Diagnostic output for the ETS solver

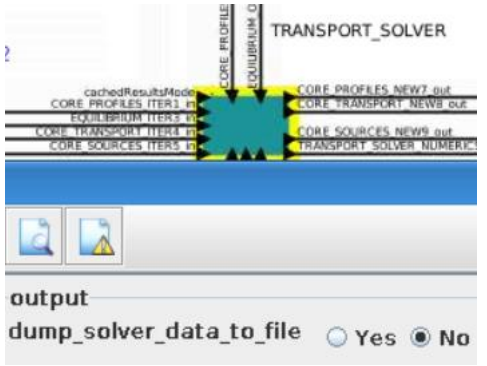
Dmitriy Yadykin

Purpose and main features

- Purpose
  - to check input to the solver for particular equation (psi, n,T)
  - to compare/benchmark different ETS version (5 vs 6) or reference vs actual version (some kind of regression test)
- Main features:
  - number of profiles are written to the file (placed in $KEPLER folder)
  - one instance of the profiles is stored (iteration or time step); the profiles from last time step/iteration are saved
  - dedicated python script is used for post-processing (plotting)

Implementation

- Equations in ETS are defined in ‘generalized form’ using the ‘same’ coefficients (a,b,…g).
- Coefficients are composed from the input profiles (equilibrium, transport, sources)
- These two types of profiles are gathered and put to the data file for each equation
- First ion species is used for ion equations
- the diagnostic output is implemented for now (20-04-15) for psi, Te, Ti, ne
- diagnostic output is activated by the code parameter of the TRANSPORT_SOLVER actor



Example:psi equation, source code

```
!> Formulation of the differential equation for \f$Y(t)\f$
!> for a given\f$Y(t-dt)\f$ in terms of the coefficients
!> \f$\{A, B, C, D, E, F, G\}\f$ in the SOLVER object.
!>
!> \f[
!>    [A*Y-B*Y(t-dt)]/H + 1/C * [-D*Y(t)' + E*Y(t)]' = F - G*Y
!> \f]
!-------------------------------------------------------

  NUMERICAL_COEFFICIENTS_PSI: DO IRHO=1,NRHO
     SOLVER%Y(1,IRHO)     =  PROFILES%PSI(IRHO)
     SOLVER%DY(1,IRHO)    =  PROFILES%DPSI(IRHO)
     SOLVER%YM(1,IRHO)    =  PROFILES%PSIM(IRHO)
     SOLVER%DYM(1,IRHO)   =  PROFILES%DPSIM(IRHO)

     SOLVER%A(1,IRHO)     =  TRANSPORT%SIGMA(IRHO)
     SOLVER%B(1,IRHO)     =  TRANSPORT%SIGMA(IRHO)
     SOLVER%C(1,IRHO)     =  IMAS_CONSTANTS%MU0*GEOMETRY%BGEO*GEOMETRY%RHO(IRHO)*GEOMETRY%RHO_BND/GEOMETRY%FDIA(IRHO)**2
     SOLVER%D(1,IRHO)     =  GEOMETRY%VPR(IRHO)/4.e0_IDS_REAL/IMAS_CONSTANTS%PI**2*GEOMETRY%G3(IRHO)/GEOMETRY%FDIA(IRHO)/GEOMETRY%RHO_BND
     SOLVER%E(1,IRHO)     = -IMAS_CONSTANTS%MU0*GEOMETRY%BGEO*(GEOMETRY%RHO(IRHO)/GEOMETRY%FDIA(IRHO))**2*TRANSPORT%SIGMA(IRHO)      &
                            * GEOMETRY%PHI_BND_PRIME/GEOMETRY%PHI_BND/2.e0_IDS_REAL
     IF (GEOMETRY%RHO(IRHO).NE.0.e0_IDS_REAL) THEN
        SOLVER%F(1,IRHO)  =  - GEOMETRY%VPR(IRHO)/2.e0_IDS_REAL/IMAS_CONSTANTS%PI/GEOMETRY%RHO(IRHO)*SOURCES%CURR_EXP(IRHO)
        SOLVER%G(1,IRHO)  =  GEOMETRY%VPR(IRHO)/2.e0_IDS_REAL/IMAS_CONSTANTS%PI/GEOMETRY%RHO(IRHO)*SOURCES%CURR_IMP(IRHO)
                            + TRANSPORT%SIGMA(IRHO) * (2.0_IDS_REAL + FUN1(IRHO))                                              &
                            * GEOMETRY%PHI_BND_PRIME/GEOMETRY%PHI_BND/2.e0_IDS_REAL
     ELSE
        SOLVER%F(1,IRHO)  =  0.0_IDS_REAL
        SOLVER%G(1,IRHO)  =  TRANSPORT%SIGMA(IRHO) * GEOMETRY%PHI_BND_PRIME/GEOMETRY%PHI_BND

     ENDIF
  END DO NUMERICAL_COEFFICIENTS_PSI
```

```
IF (dump_out_flag.EQ.1) THEN
  FILENAME_DIAGOUT='diagout_sover_psi_ets6'
  CALL DIAGNOSTIC_SOLVER (FILENAME_DIAGOUT,NRHO,GEOMETRY%RHO,SOLVER%Y(1,:),SOLVER%YM(1,:),&
       SOLVER%DY(1,:),SOLVER%A(1,:),SOLVER%B(1,:),SOLVER%C(1,:),SOLVER%D(1,:),SOLVER%E(1,:), &
       SOLVER%F(1,:),SOLVER%G(1,:),SOLVER%H)
  PRIME_TERM=GEOMETRY%BGEO*GEOMETRY%PHI_BND_PRIME/GEOMETRY%PHI_BND/2.e0_IDS_REAL
  FILENAME_diagout='diagout_profiles_psi_ets6'
  CALL DIAGNOSTIC_PROFILES_PSI(FILENAME_DIAGOUT,NRHO,GEOMETRY%RHO,TRANSPORT%SIGMA,GEOMETRY%FDIA,&
       GEOMETRY%VPR,GEOMETRY%G3,SOURCES%CURR_EXP,SOURCES%CURR_IMP,PRIME_TERM)
END IF
```

