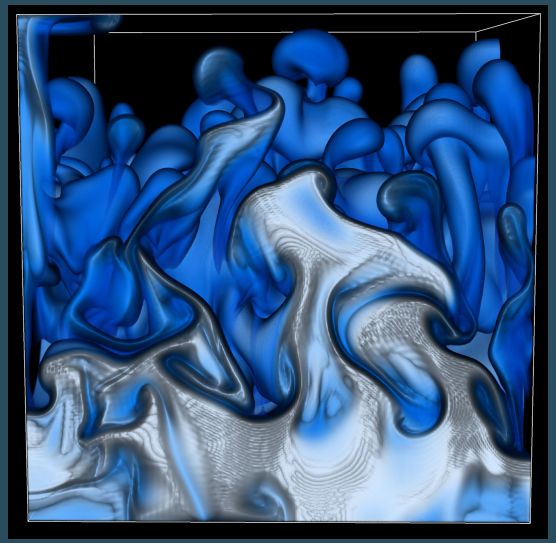


Types of plots:

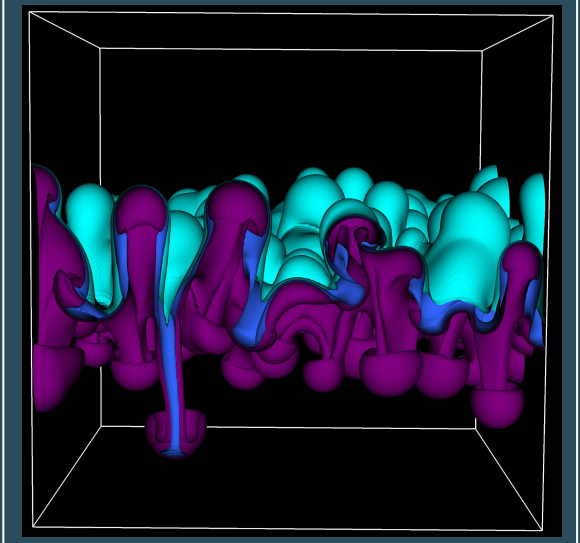
Vector



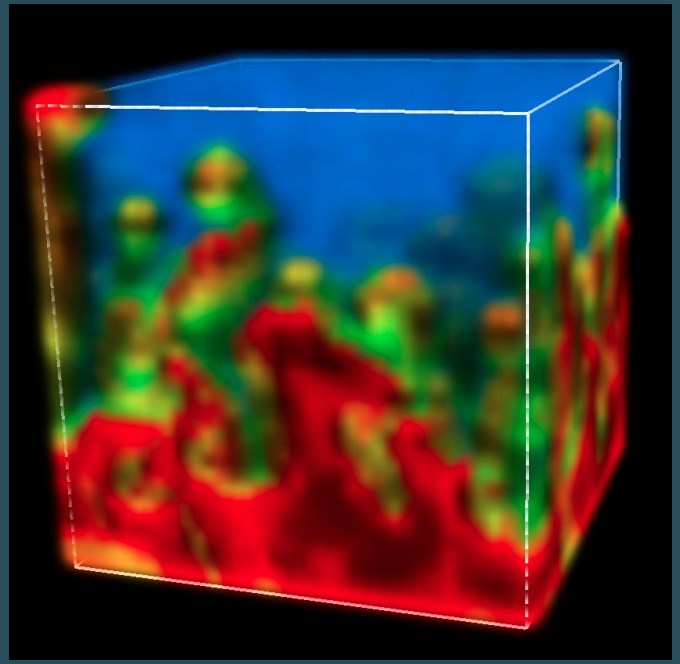
Volume



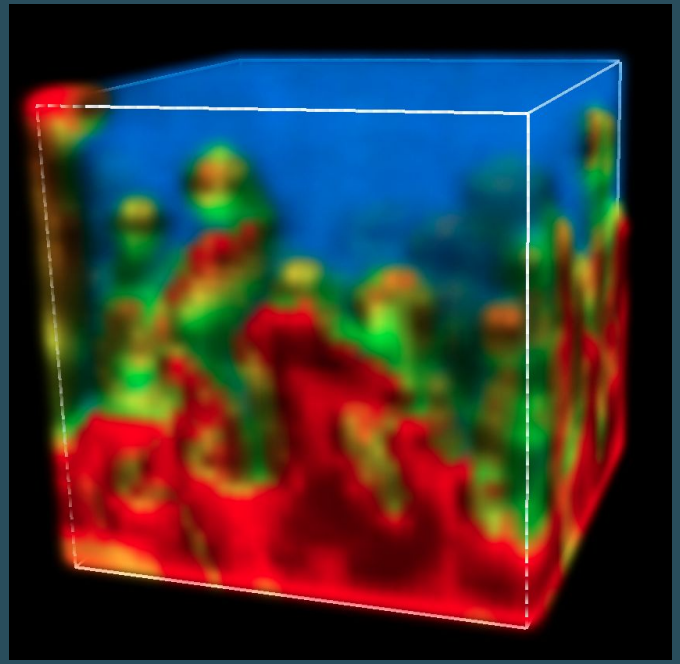
Contour



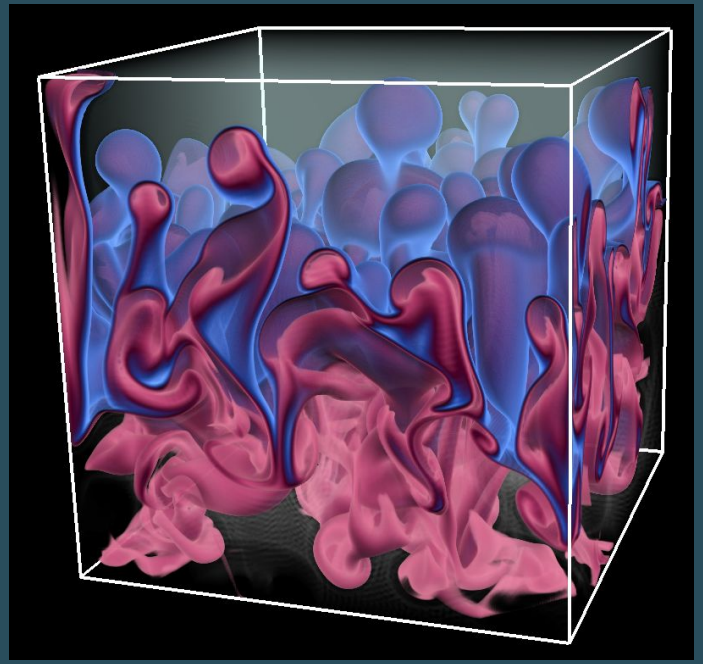
Volume Plots



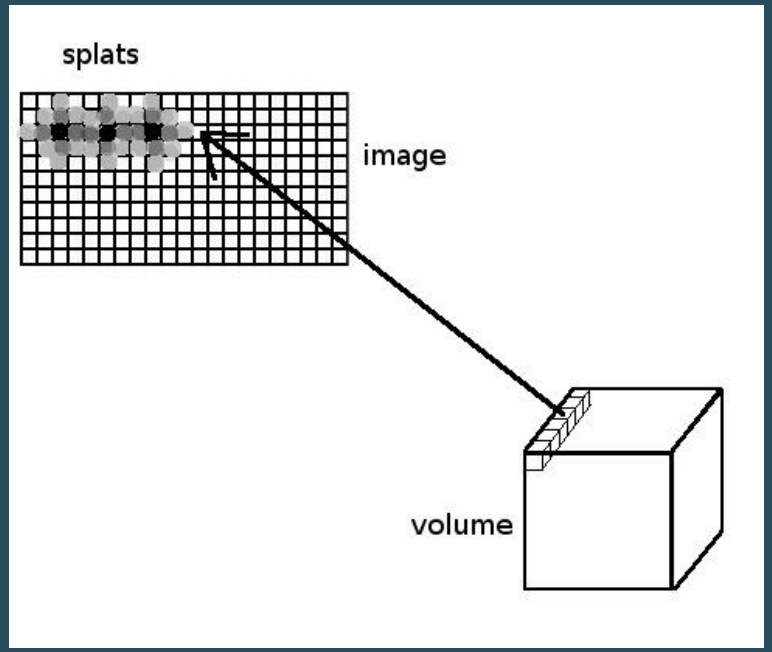
