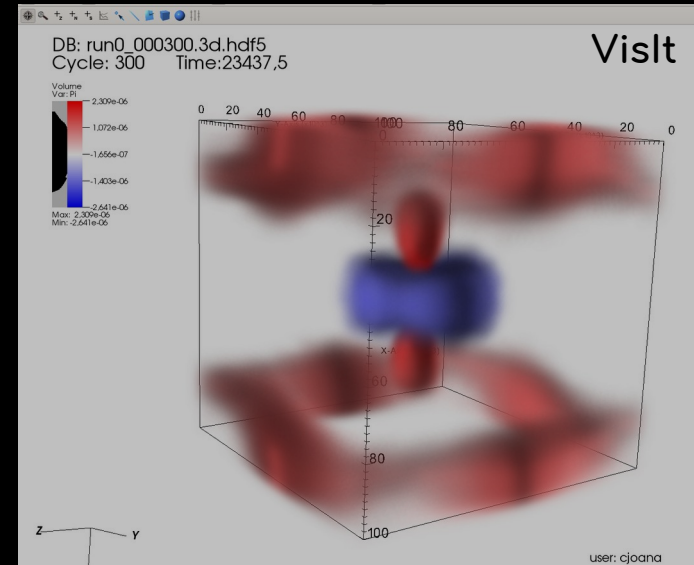


Visualisation tools for GRChombo

```
import yt

dfn = './data/run0p_000300.3d.hdf5'
ds = yt.load(dfn)
L, _, _ = ds.domain_width

normal = 'z'
var = "K"
center = [L/2, L/2, L/2]
plot = yt.SlicePlot(ds, normal=normal,
                    fields=var, center=center)
plot.set_cmap(var, 'RdBu_r')
plot.save('./plots/{v}_slice.png'.format(v=var))
```



Cristian Joana - UCLouvain (CURL)

GRChombo Meeting 26/03/2020

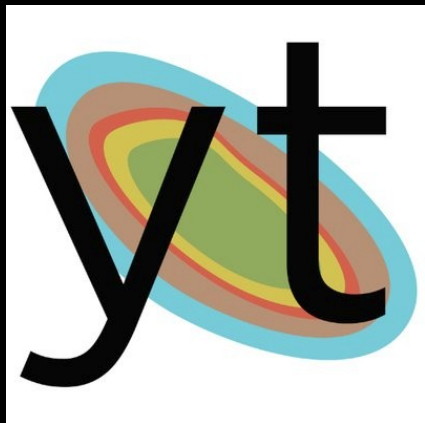
Visualisation tools for GRChombo

- Using YT (python)
- Using Visit (GUI)
- Installation
- Utilities
- Examples

For movies → Josu slides 2019



YT - project



+



```
import yt

dfn = './data/run0p_000300.3d.hdf5'
ds = yt.load(dfn)
L, _, _ = ds.domain_width

normal = 'z'
var = "K"
center = [L/2, L/2, L/2]
plot = yt.SlicePlot(ds, normal=normal,
                    fields=var, center=center)
plot.set_cmap(var, 'RdBu_r')
plot.save('./plots/{v}_slice.png'.format(v=var))

['./plots/K_slice.png']
```

<https://yt-project.org/docs/dev/>

(documentation)

<https://yt-project.org/community.html>

(mailing list, Slack, etc.)



YT - Installation

- Via conda:

```
$ conda install -c conda-forge yt
```

- Via PiP:

```
$ pip install yt
```

- Via github repository:

```
$ git clone https://github.com/yt-project/yt
```

```
$ cd yt && python setup.py install
```



YT – Loading files

Loading hdf5 file

```
# define dataset's path  
dfn = './data/run0p_000300.3d.hdf5'  
  
#load dataset  
ds = yt.load(dfn)  
  
# Equivalent to:  
# ds = yt.frontends.combo.ChomboDataset(dfn)
```



YT – Loading files

Loading hdf5 file

```
# define dataset's path
dfn = './data/run0p_000300.3d.hdf5'

units_override = {"length_unit": (1.0, "l_pl"),
                  "time_unit": (1.0, "t_pl"),
                  "mass_unit": (1.0, "m_pl")}
unit_system = 'planck'

# load dataset
ds = yt.load(dfn,
             unit_system=unit_system, units_override=units_override)

# ds = yt.frontends.combo.ChomboDataset(dfn,
#   unit_system=unit_system, units_override=units_override)
```



YT – Handling data

Example: extraction of data & AMR coordinates

```
import matplotlib.pyplot as plt

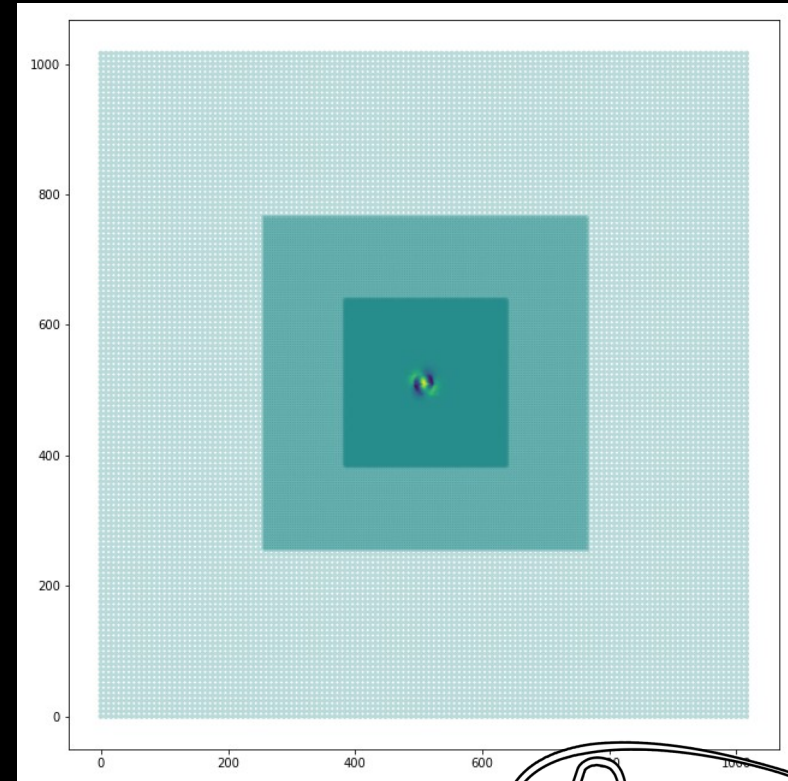
#Loading values of 'Avec0' and coordinates (taking into account AMR)
reg = ds2.all_data() # indexing data as flat array (contain all variables)
values = reg['Avec0'] # flat array that contains variable "Avec0"

# Loading grid-cell centered coordinates
X = reg['x']
Y = reg['y']
Z = reg['z']

# define position as bin-border of the grid
xpos = reg['x'] - reg['dx']/2
ypos = reg['y'] - reg['dy']/2
zpos = reg['z'] - reg['dz']/2

_, _, L = ds.domain_width
c_z = zpos[zpos >= L*0.05][0] #choosing grid-coord of interest
mask_cslice = np.array(zpos == c_z, dtype=bool) # mask for data selection

# plot
fig, ax = plt.subplots(figsize=(9,9))
ax.scatter(xpos[mask_cslice], ypos[mask_cslice], c=values[mask_cslice],
          s=20, edgecolor='', alpha=0.3)
plt.tight_layout()
plt.savefig("./plots/plot_amr.png")
```



