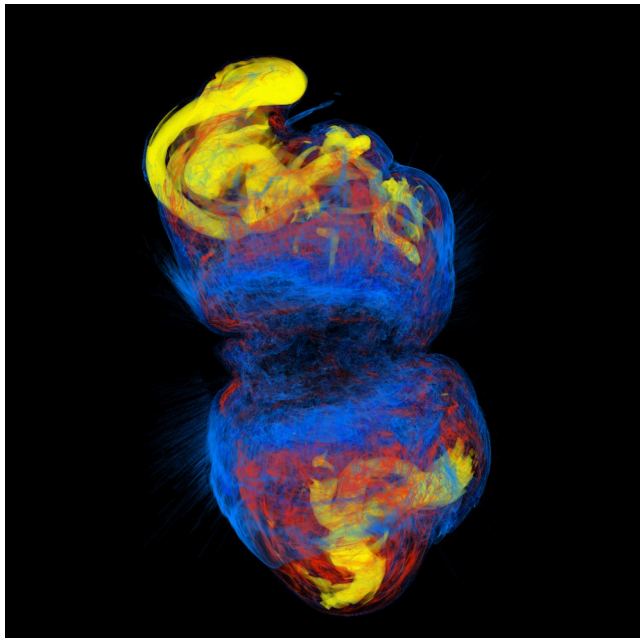
Postprocrocessing data in Cactus

Roland Haas, AEI

Postprocessing vs. visualization

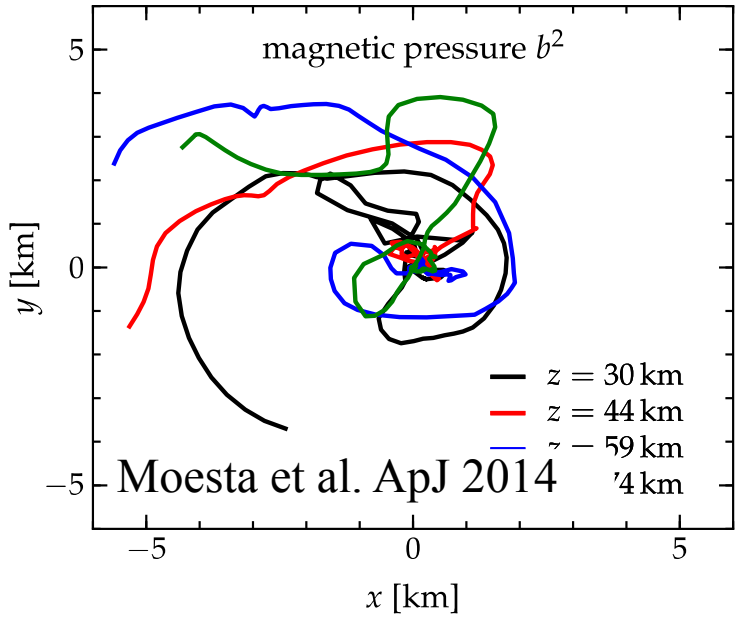
- quantitative analysis of simulation results
- compute extra quantities after simulation finishes
- complex calculations involving multiple variables, timesteps, refinement levels
- needs “human touch” and cannot be automated during runtime



Moesta et al. ApJ 2014

Postprocessing vs. visualization

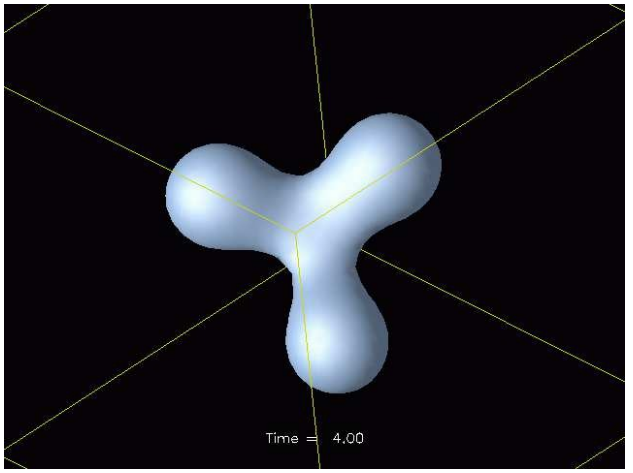
- quantitative analysis of simulation results
- compute extra quantities after simulation finishes
- complex calculations involving multiple variables, timesteps, refinement levels
- needs “human touch” and cannot be automated during runtime




```
<< SimulationTools`

$SimulationPath =
  {$HomeDirectory <>
    "/simulations/sliced_data/mode-analysis/"};

$Simulation = "sherry/E25b12z1-p2sc3w-ft/z0-15";

ReadIterations[$Simulation, "entropy", "xy"]

{5 900 288, 5 902 336, 5 904 384, 5 906 432, 5 908 480,
 5 910 528, 5 912 576, 5 914 624, 5 916 672, 5 918 720,
 5 920 768, 5 922 816, 5 924 864, 5 926 912,
 5 928 960, 5 931 008, 5 933 056, 5 935 104,
 5 937 152, 5 939 200, 5 941 248, 5 943 296}
```