Volume Plots



???



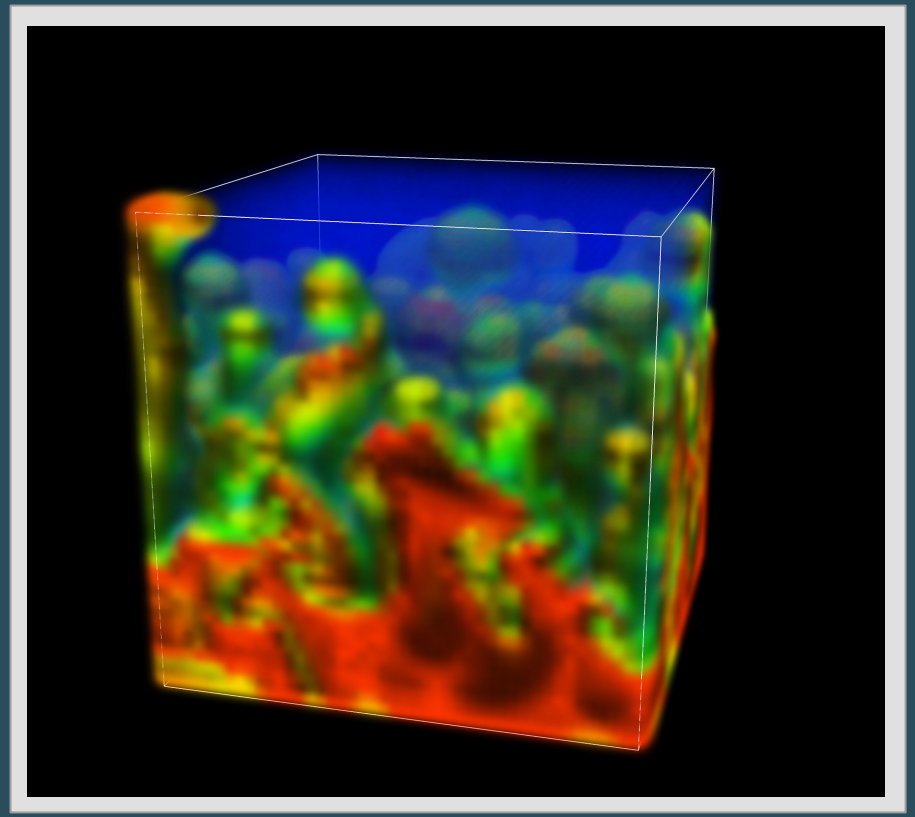
Volume Plots - Splatting



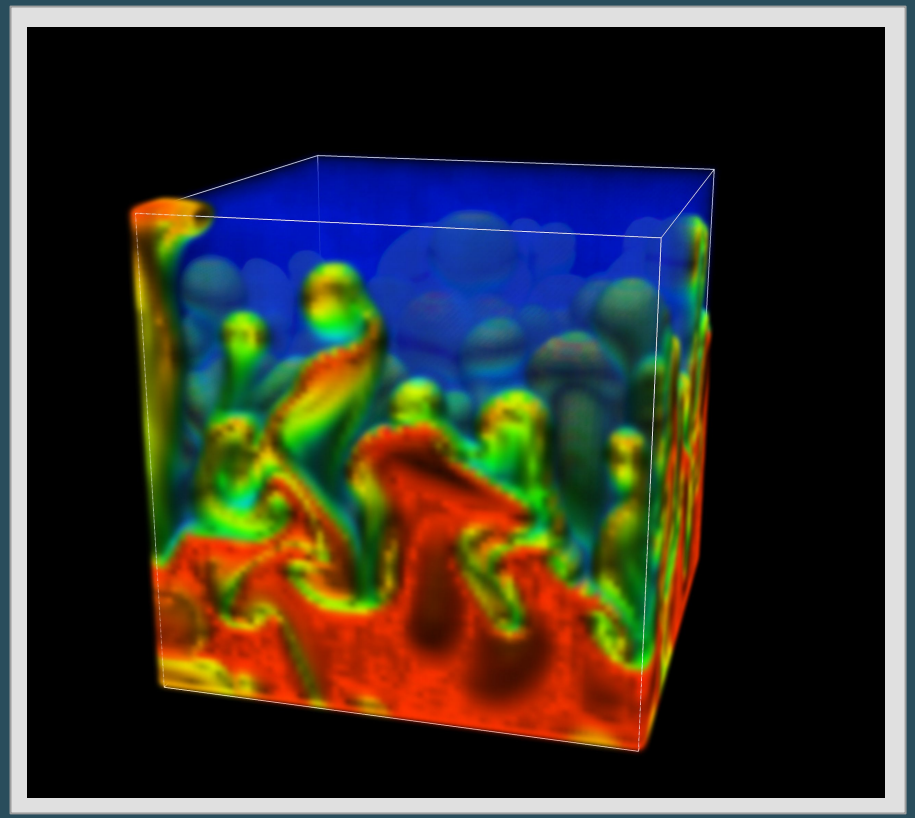
Splatting - draws every single voxel from the 3D mesh onto the 2D image, from back to front.

Splatting is faster since every voxel is a sample of many mesh points.