YT – Handling data



Setting up derived variables

```
def _cell_volume(field, data): # 'field', 'data' arguments needed
    vol = data["chi"]**(-1.5) * data["dx"]**(3)
    return vol

ds.add_field(('chombo', 'cell_vol'), sampling_type="cell",
            units = "l_pl**3", function=_cell_volume)

reg = ds.r[:,::,:] # flat array
con_L = np.sum(reg['dx']**3)**(1/3)
eff_L = np.sum(reg['cell_vol'])**(1/3)

print("conformal / effective grid-size: {c:.2e} {e:.2e}".format(c=con_L, e=eff_L))
print("domain L", ds.domain_width[0])
```

```
conformal / effective grid-size: 1.00e+05 l_pl 2.79e+05 l_pl
domain L 100000.0 code_length
```

NB: `dx`, `dy`,... & `x`, `y`... are automatically yt-generated grid variables.

YT – Handeling data

Setting new fields : gradients

```
ds.add_gradient_fields(('chombo', 'K')) # uses second-order centered differences
```

```
[('chombo', 'K_gradient_x'),  
 ('chombo', 'K_gradient_y'),  
 ('chombo', 'K_gradient_z'),  
 ('chombo', 'K_gradient_magnitude')]
```

```
print(reg['K_gradient_x'])
```

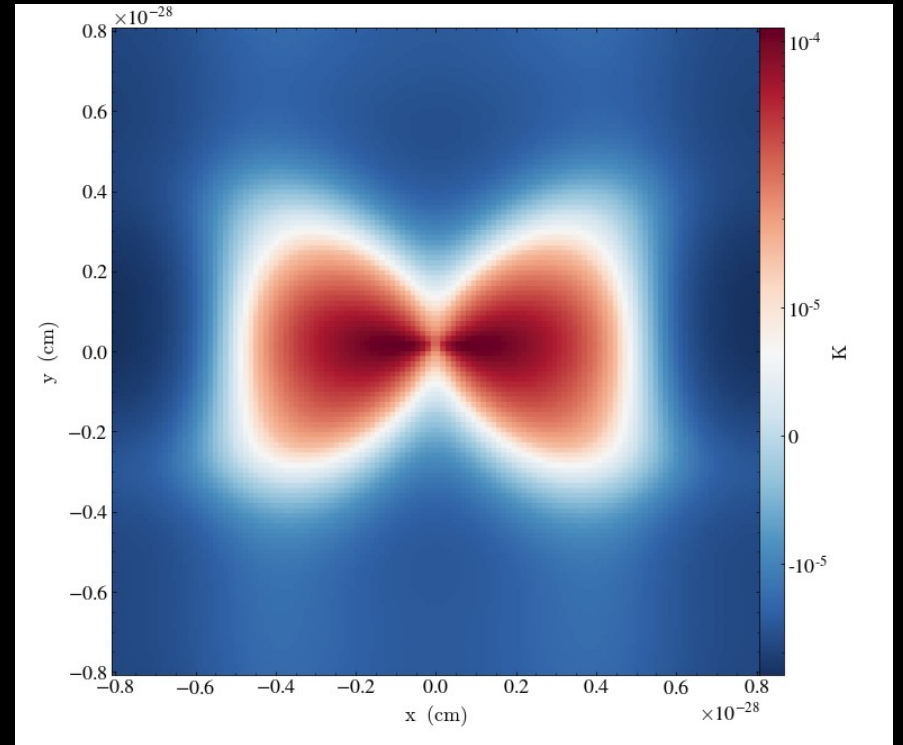
```
[ 7.49475932e-12  7.48383738e-12  7.47916149e-12 ... -7.53537997e-12  
 -7.50883771e-12 -7.48724252e-12] 1/l_pl
```



YT – Plotting utilites

yt.SlicePlot()

```
dfn = './data/run0p_000300.3d.hdf5'  
ds = yt.load(dfn)  
L, _, _ = ds.domain_width  
  
normal = 'z'  
var = "K"  
center = [L/2, L/2, L/2]  
plot = yt.SlicePlot(ds, normal=normal,  
                    fields=var, center=center)  
plot.set_cmap(var, 'RdBu_r')  
plot.save('./plots/{v}_slice.png'.format(v=var))  
  
['./plots/K_slice.png']
```



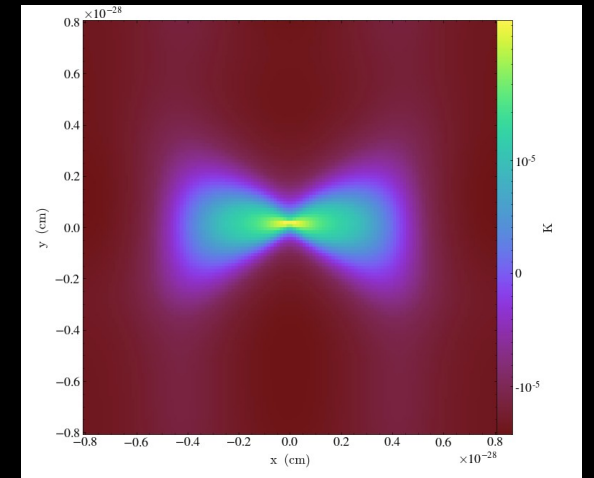
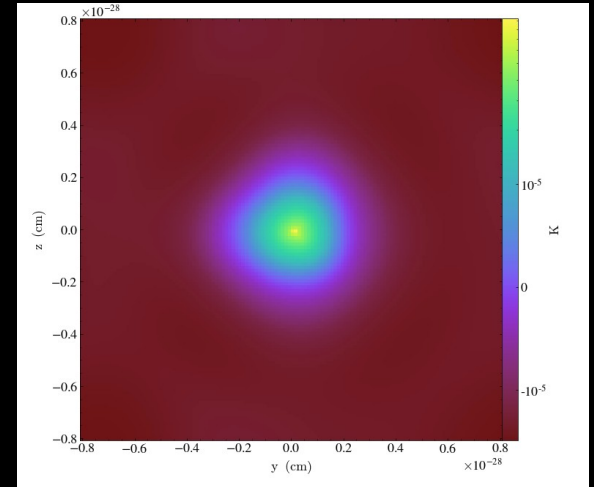
→ <https://yt-project.org/doc/visualizing/plots.html>

