TOV-Solver – Manual

David Kling

June 17, 2016

Define parameters

Go to `src` and open the file `params.f90`. In this file we have to define the two gauging constants Γ and K. There are predefined values for those constants for some characteristic cases:

- For a degenerated cold non-relativistic electron gas choose: `K=K_e_non_rel`, `gamma_poly=gamma_poly_e_non_rel`.
- For a degenerated cold relativistic electron gas choose: `K=K_e_rel`, `gamma_poly=gamma_poly_e_rel`.
- For a degenerated cold non-relativistic neutron gas choose: `K=K_n_non_rel`, `gamma_poly=gamma_poly_n_non_rel`.

Additionally choose how many plots you want to make by changing the constant `points_M_R` and by defining the pressure range – variables `P_init` and `P_fin`. I have added some good ranges depending on the gauging constants you have chosen for the equation of state (see comments in that file). The TOV-solver will go through this range logarithmically (for details see source code of the module `integrator.f90`).

Compile & run the code

- Create a directory for the output called `output_TOV_solver` (do not use any other name, else the program will not understand where to save the data). Create the directory in the same one in which `src` is located.
- Compile the program using `gfortran` by:
  `gfortran -o TOV_solver.e src/params.f90 src/ODE_system.f90 src/params.f90 src/rk4.f90 src/integrator.f90 src/TOV_main.f90`
  (do not use any other order of the module names); this creates the executable.
- Execute the executable by: `./TOV_solver.e`.

Plot the data

- To plot the results you need `python3` and `matplotlib` for `python3`. Make sure the script `plot_TOV.py` is located in the same directory as the directory `output_TOV_solver`.
- Use `python3 plot_TOV.py` to run the script. It automatically creates a directory called `plots` and some additional sub-directories in which you find the plots (for details check the code of `plot_TOV.py` and the comments in it).