Tools -> Start trace > incremental
then stop trace
Gives Python code for everything you did

ParaView Walkthrough

A tool to interactively explore 3D data sets

Note, you can use ParaView interactively, or through a built-in Python Shell:

- View / Python Shell
- Tools / Start Trace (Select “Show Incremental Trace” to see in realtime)

We will use ParaView interactively:

1. File / Open the Density.vtk and Temperature.vtk files
2. Turn the “eye” on to view Density
3. Change Representation to Volume, and the Coloring to Density.
4. Do the same for Temperature
5. Change the Color Map for Density and/or Temperature
   - Coloring → Edit → Choose Presets (folder with heart)
   - I kept the blue-red for temperature but tweaked the curve a bit
   - I changed the Density to green and inverted (white circle with black triangle)
   - Click Apply when you’re happy with the color choices
6. Add a Slice to the Density (from the top bar)
   - Grabbing from edge moves it up and down; grabbing the arrow tilts it
   - Click Apply
7. Hide the Density and Temperature, to view only the Slice
8. Add Contour to Slice (from the top bar)
   - In Value Range select “add a range of value”
   - I chose from -2 to 2 with 20 steps
   - Click Apply
9. Add Clip to Temperature (from the top bar)
   - Copy the same positions from the Density Slice to the Temperature Clip
   - Move it down just a touch so that the contours a more visible
   - Click apply
10. Add a Filters / Data Analysis / Calculator to Temperature
    - I did `10^Temperature`, and saved as `Linear`
    - Add a Clip to this too, to compare with the original data
    - Change Representation to Surface
    - Change Coloring to Linear (and back for comparison)
11. Add Filters / Data Analysis / Programmable Filter by selecting both Density and Temperature
    - Add the following Script. (You may need to swap inputs order depending on how you loaded the data):

```
D = inputs[0].PointData['Density']
T = inputs[1].PointData['Temperature']
output.PointData.append(D*T, 'multiply')
output.PointData.append(D/T, 'divide')
output.PointData.append(D+T, 'add')
output.PointData.append(cos(T), 'cosT')
output.PointData.append(10**T, 'linearT')
```

    - Change the “Coloring” to one of these new arrays
    - Apply a Threshold (or other) to this (from the top bar)
12. File / Export Scene
    - I chose an eps format ; this would need a lot of work to make it publication ready
    - You can also export to x3dom, which is a format currently used in astrophysical journals.