YT – Plotting utilites



yt.ProjectionPlot()

```
dfn = './data/run0p_000300.3d.hdf5'  
ds = yt.load(dfn)  
ds.add_field(('chombo', 'cell_vol'),  
            sampling_type="cell", units = "l_pl**3", function=_cell_volume)  
  
var = "K"  
plot = yt.ProjectionPlot(ds, fields=var, axis='x',  
                        method='integrate', weight_field='cell_vol')  
plot.save('./plots/{v}_projection_x.png'.format(v=var))  
plot = yt.ProjectionPlot(ds, fields=var, axis='z',  
                        method='integrate', weight_field='cell_vol')  
plot.save('./plots/{v}_projection_z.png'.format(v=var))
```



→ <https://yt-project.org/doc/visualizing/plots.html>

(non) YT – Plotting

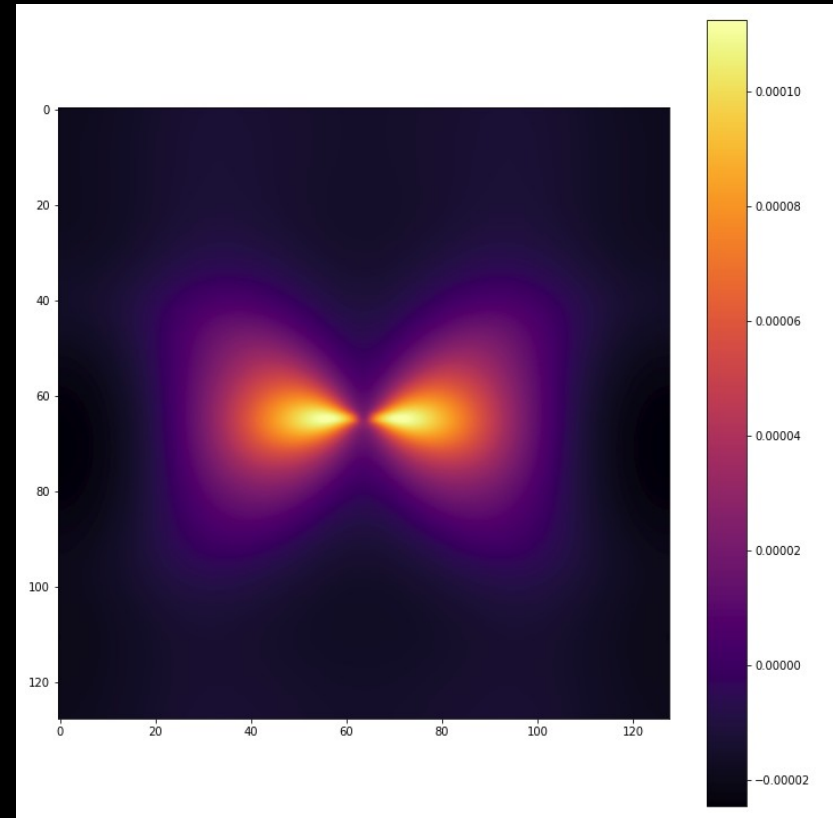
Via matplotlib

```
import matplotlib.cm as cm
import matplotlib.pyplot as plt

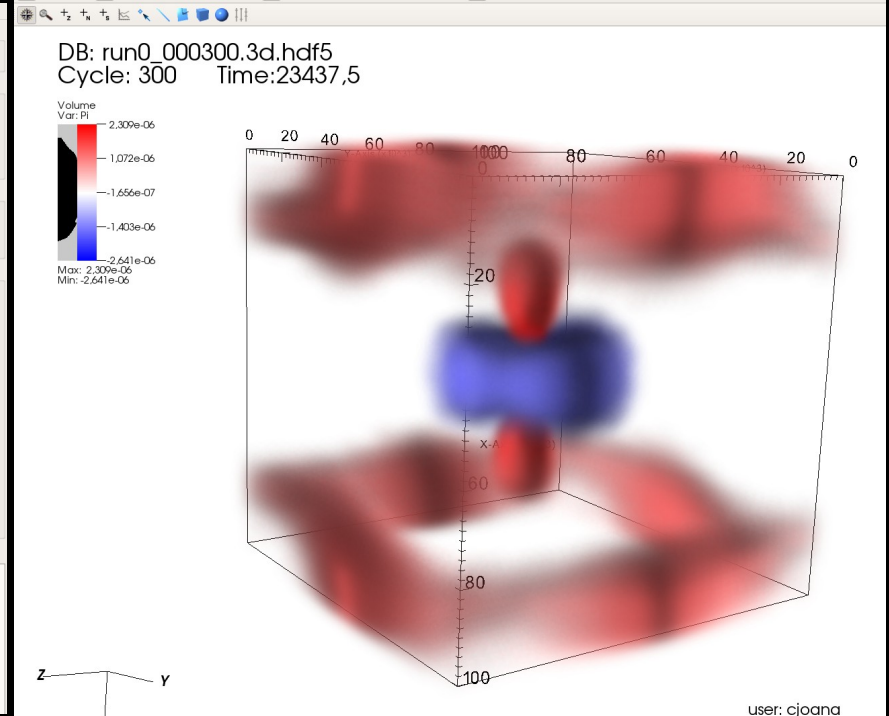
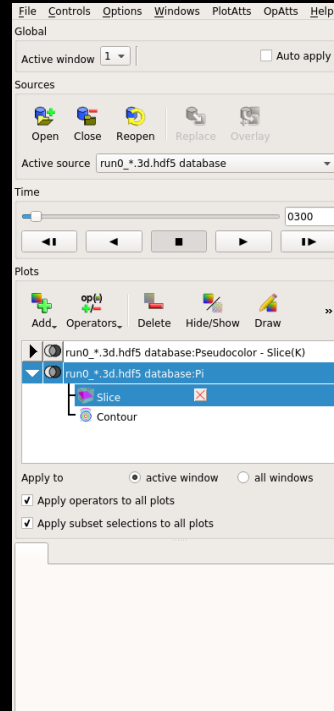
%matplotlib inline

dfn = './data/run0p_000300.3d.hdf5'
ds = yt.load(dfn)
L, _, _ = ds.domain_width
slc = ds.r[:,::128j,::128j, L/2]

fig, ax = plt.subplots(figsize=(10,10))
plot1 = ax.imshow(slc['K'], interpolation='spline16',
                  cmap=cm.inferno)
fig.colorbar(plot1, ax=ax)
plt.savefig('./plots/K_slice_matplotlib.png')
```



Visit



Download & documentation:

<https://wci.llnl.gov/simulation/computer-codes/visit/downloads>

https://visit-sphinx-github-user-manual.readthedocs.io/en/develop/gui_manual/



Visit - Installation



You probably already have it installed. If not, **don't worry**, you will manage :)

+ info:

<https://github.com/GRChombo/GRChombo/wiki/Visualising-outputs>

https://github.com/GRChombo/GRChombo/wiki/files/grchombo_ubuntu.pdf

shared by Leonard Werneck



1.9 Installing VisIt

shared by Leonard Werneck

VisIt is the software used by the developers of GRChombo to make beautiful plots and visual simulations. It is compatible with .hdf5 files, so it is a nice idea to install it.