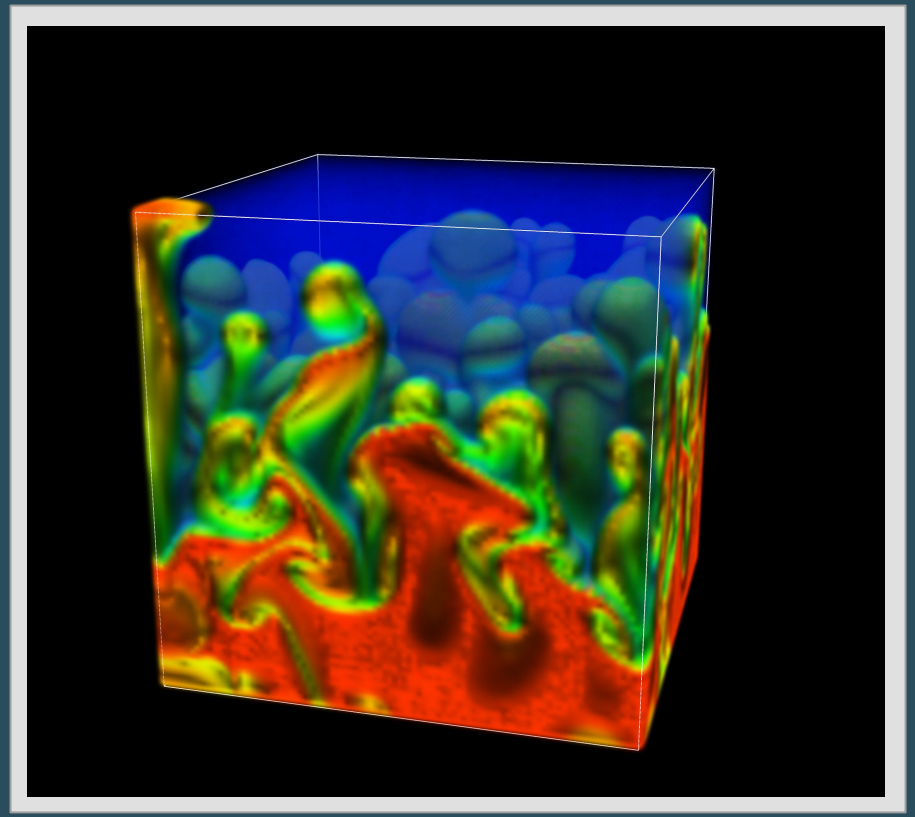
Image: Bloch (2006)



