CSC 7700: Scientific Computing

Module D: Simulations and Application Frameworks

Homework 2: Write a Cactus thorn

Dr Peter Diener

Center for Computation and Technology
Louisiana State University, Baton Rouge, LA

November 22, 2013

Homework

Homework

Your homework will be to design and implement a (fairly) simple Cactus thorn.

- Physics problem: Find out "where" the star from last weeks homework assignment is oscillating, i.e. what radius contributes most to the kinetic energy of the oscillations.
- Method: Calculate the kinetic energy density everywhere in the neutron star and figure out where it is largest.
- The kinetic energy density is:

$$e_{\mathrm{kin}} = \frac{1}{2}\rho v^2 = \frac{1}{2}\rho\left(v_x^2 + v_y^2 + v_z^2\right),$$

i.e. it is large where the density ρ is high and the velocity v is high.

Homework

- To get started use the command:
  `make newthorn`
  in the Cactus directory. This will ask you for:

  Thorn name: Pick one you like.
  arrangement: Choose an existing one or create your own with a name you like.
  Thorn Author Name: Type your name.
  Email Address: Type your email address.
  Add another author? Y/N: Type `n` or `N`.
  License: Type in your favorite source code license (e.g. GPL).

- After this a new template for a thorn will have been created with `README`, `configuration.ccl`, `interface.ccl`, `param.ccl` and `schedule.ccl` files and `doc`, `par`, `src` and `test` directories to be filled in.

Homework

- The variables `rho` (grid function) and `vel` (3-vector gridfunction) are defined in `EinsteinBase/HydroBase`.
- The kinetic energy density is defined at every gridpoint and needs to be stored in a grid function.
- Write a routine (placed in the `src` directory) to iterate over all grid points...
- ...and then evaluate the formula from the previous slide for each grid point and store the result in the grid function.
- It's up to you if you want to use C, C++ or F90. The grid functions will be accessed differently depending on your choice.
  - F90: look at `EinsteinInitialData/GRHydro_InitData/src/GRHydro_ShockTube.F90`
  - C or C++: look at `EinsteinInitialData/TOVSolver/src/tov.c`.

Homework

- `interface.ccl`:
  - Define the implementation name that the thorn implements.
  - Inherit from `HydroBase` where `rho` and `vel` are defined.
  - Declare a new grid function for kinetic energy density.
- `schedule.ccl`:
  - Schedule your routine that performs the calculation in the `analysis` schedule bin.
  - Request storage for the new grid function for this calculation.
- `param.ccl`:
  - This thorn doesn't need any parameters.
- `configuration.ccl`:
  - This thorn doesn't use any special capabilities.

Homework

- Add your source filename to `make.code.defn` so that the make system will compile your code. Look at other thorns to see how.
- Add your new thorn to the thornlist you used to compile the Cactus configuration from last week's homework and recompile. Go back and fix problems if the compile fails and try again.
- Copy the parameter file from last week's homework to a different name and edit it:
  - Activate your thorn.
  - Request 1D output of your new grid function.
- Run as you did last week but specify the new parameter file.
- Visualise the 1D output. Note that this run uses AMR so the 1D output contains data from multiple refinement levels. We are here only interested in the output from the finest refinement level (which is refinement level 4 in the output file).

Homework

- Your homework will consist of the a tar ball of your thorn and a plot of the kinetic energy density as a function of x. Use a gnuplot command like:
  `p '<file_name>' u 10:(($3==4 && $9==1000)?$13:1/0) w lp`
  to extract the relevant data.
- Brownie points to whomever can explain how the gnuplot command above works.
- Homework is due on December 6, 2013.