Let us start by going to the following webpage

<https://wci.llnl.gov/simulation/computer-codes/visit/executables>

and downloading both the install script (copy the page to a file, in my case I have created the file `visitinstall.sh`) and the Ubuntu 14.04 executable. At the time of writing, version 2.13.0 was downloaded.

Then give permission so that the file can be executed

```
chmod 755 visitinstall.sh
```

and type

```
./visitinstall.sh 2.13.0 linux-x86_64-ubuntu14 /usr/local/visit
```

When prompted, choose the “No System Configuration” option. Then open your `~/.bashrc` file again and include at the bottom of the file the line

```
export PATH="/usr/local/visit/bin:$PATH"
```

Save the file and close it. Close all terminal windows and open a new one.

Visit - Installation



In Ubuntu: (not tested)

Last version avail. : Visit 3.1.1 (Feb 2020)

StackExchange Search on Ask Ubuntu...

Home
Questions
Tags
Users
Unanswered

Visit 2.1.3 Ubuntu 18.10 setup without root

0

- go to the download site: <https://wci.llnl.gov/simulation/computer-codes/visit/executables>
- download the "Linux - x86_64 64 bit" version
- extract:

```
tar xvf visit2_13_3.linux-x86_64-ubuntu18.tar.gz`
```
- run:

```
./visit2_13_3.linux-x86_64/bin/visit
```

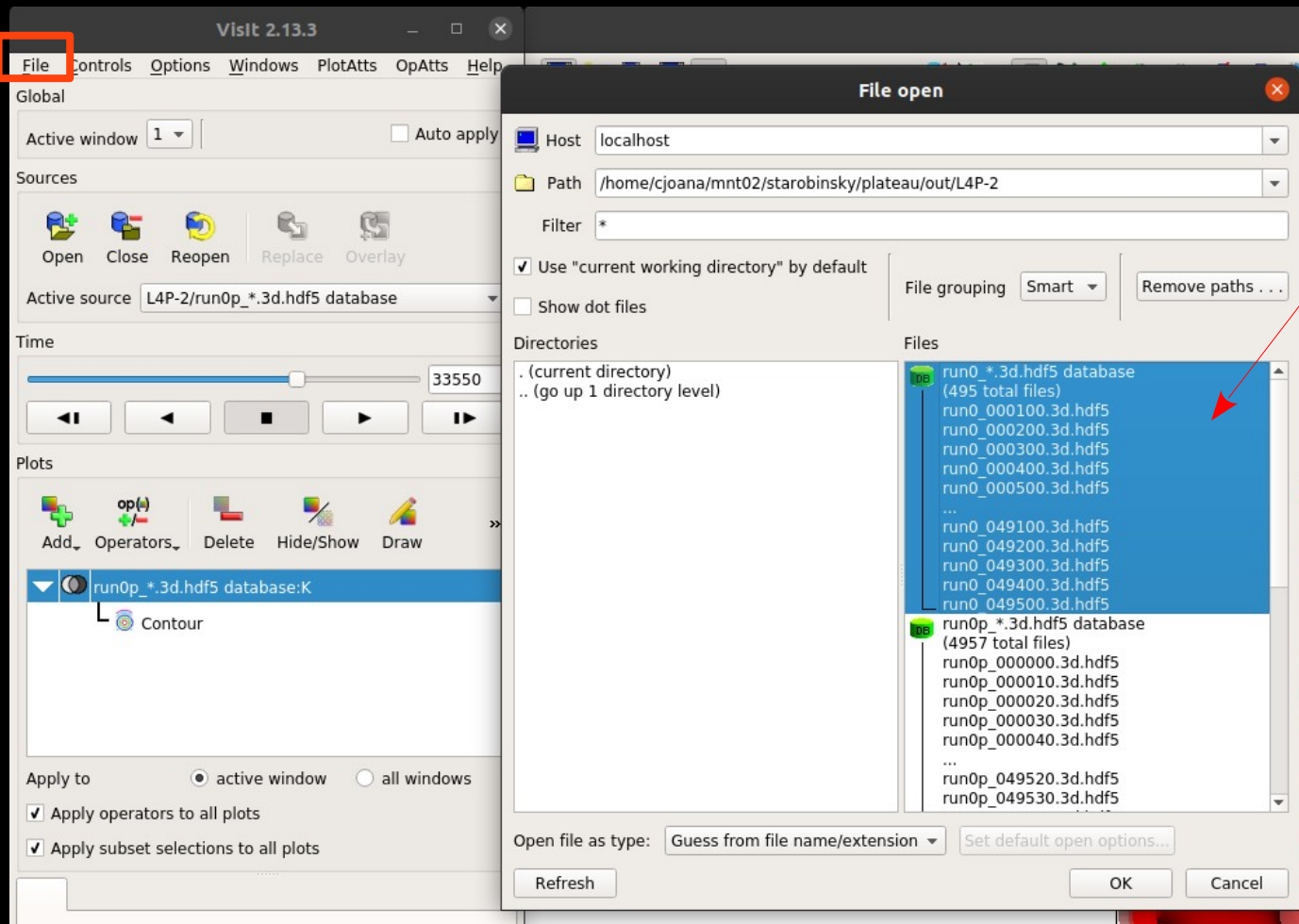
I have for example used it at: <https://stats.stackexchange.com/questions/376361/how-to-find-the-sample-points-that-have-statistically-meaningful-large-outlier-r>