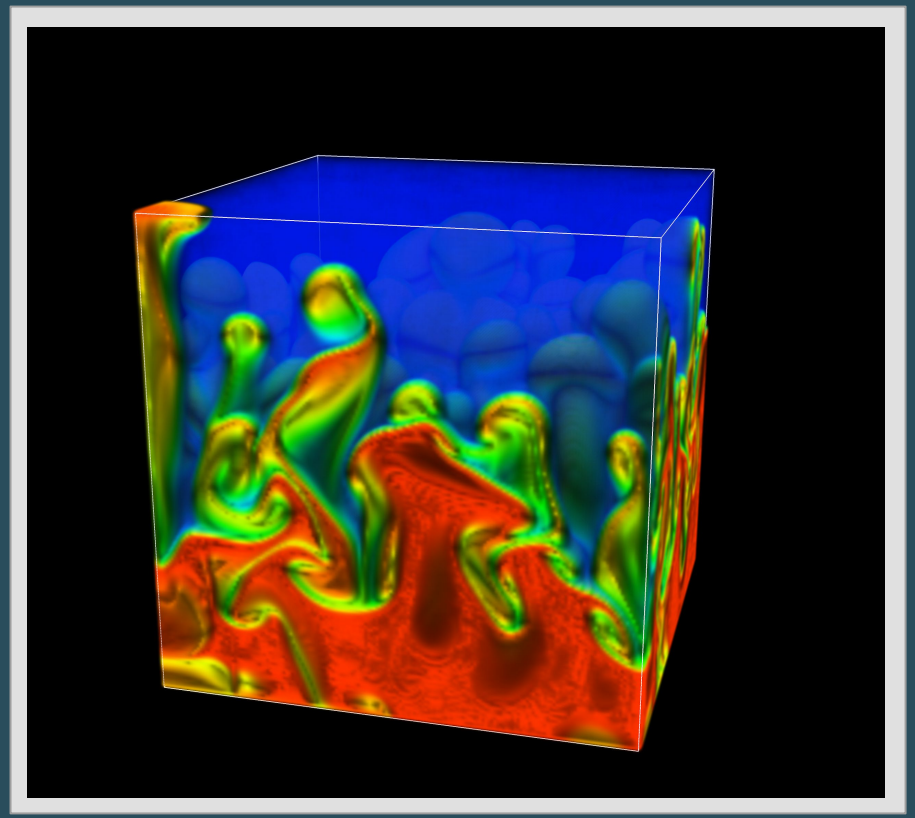
Splatting - 100 000 samples



Splatting - 500 000 samples

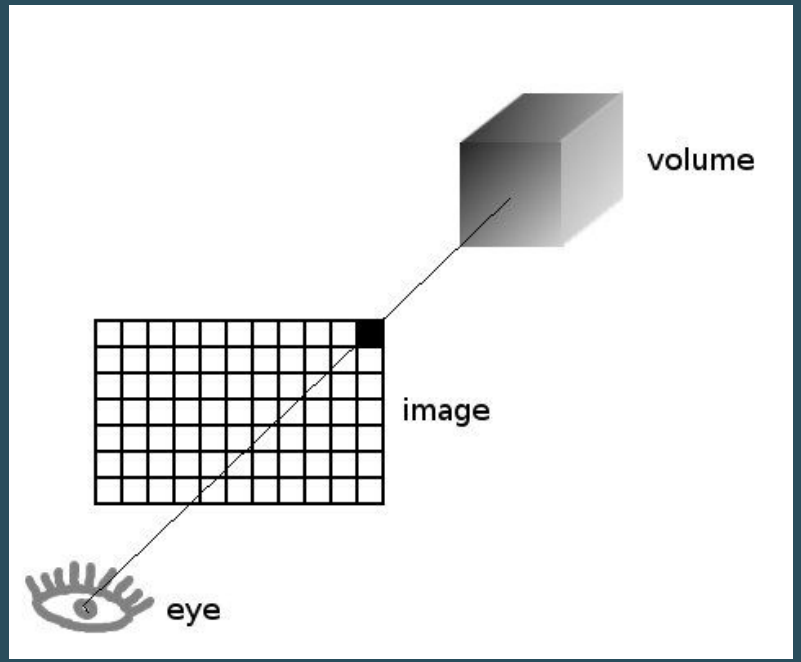


Splatting - 1 000 000 samples



Splatting - 5 000 000 samples

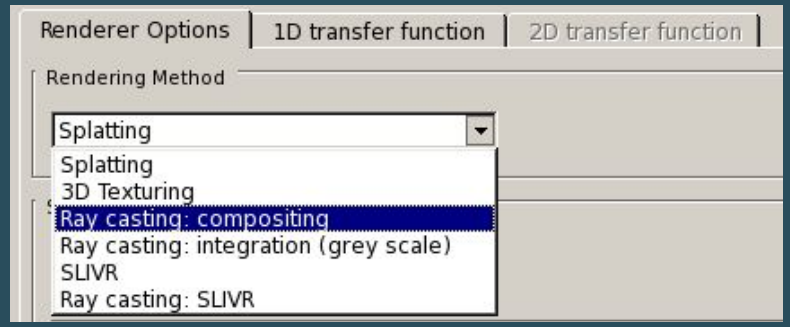
Volume Plots - Ray Casting

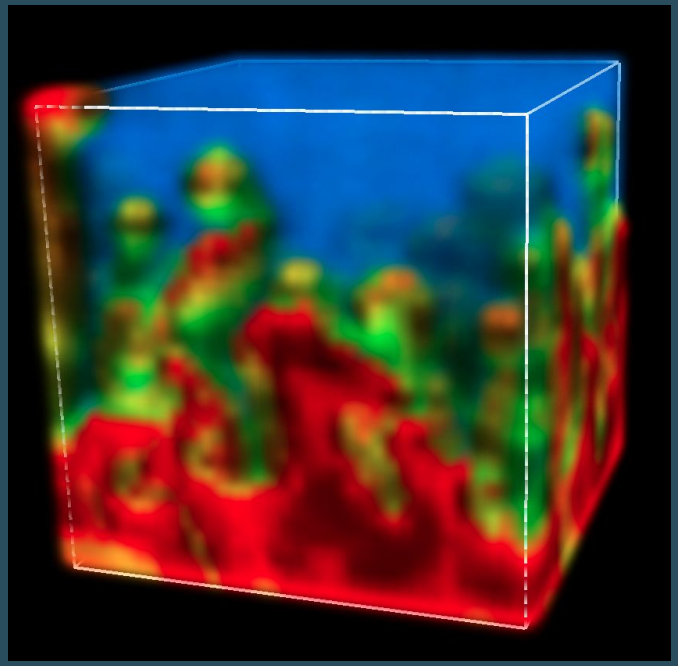


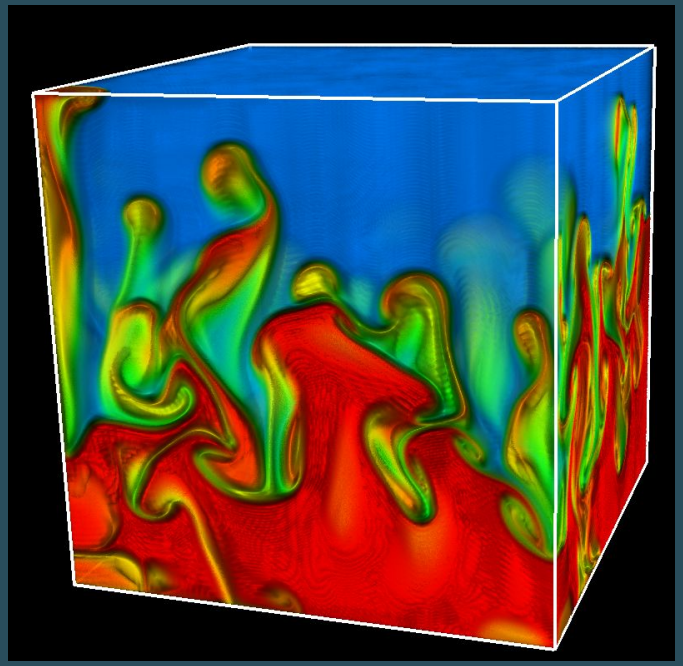
Ray Casting - converts a 3D mesh to 2D by drawing only the pixels needed, as seen by an ‘eye’ (or a ‘ray’).

Even though Ray Casting is parallel, it’s slower since it shows all the data instead of a sample.

Image: Bloch (2006)

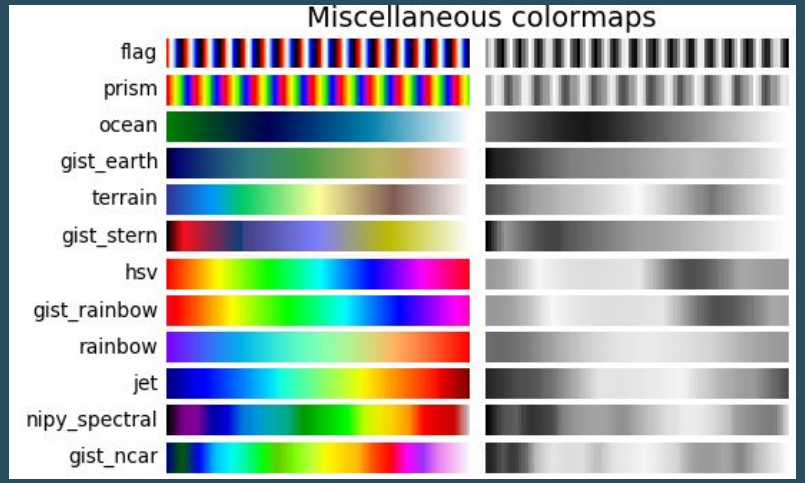

Renderer Options
1D transfer function
2D transfer function
Rendering Method
Splatting
Splatting
3D Texturing
Ray casting: compositing
Ray casting: integration (grey scale)
SLIVR
Ray casting: SLIVR

