Operation guide for plotting of RANS X transport, flux and variance equations based on ransdat data from PROMPI

Date	Version	Person	Change
25/March/2018	1.0	Miroslav Mocak	Initial instructions for plotting of ransX equations (code still needs to be tested with proper turbulence data)
20/April/2018	1.1	Miroslav Mocak	Introducing parameter ig for switching between spherical and Cartesian geometry

Prerequisite:

- Linux/Windows operating system
- Python 2.7 + ipython

1. Go to https://github.com/mmicromegas/PROMPI_DATA

2. To a dedicated directory <DIR> download the following files (or download the whole repo but some classes and still not finished):
   - PROMPI_RANS_xnu.py (class for plotting of the ransX equations)
   - PROMPI_data.py (class for reading ransdat data)
   - CALCULUS.py (class with useful calculus methods)
   - plot_rans_xnu.py (control script for plotting)
   - rans_tseries.py (script for calculating of time-averaged fields from ransdat)

3. In <DIR> create two sub-directories DATA and RESULTS

4. Copy your *.ransdat and *.ranshead data to folder DATA

5. Open rans_tseries.py and adjust the following parameters:
   - trange (line 9). Restrict time-range of your ransdat data in DATA folder for time-averaging.
   - tavg (line 10). Set time-averaging window (at least 2 convection turnover timescales)

6. From <DIR>, start ipython and execute > run rans_tseries.py

7. After successful completion of rans_tseries.py, the time-averaged data are stored in a file called tseries_ransout.npy. Check if the file was in the <DIR> created successfully.

8. Open plot_rans_xnu.py and adjust the following parameters according to your needs:

- intc . Choose index of central time for which you wish to plot the ransX fields.
- inuc . Choose ID of the element for which you want to plot the ransX fields. It has to have the format 00xx , for example 0001 is neutrons, 0002 is protons, 0003 is he4, 0004 could be c12 (all depends on your network)
- ig . Enter geometry of your simulation (1 is Cartesian, 2 is spherical)
- LGRID . Choose whether you want to limit your x-grid. Good if you want to get rid of boundary noise (1-true, 0-false)
- xbl, xbr . Set left/right radius for which you want to limit x-grid in your plots. Y-axis will adjust itself automatically.
- lc . Optional. Estimated size of convection zone. This is still work in progress. Set it properly, if you want to get Eulerian diffusivities right.

9. From <DIR>, start ipython and execute > run plot_rans_xnu.py

10. Wait for the plots to be displayed.

11. If you wish to display also radial profiles of element density, flux and variance, uncomment the following lines #RANSX.plot_Xrho(xbl,xbr,inuc,data_prefix), #RANSX.plot_Xflux(xbl,xbr,inuc,data_prefix), #RANSX.plot_Xvariance(xbl,xbr,inuc,data_prefix)

12. If you want to display various diffusivities for the target element, uncomment #RANSX.plot_X_Ediffusivity(xbl,xbr,inuc,data_prefix)