Postprocessing in Cactus

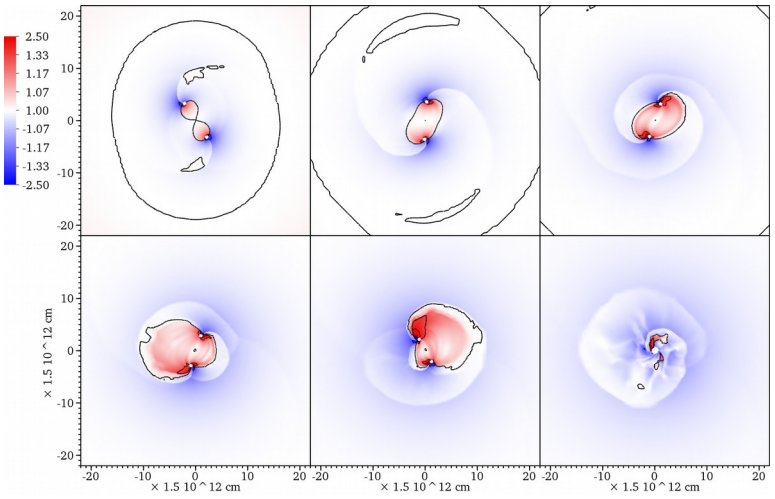
- specialized postprocessing codes
  - CCE (PITTNullCode)
  - Event horizon finder (Diener)
  - MCMC poset analysis (Rideout)
- GW analysis tools
  - pyGWAnalysis (in ET)
  - SimulationTools (Hinder, public)
  - PostCactus (Kastaun, TBA)
  - <enter your name here>
- Dataset readers
  - SimulationTools, PostCactus
  - SciData (Radice, public)
  - Parma-postprocessessing tools (De Pietri, TBA)
  - FileReader (in ET), ReadInterpolate (Haas, public)
  - Replay (Haas, currently private)



Diener, CQG 2003

Visualization++

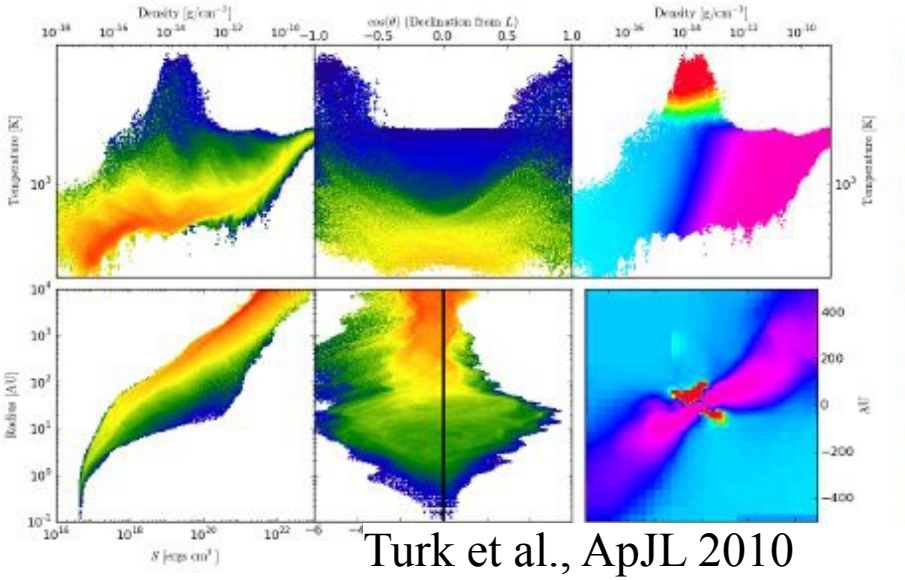
- VisIt
  - basic point-wise operations
  - python-based data analysis
- yt
  - designed to postprocess Enzo data
  - basic reader support for Carpet
- matplotlib/numpy/scipy
  - no direct support for anything but 1d ASCII data
  - <add you name here>



Bode et al., ApJ 2011

Greener grasses...

- Enzo/yt
  - arithmetic on variables, incl. derivatives
  - subset selection based on shapes, thresholds
  - reductions of data: mimimum location, center of mass
- SpEC/ApplyObservers
  - swiss army knife postprocessing
  - “replays” saved data to evolution code
  - used extensively for regular simulations