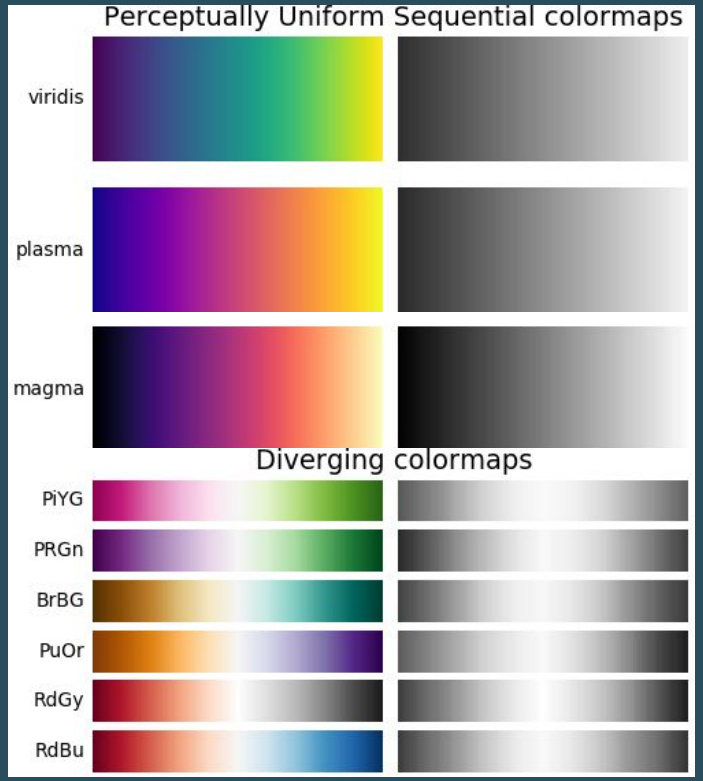


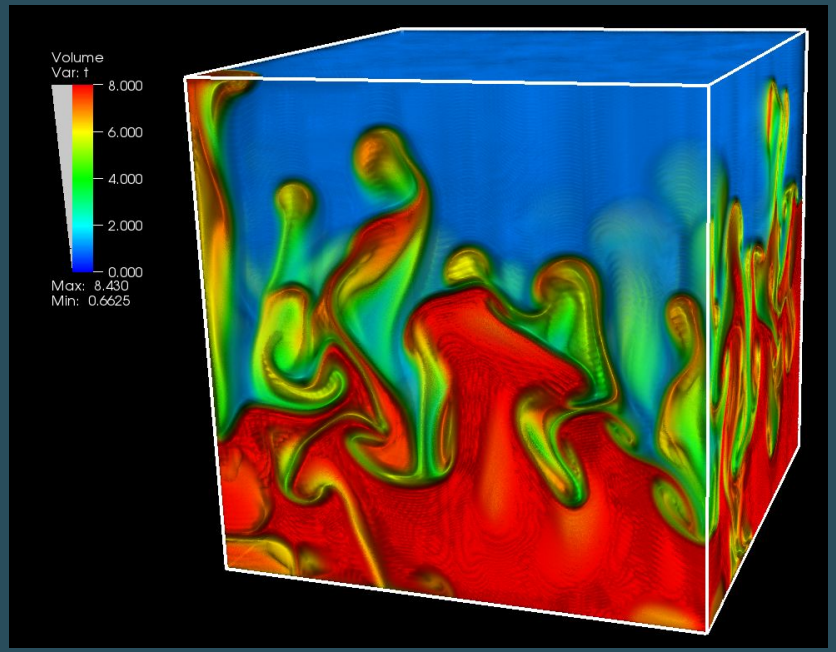


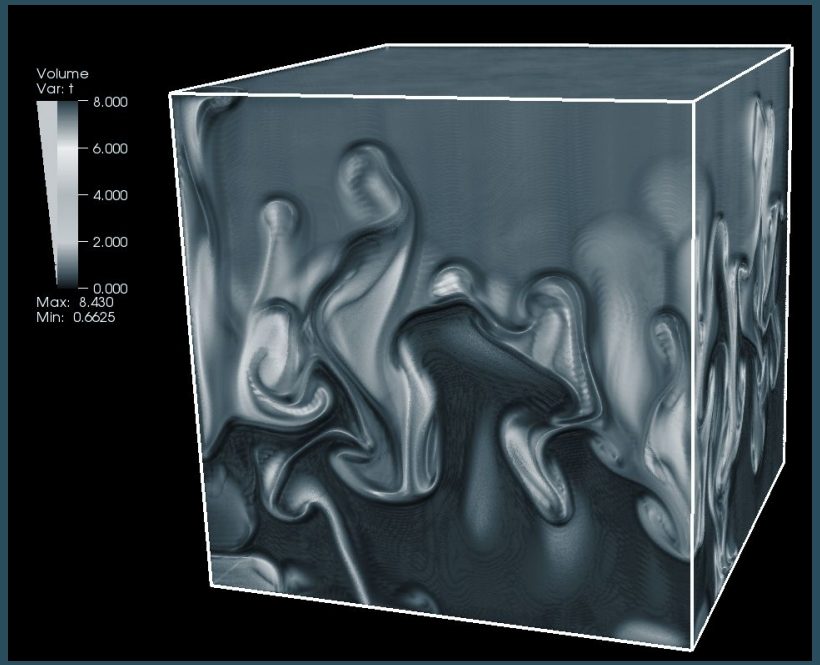
Colormaps - make your own!

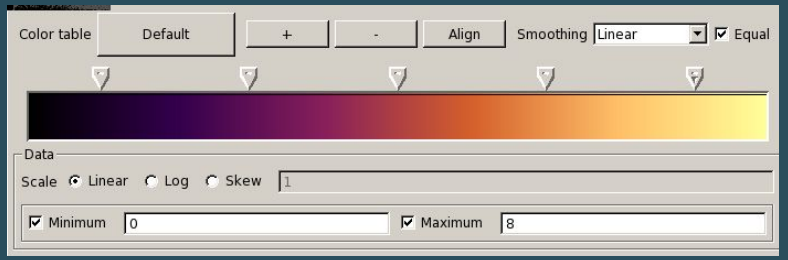
Choose colormap based on your data
Consider how it looks in grayscale, avoid red/green together (VisCheck)