share improve this answer

answered May 1 '19 at 9:31

 **Ciro Santilli** 新疆改造中

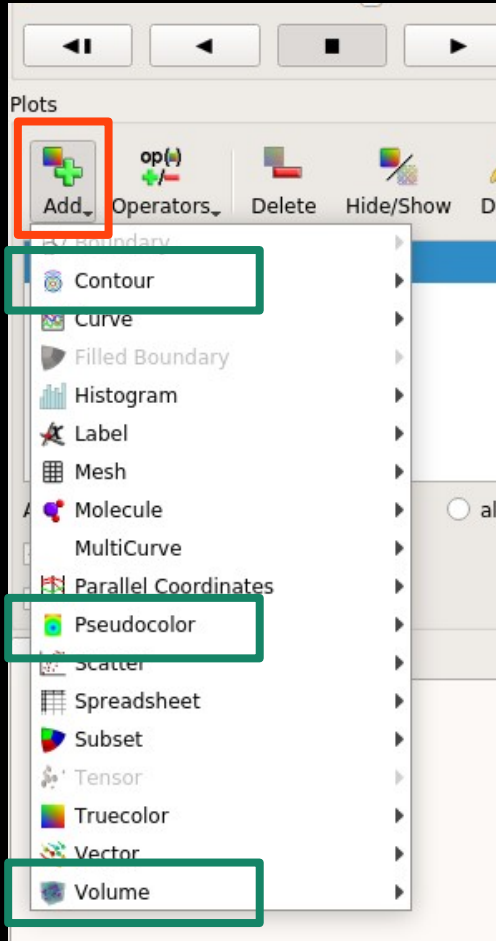
Visit – GUI commands



Select multiple files with the same prefix

`./<prefix>_000000.hdf5`

Visit – GUI commands

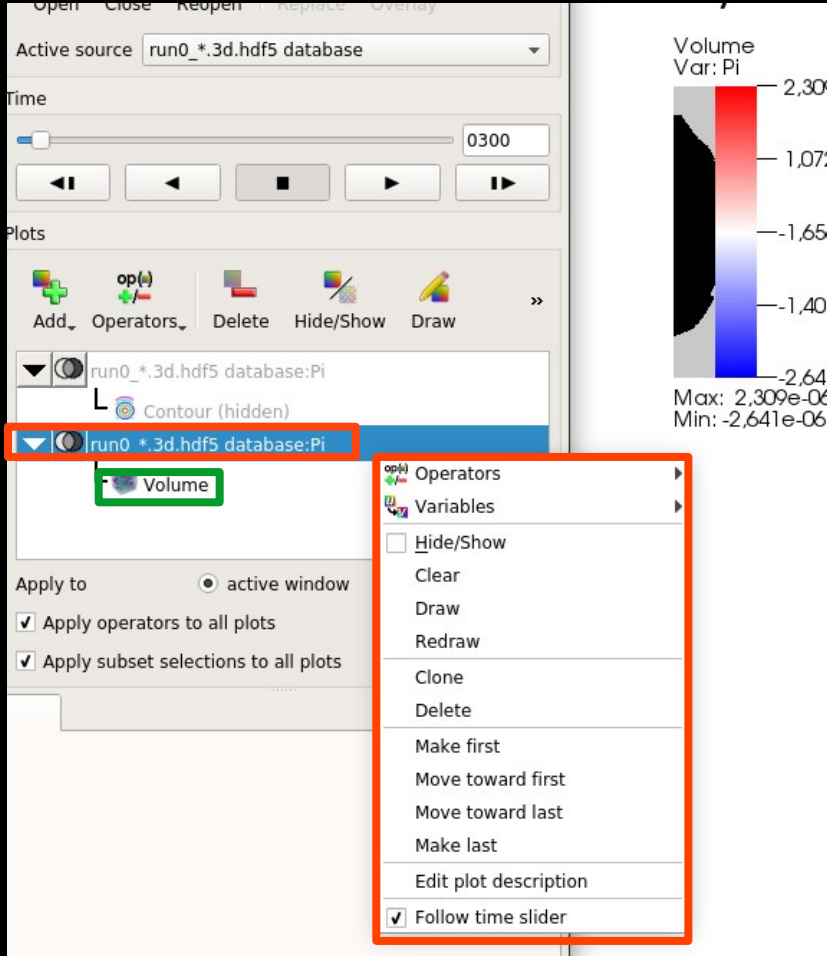


Different plotting functions:

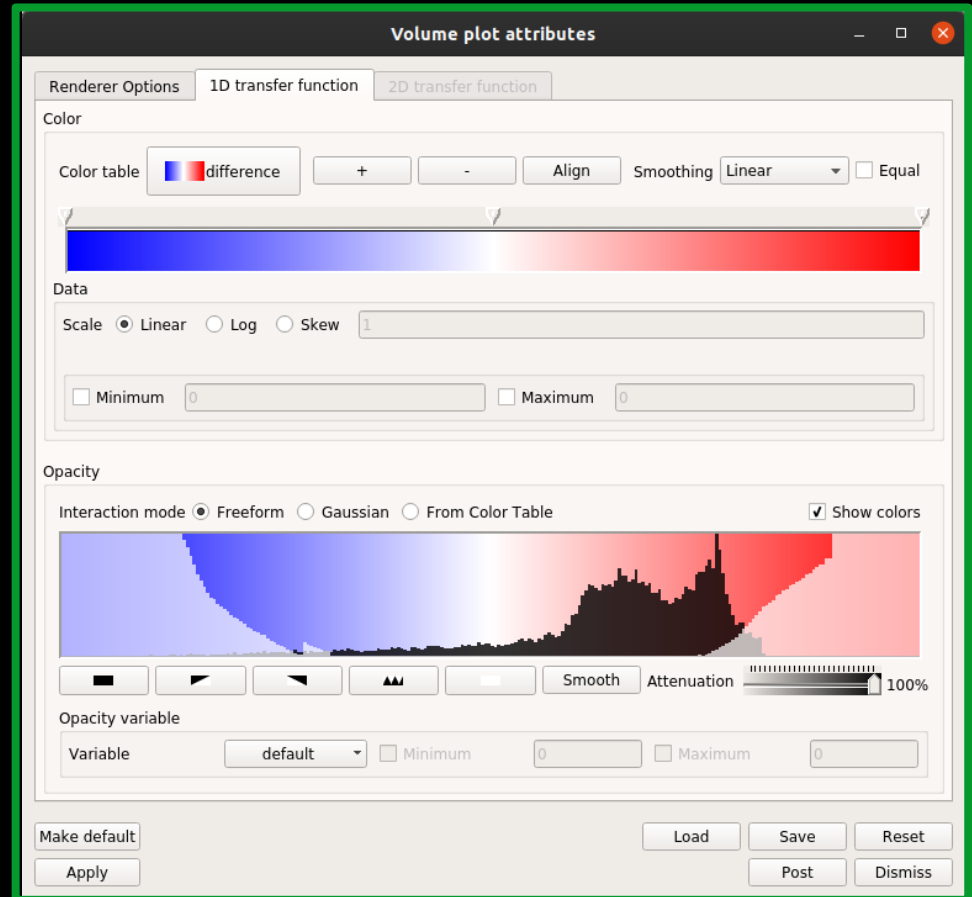
- Contour, for 2D & 3D plots
- Pseudocolor, mainly for 2D plots.
- Volume, mainly for 3D plots

Feel free to play with the other options!