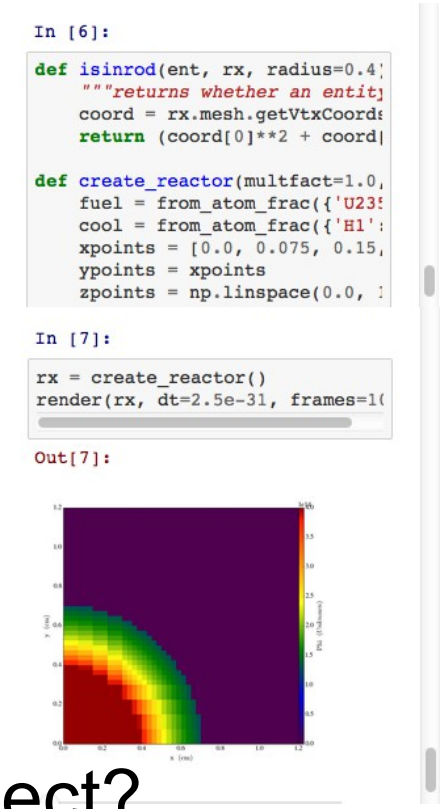


Turk et al., ApJL 2010

```
DataBoxItems=
Add3Plus1ItemsFromGhPsiKappa(
 psi=psi;kappa=kappa;OutputPrefix=;
),
AdmEnergyIntegrandEvo(
 Output=EadmIntegrand;
 InvMetric=Invg; kappa=kappa;
 Coordinates=Inertial::MappedCoords;
);
Observers =
IntegralOverStrahlkorperH5Dat(
 StrahlkorperDataBoxBaseName = ADMSurfIntegral;
 Input = EadmIntegrand;
 OutputNames = EtotADM;
 ApplyFormulae = A;
 FileName  = totADM2090.h5;
 VolumeMetric = g;
);
```

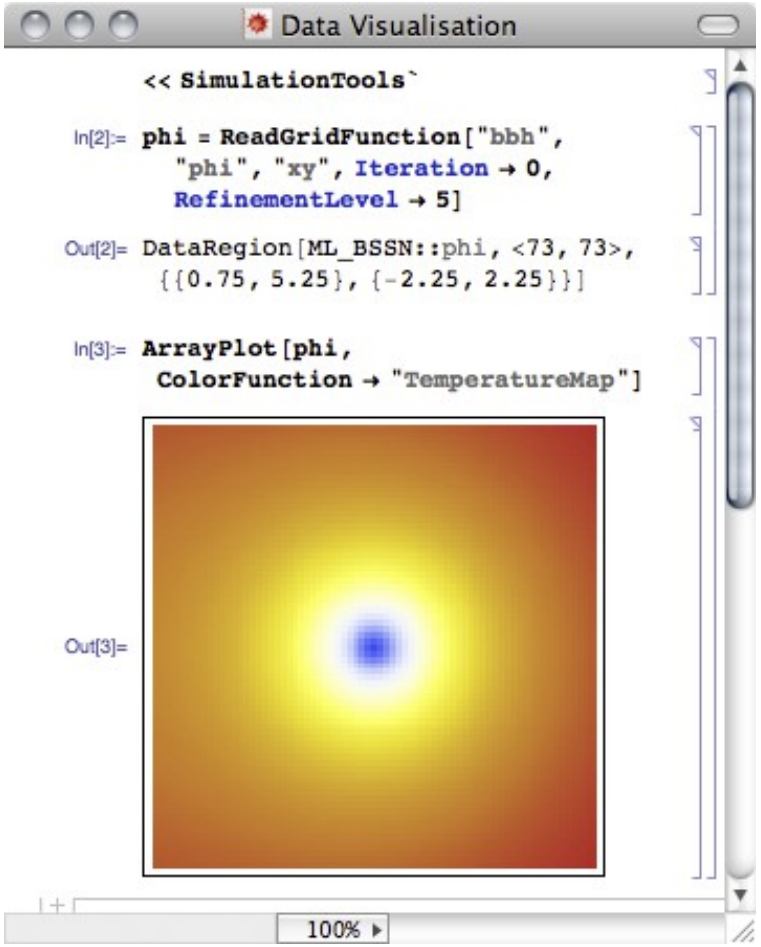
Future postprocessing needs

- GW analysis seems in good shape
  - support for precessing datasets (GWFrames, Boyle)
- multi-d postprocessing is mixed bag right now
  - yt-style data manipulation
  - interactive and scripted use
  - reductions and integrals
  - slices and subsets, isosurfaces
  - MPI parallelized for large datasets
  - full support for mesh refined datasets
- issues, choices
  - pure scripting code? reuse C code?
  - free as in beer?
  - homegrown solution? join existing project?
  - reading many hdf5 files is slow (indexing helps a bit)
  - Speed. May need to process TBs of data



Contenders

- SimulationTools
  - currently in use, mature
  - simple, powerful user interface
  - not designed for 3d datasets
  - Mathematica based,
    no multi-node support
- PostCactus
  - several users
  - simple user interface
  - not designed for 3d datasets
- yt
  - many users, mature
  - simple user interface
  - can do almost all we want
  - cannot read our data



Hinder, SimulationTools

Replay thorn

- generalized file reader to read 3d data into grid during evolution
  - easy to integrate into evolution code once happy
- uses Carpet
  - everything looks the same as during evolution
  - full access to interpolators/reduction ops
  - MPI parallelization
- uses Cactus file reader
  - slow
  - parses all files twice
  - no way to set `cctk_iteration`
- compiled code
  - fast
  - hard to experiment
- not loved by people who used it.

References

- SimulationTools: http://www.simulationtools.org
- PostCactus: see Wolfgang Kastaun's talk
- yt: see Jonah Miller's talk
- PyGWAnalysis: https://svn.einsteintoolkit.org/pyGWAnalysis
- ReadInterpolate: https://github.com/rhaas80/ReadInterpolate
-
-