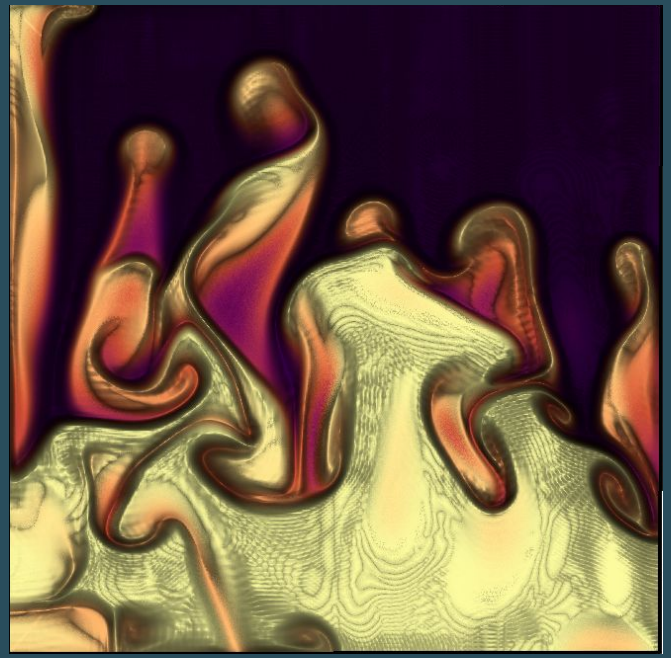


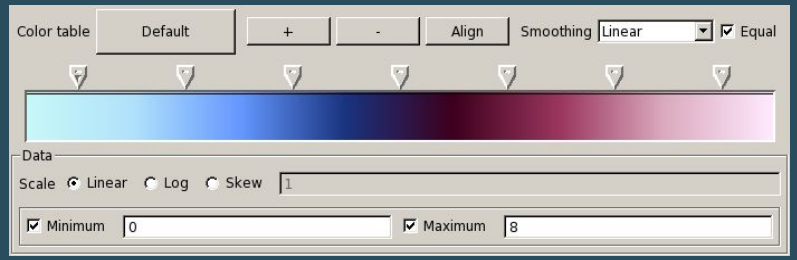


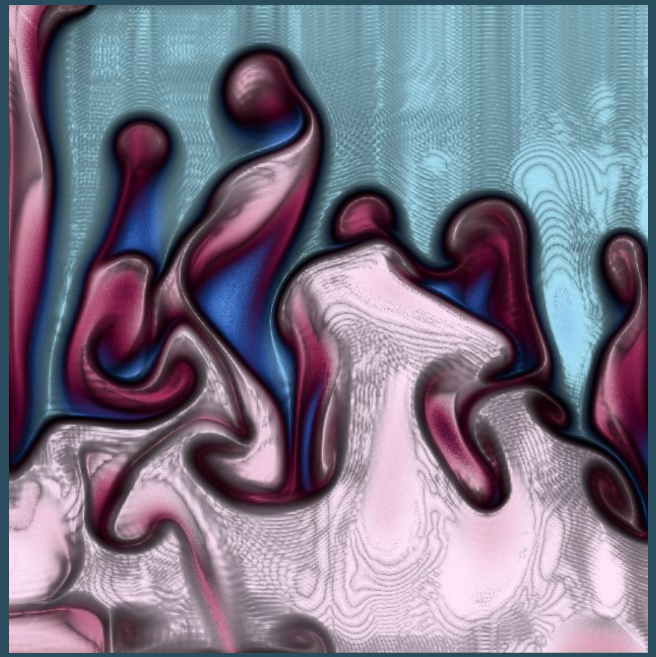

Volume
Var: t
8.000
6.000
4.000
2.000
0.000
Max: 8.430
Min: 0.6625


Volume
Var: t
8.000
6.000
4.000
2.000
0.000
Max: 8.430
Min: 0.6625

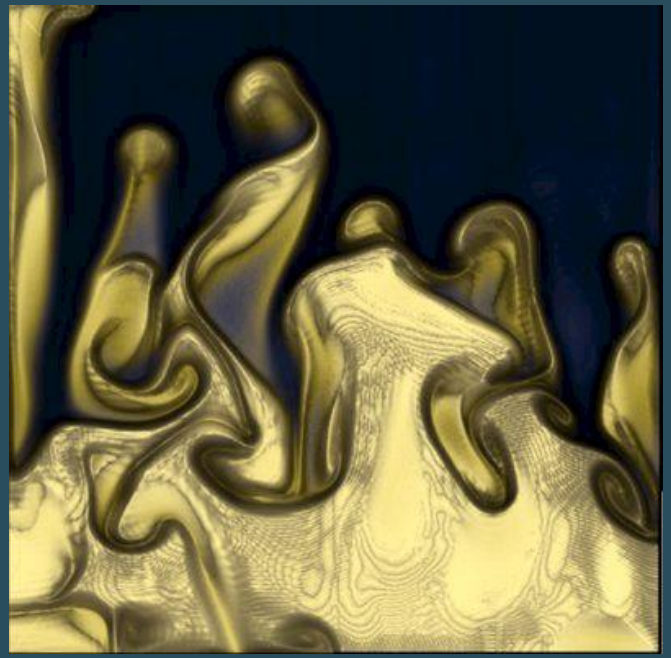

Color table
Default
+
-
Align
Smoothing
Linear
Equal
Data
Scale
Linear
Log
Skew
1
Minimum
0
Maximum
8



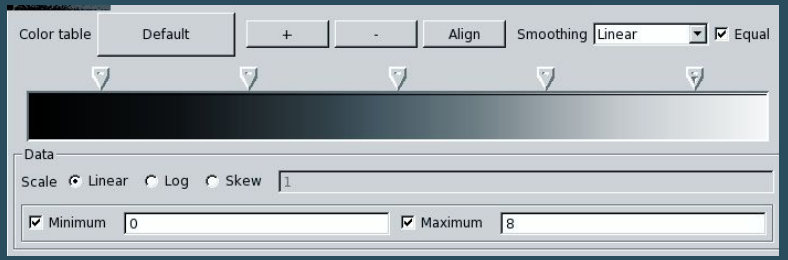

Color table
Default
+
-
Align
Smoothing
Linear
Equal
Data
Scale
Linear
Log
Skew
1
Minimum
0
Maximum
8

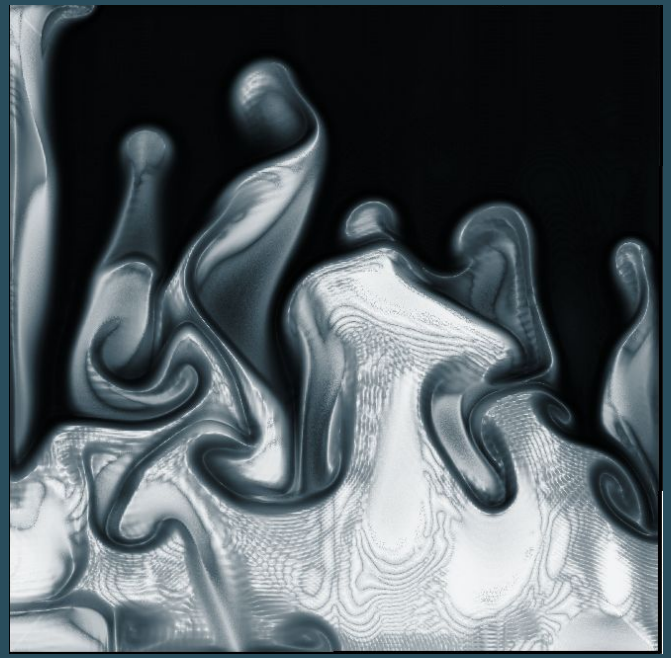


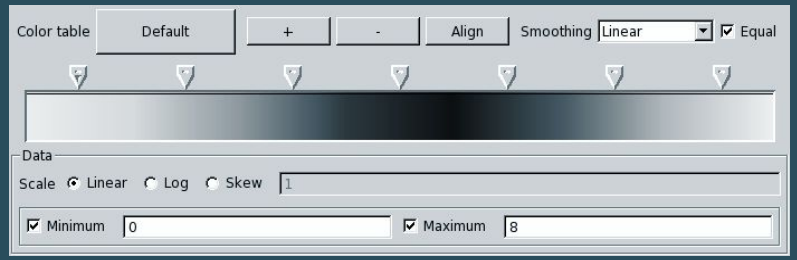

Color table
Default
+
-
Align
Smoothing
Linear
Equal
Data
Scale
Linear
Log
Skew
1
Minimum
0
Maximum
8



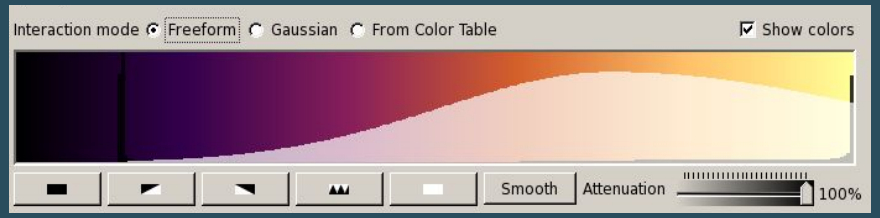

Color table
Default
+
-
Align
Smoothing
Linear
Equal
Data
Scale
Linear
Log
Skew
1
Minimum
0
Maximum
8

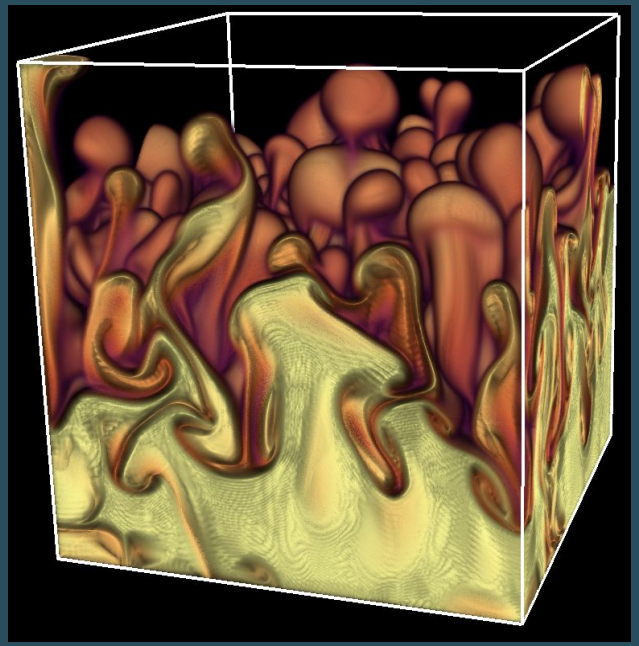

Color table
Default
+
-
Align
Smoothing
Linear
Equal
Data
Scale
Linear
Log
Skew
1
Minimum
0
Maximum
8