Visit – GUI commands



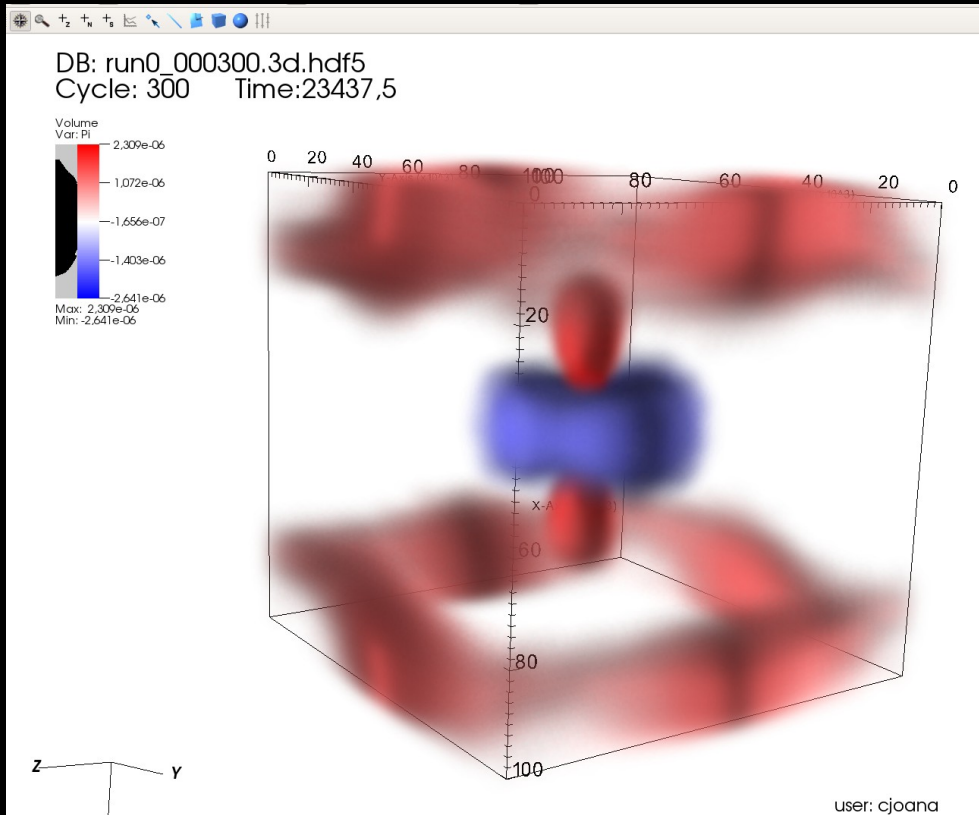
The screenshot shows the Visit GUI interface. The top menu bar includes 'Open', 'Close', 'Reopen', 'Replace', and 'Overlay'. The 'Active source' dropdown is set to 'run0_*.3d.hdf5 database'. The 'Time' slider is at 0300. The 'Plots' panel shows a tree view with 'run0_*.3d.hdf5 database:Pi' selected, and a 'Volume' plot highlighted with a green box. A context menu is open over the 'Volume' plot, listing various actions: Operators, Variables, Hide/Show, Clear, Draw, Redraw, Clone, Delete, Make first, Move toward first, Move toward last, Make last, Edit plot description, and Follow time slider (checked).



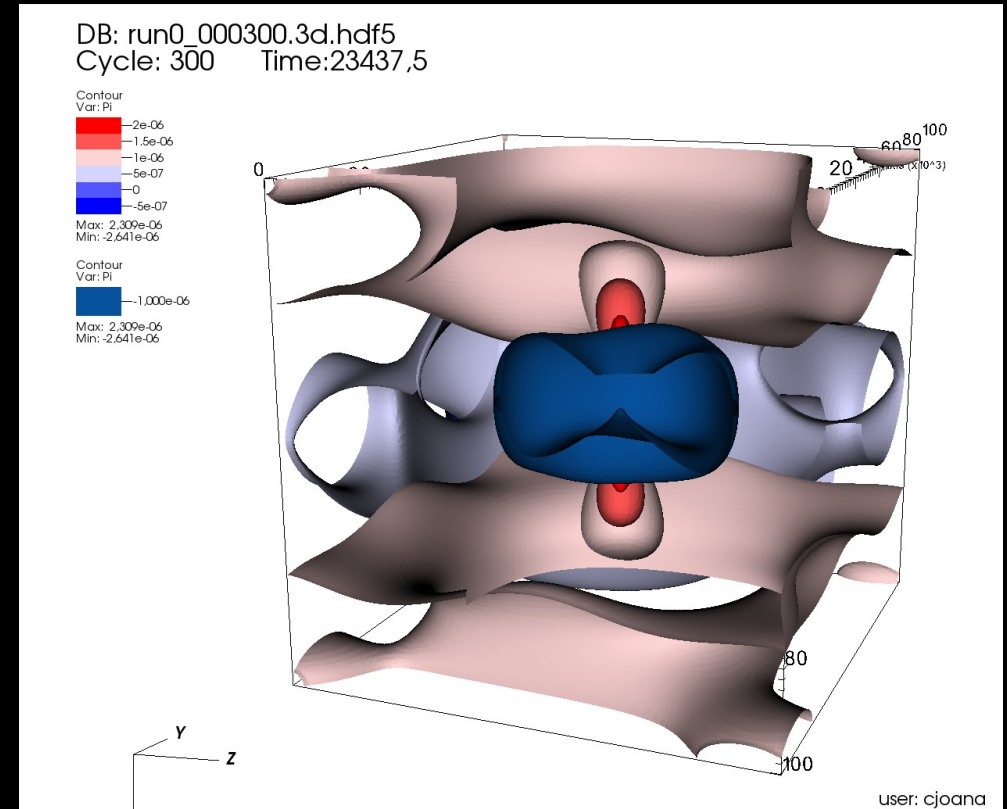
The 'Volume plot attributes' dialog box is shown, highlighting the '1D transfer function' tab. The 'Color' section includes a 'Color table' dropdown set to 'difference', a color gradient bar, and options for '+', '-', 'Align', 'Smoothing', 'Linear', and 'Equal'. The 'Data' section has 'Scale' options: Linear (selected), Log, and Skew, with a value of 1. There are also 'Minimum' and 'Maximum' input fields, both set to 0. The 'Opacity' section includes 'Interaction mode' options: Freeform (selected), Gaussian, and From Color Table, with a 'Show colors' checkbox checked. A histogram and a color gradient bar are visible, along with 'Smooth' and 'Attenuation' controls. The 'Opacity variable' section has a 'Variable' dropdown set to 'default' and 'Minimum' and 'Maximum' input fields, both set to 0. At the bottom, there are buttons for 'Make default', 'Apply', 'Load', 'Save', 'Reset', 'Post', and 'Dismiss'.

