New frameworks for equation of state and postprocessing

Wolfgang Kastaun[1]

[1]Trento University

ET Workshop, Stockholm, Aug. 2015

PostCactus framework

- Access various Cactus data formats from Python
- Interactive or from scripts
- Can be used locally or remotely via IPython notebook server
- History: grown from pile of postprocessing scripts
- Download from public repository
  `hg clone ssh://hg@bitbucket.org/DrWhat/pycactuset`
- Rudimentary documentation (requires Sphinx)
  `cd PostCactus/doc`
  `make html`

PostCactus framework

Features

- Can read 1D,2D,3D hdf5 data (chunked, unchunked, multiple files)
- Resampling to uniform grids, dimensional cuts
- 1D ASCII data
- Scalars: min, max, norms, integrals
- Multipole data
- GW signal from Ψ_4 multipoles or WaveExtract
- Combined BH information from AHFinderDirect and QuasiLocalMeasures/IsolatedHorizons
- Reads parfiles, transforms into Python objects
- Transparent merging of data from different restarts

PostCactus framework

Limitations

- Does not support one file per group hdf5 format
- Needs helper thorn to store grid volume to convert average to integral
- Parsing parfiles is a mess, and default parameters are inaccessible
- Need machine readable file with all parameters
- No interface for restarts for lack of well defined metadata
- No MPI support for postprocessing

PostCactus framework

Dependencies

- Python 2.7 (might change to Python 3)
- H5Py
- PyTables (soon replaced by H5Py)
- NumPy, SciPy

Can be used with

- Ipython, Ipython notebook server
- Matplotlib
- Yt, MayaVi

SimRep framework

- Automatic generation of html reports from simulation data
- Modular design, easy to design own report pages
- Python based document description language
- Can run arbitrary postprocessing scripts to get plots
- Available modules
  - Logfiles
  - Global quantities (total baryon mass, max density, lapse, ..)
  - Constraint violation
  - Performance (rudimentary, only memory and speed)
  - GW signal

EOS framework

- Interfaces for barotropic $P(\rho)$ and thermal $P(\rho, \epsilon, Y_e)$ EOSs
- EOSs have validity range (important for con2prim)
- Consistent range checking
- Completely independent from Cactus
  (important for unit testing)
- Implemented barotropic EOSs: polytropic, piecewise polytropic, tabulated
- Implemented thermal EOSs: ideal gas, hybrid EOS
- Not public yet: tabulated thermal EOS

EOS framework

- C++ EOS objects
- Clean separation between interface and implementation
- Implementation of particular EOS only needs to provide some virtual methods that define the EOS
- EOS objects can be copied without worrying about memory or ownership
- Uses reference counted pointer to implementation internally
- Cactus thorns to register global evolution and initial data EOSs
- Fortran wrappers possible