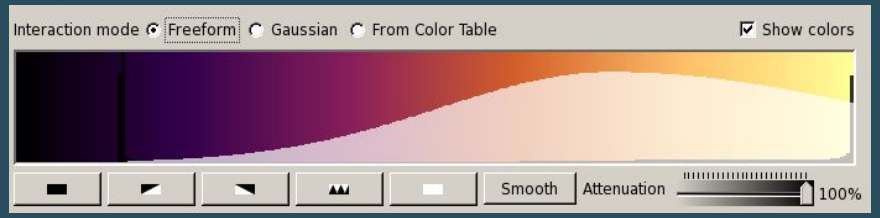

Color table
Default
+
-
Align
Smoothing
Linear
Equal
Data
Scale
Linear
Log
Skew
1
Minimum
0
Maximum
8

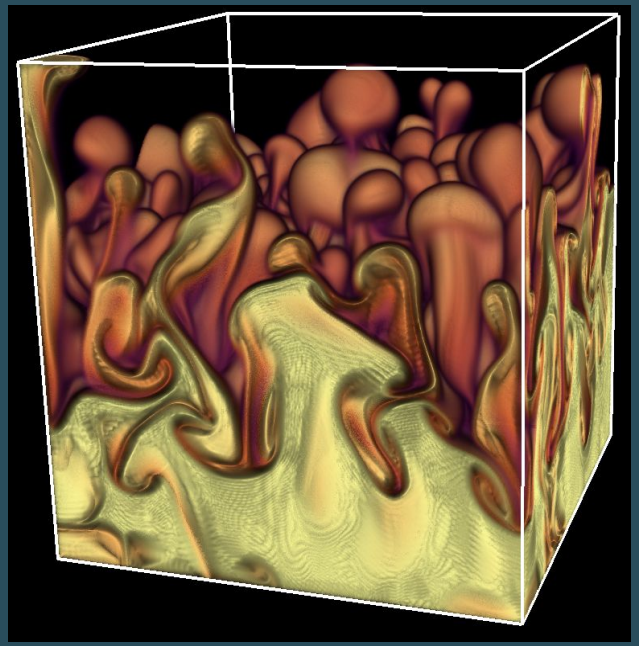
Opacity- use the ‘Volume Transfer Function’

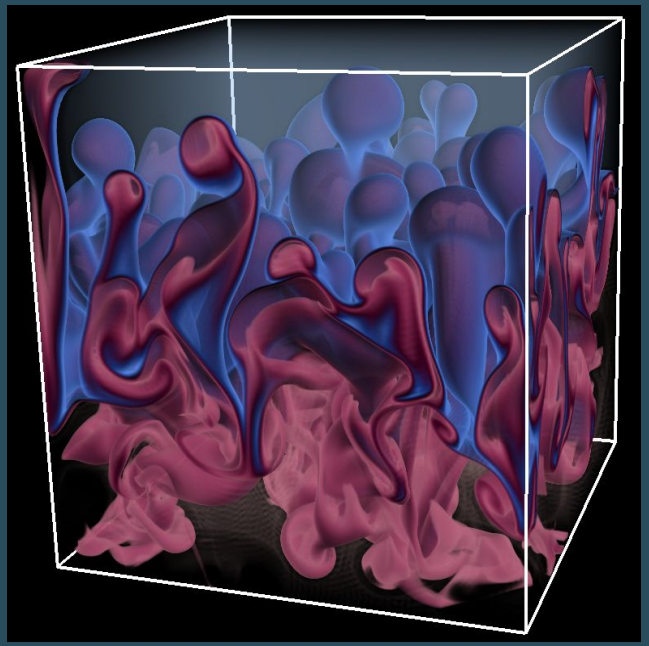




Opacity- use the ‘Volume Transfer Function’

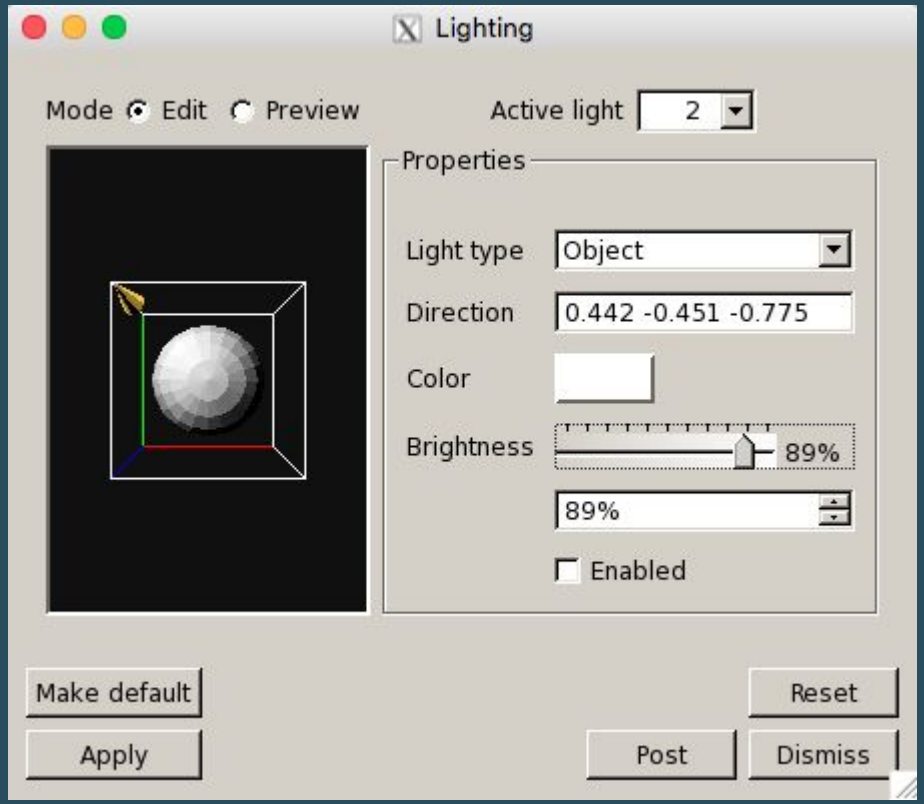






Other features

- Lighting
- Axes
- Labels
- Superimposing plots
- Movies
  - Time series
  - Spinning
  - Slices