Visit – GUI commands

Volume (3D)

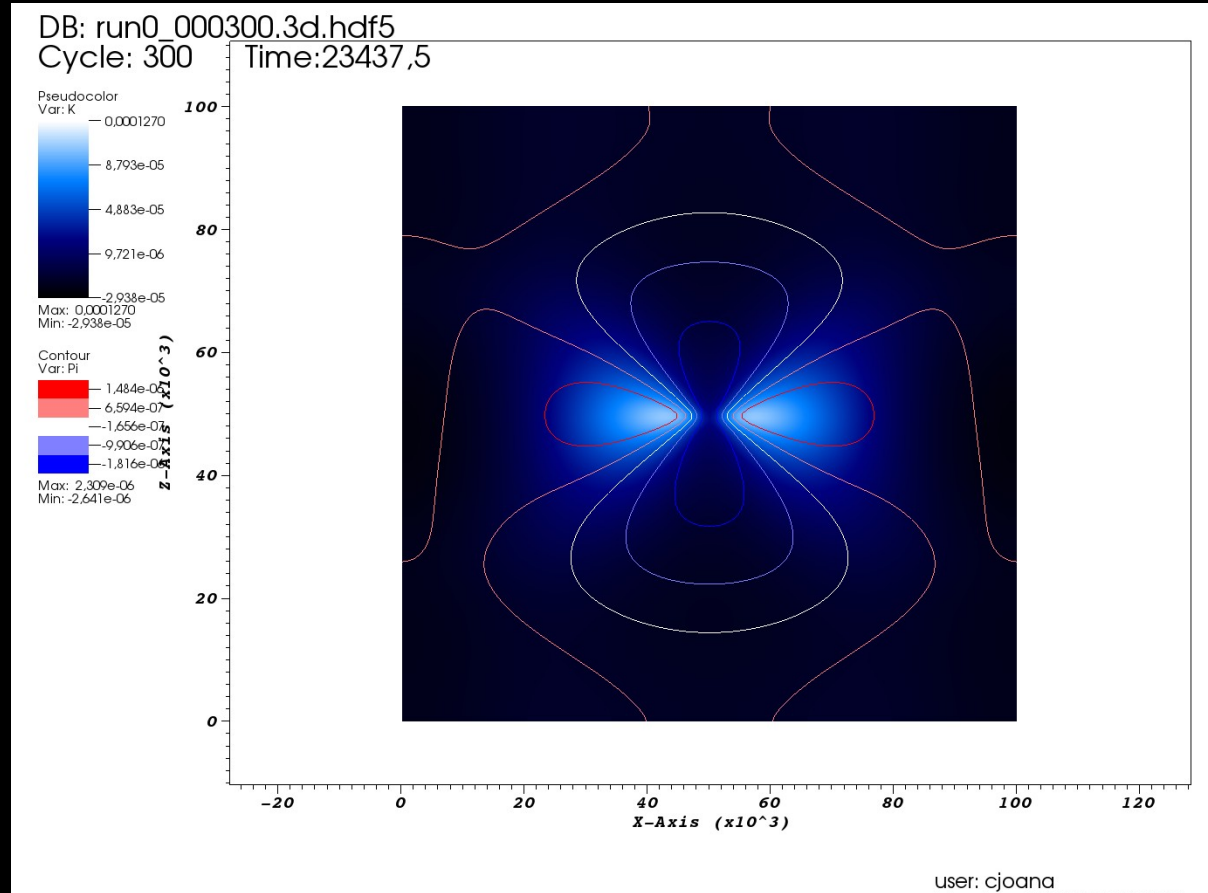
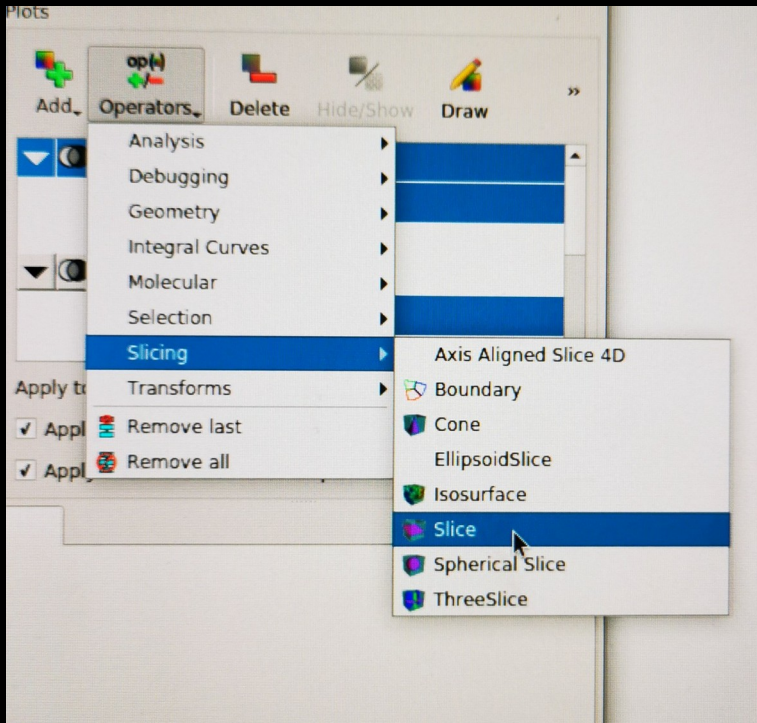


Contour (3D)



Visit – GUI commands

Pseudocolor + Contour (2D)



Visit – GUI commands



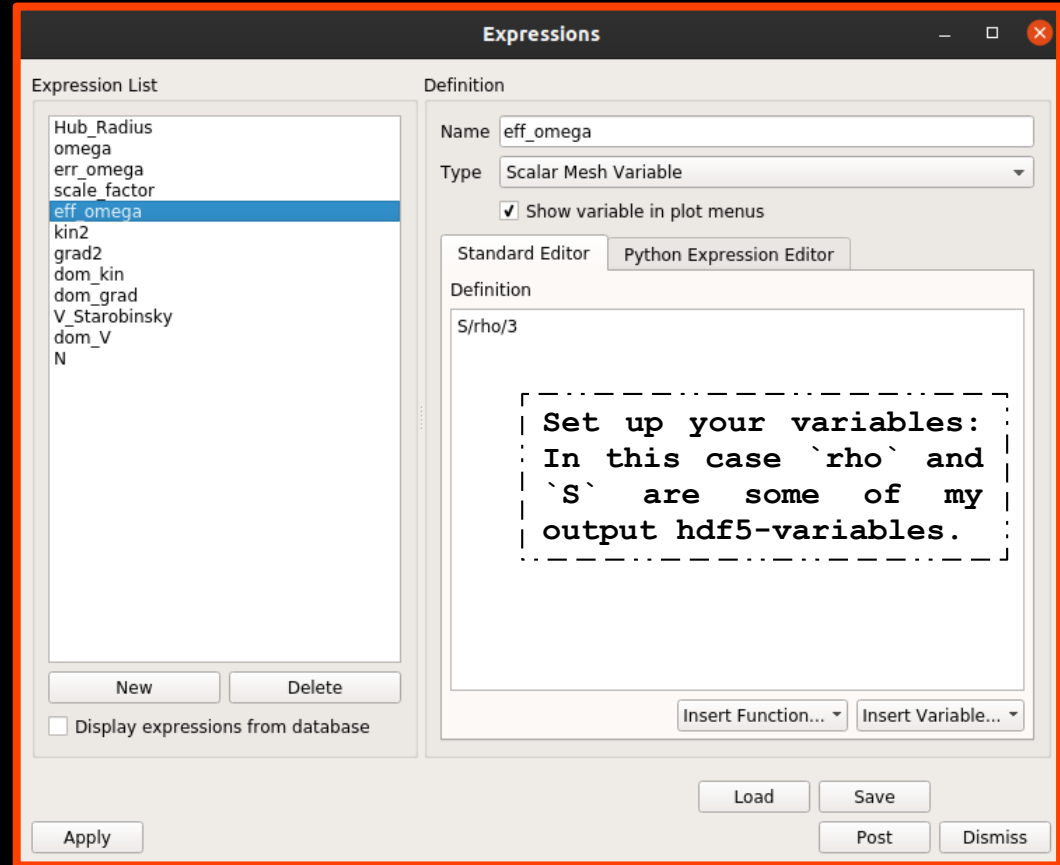
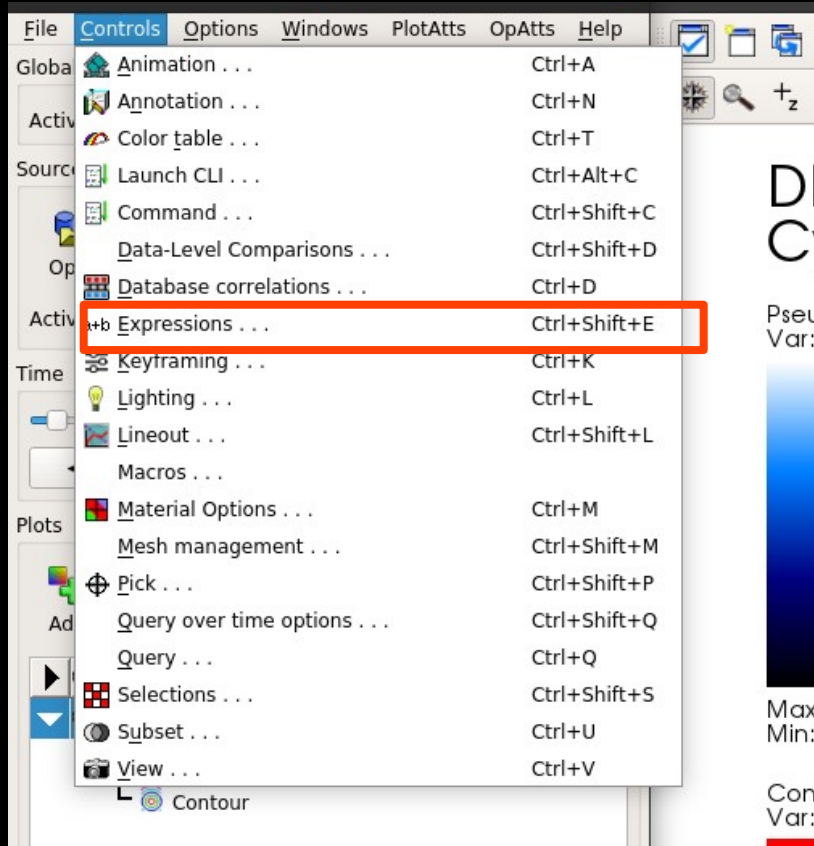
The screenshot displays the Visit GUI interface. On the left, the 'Sources' panel shows the active source as 'run0_*.3d.hdf5 database'. Below it, the 'Time' panel has a slider set to 0300. The 'Plots' panel lists several plots, with 'run0_*.3d.hdf5 database:Pi' selected. A 'Slice operator attributes' dialog box is open, showing settings for a slice operation. The dialog has a title bar 'Slice operator attributes' and a close button. It is divided into sections: 'Normal' with radio buttons for 'Orthogonal', 'X Axis', 'Y Axis' (selected), 'Z Axis', and a 'flip' checkbox; 'Arbitrary' with a text field '0 -1 0'; 'Theta-Phi' with a text field '0 0'; 'Origin' with radio buttons for 'Point', 'Intercept', 'Percent' (selected), 'Zone', and 'Node', and a 'Percent' slider set to 50; 'Up Axis' with a checked 'Project to 2D' checkbox and a 'Direction' text field '0 0 1'; and 'Interactive' with a checked checkbox. At the bottom of the dialog are buttons for 'Make default', 'Load', 'Save', 'Reset', 'Apply', 'Post', and 'Dismiss'. The background shows the main GUI with a top status bar displaying 'DB: run0_000300.3d', 'Cycle: 300', and 'Time:'.

Suggestion:

- 1) Go to Slice settings
- 2) Select Orthogonal axis (e.g. Y)
- 3) Chose `Percent` to select the cord.

Visit – GUI commands

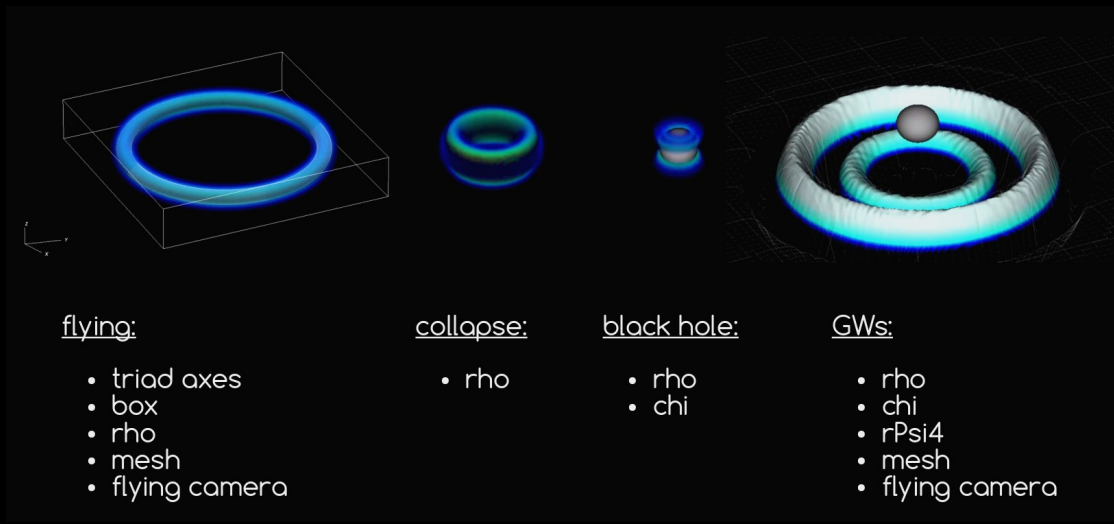
Add derived variable:



Visit – Script mode

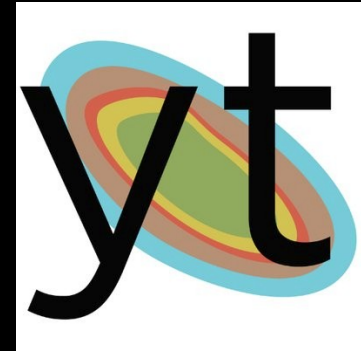
- VisIt can also be used in ``script`` mode. But this is not covered on these slides.

- In ``script`` mode, beautiful animation can be made:



→ See Josu's slides from last year! (2019)

Questions ?





Back up