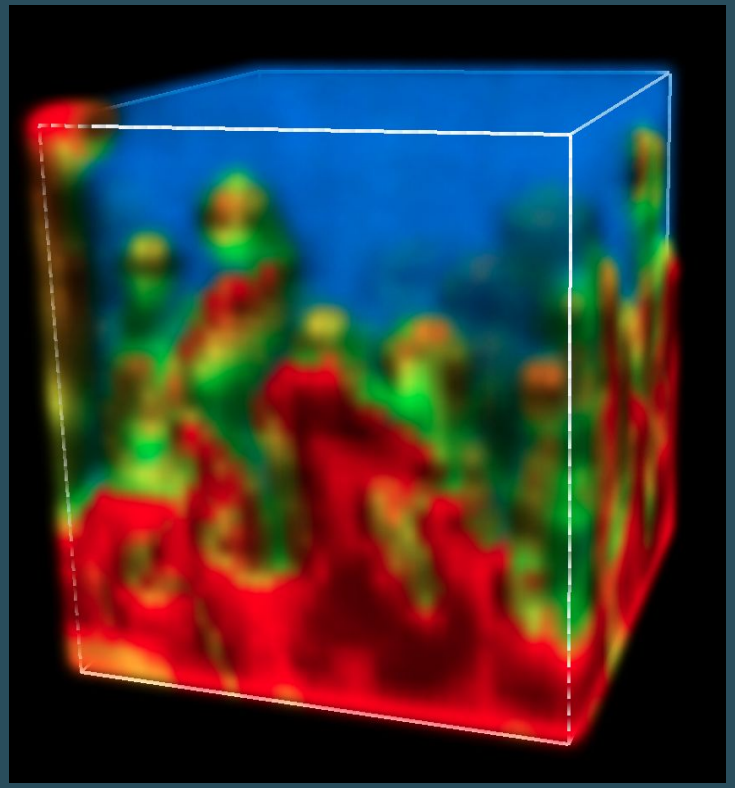
Python Scripting

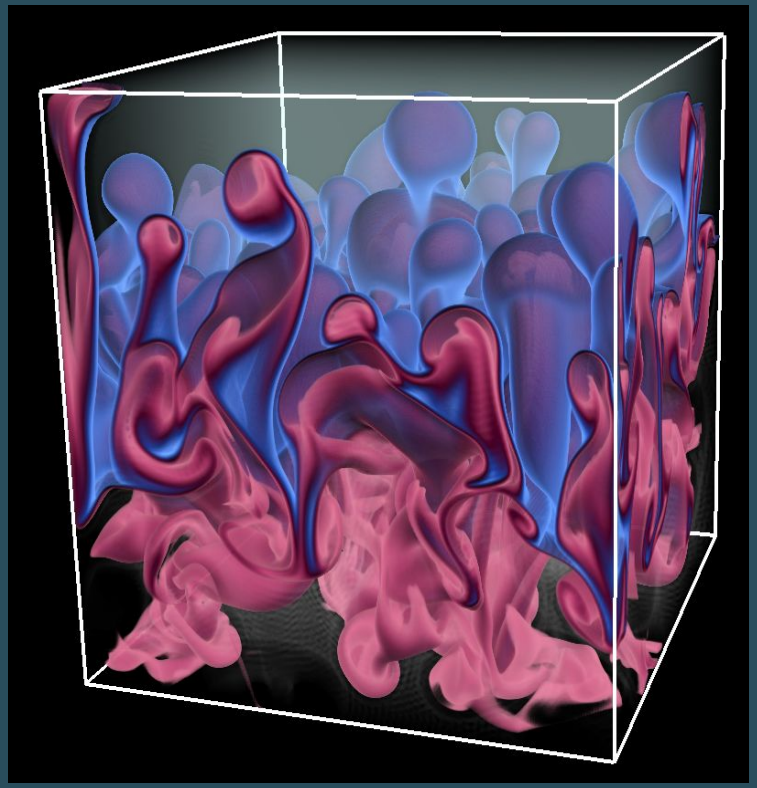
Easily record all edits (and other fine changes) in python, can be run at a later time to reproduce the plot style

```
f=open(''script.py'',''wt'')
WriteScript(f)
f.close()
```

```
Source(''script.py'').
```

```
# Create plots
# Create plot 1
OpenDatabase("localhost:/RAID2/mmstastn/bartry23deepright/*.nc database")
AddPlot("Volume", "t", 0, 0)
atts = VolumeAttributes()
#SetPlotOptions(atts)
silr = SILRestriction()
silr.TurnOffAll()
silr.TurnOnSet("domain1")
SetPlotSILRestriction(silr, 0)

SetActivePlots(0)

DrawPlots()

# Set the view
view = View3DAttributes()
view.viewNormal = (0.387158, -0.852159, 0.352042)
view.focus = (0.1, 0.025, 0.1)
view.viewUp = (-0.0892699, 0.34538, 0.934208)
view.viewAngle = 30
view.parallelScale = 0.143308
view.nearPlane = -0.286616
view.farPlane = 0.286616
view.imagePan = (0.0481336, 0.0831579)
view.imageZoom = 1
view.perspective = 1
view.eyeAngle = 2
view.centerOfRotationSet = 0
view.centerOfRotation = (0.1, 0.025, 0.1)
view.axis3DScaleFlag = 0
view.axis3DScales = (1, 1, 1)
view.shear = (0, 0, 1)
view.windowValid = 1
SetView3D(view)

# Set the annotation attributes
annot = AnnotationAttributes()
```





References

- Ahrens, James, Geveci, Berk, Law, Charles, “ParaView: An End-User Tool for Large Data Visualization.” *Visualization Handbook* (2005), Elsevier, 2005, ISBN-13: 978-0123875822
- Bloch, Cezary. " Rendering of multivariate 3D volume data." *Volumerendering.republika.pl*. N.p., 2006. Web. 03 June 2017.
- Childs, Hank. et. al. “VisIt.” *High Performance Visualization Chapman & Hall/CRC Computational Science*. (2012) doi:10.1201/b12985-21
- Childs, Hank. "VisIt Python Interface Manual v2.5.2." Ernest Orlando Lawrence Berkeley National Laboratory, 12 Sept. 2012. Web. 31 May 2017. <http://vis.lbl.gov/~hrchilds/VisItPythonManual.pdf>.
- "Choosing Colormaps." *Matplotlib 2.0.2 documentation*. N.p., n.d. Web. 31 May 2017. <http://matplotlib.org/users/colormaps.html>.
- Deepwell, David, and Marek Stastna. "Mass transport by mode-2 internal solitary-like waves." *Physics of Fluids* 28.5 (2016): n. pag. *AIP*. Web. 31 May 2017.
- Penney, Jared, and Marek Stastna. "Direct numerical simulation of double-diffusive gravity currents." *Physics of Fluids* 28.8 (2016): n. pag. *AIP*. Web. 31 May 2017.
- VanderPlas, Jake. "How Bad Is Your Colormap?" *Pythonic Perambulations*. N.p., 16 Oct. 2014. Web. 31 May 2017. <https://jakevdp.github.io/blog/2014/10/16/how-bad-is-your-colormap/>.
- "VisIt Tutorial Basics." *VisIt Users*. N.p., 11 Nov. 2016. Web. 31 May 2017. <http://visitusers.org/index.php?title=VisIt-tutorial-basics>.
- "VisIt User's Manual." Lawrence Livermore National Laboratory, n.d. Web. 31 May 2017. <https://wci.llnl.gov/codes/visit/1.5/VisItUsersManual1.5.pdf>.