Post-processing

Python script:
https://gforge6.eufus.eu/svn/keplerworkflows/trunk/imas/ETS/kplots/diagout.py

Available arguments:

```
[~/wfimas/kplots] python diagout.py -h
usage: diagout.py [-h] [--folder [FOLDER [FOLDER ...]]]
                  [--filenames [FILENAMES [FILENAMES ...]]]
                  [--labels [LABELS [LABELS ...]]] [--mode [MODE [MODE ...]]]

optional arguments:
  -h, --help            show this help message and exit
  --folder [FOLDER [FOLDER ...]]
                        OPTIONAL:folder name where diagnostic output files are
                        stored, single place for now
  --filenames [FILENAMES [FILENAMES ...]]
                        OPTIONAL:file name(s) of the diagnostic files to be
                        plotted
  --labels [LABELS [LABELS ...]]
                        OPTIONAL:names to label plotted cases
  --mode [MODE [MODE ...]]
                        OPTIONAL:execution mode, can be solver_<equation_name>
                        or profiles_<equation_name>, default is solver_psi
```

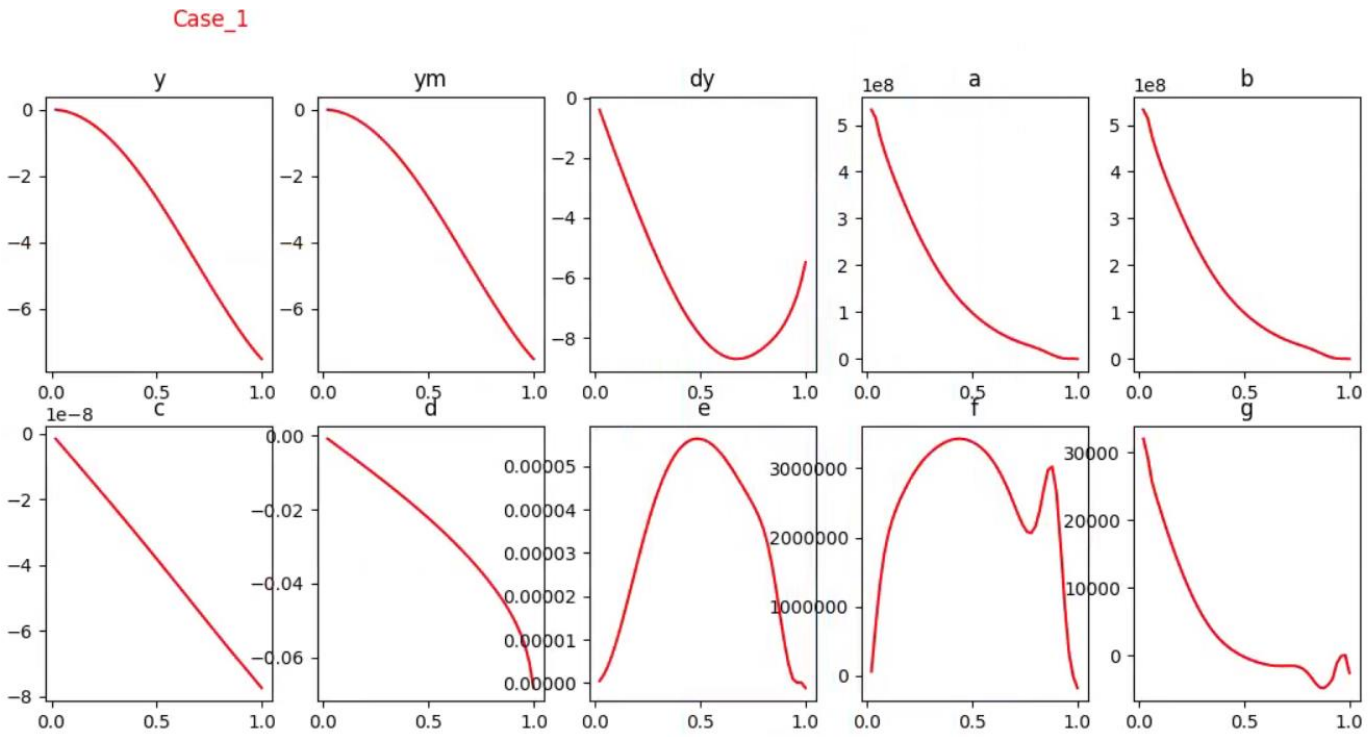
Default values:
-folder - **$KEPLER** (local kepler should be loaded)
- filenames - **diagout_solver_psi_ets6**
- labels - **Case_1**
- mode - **solver_psi**

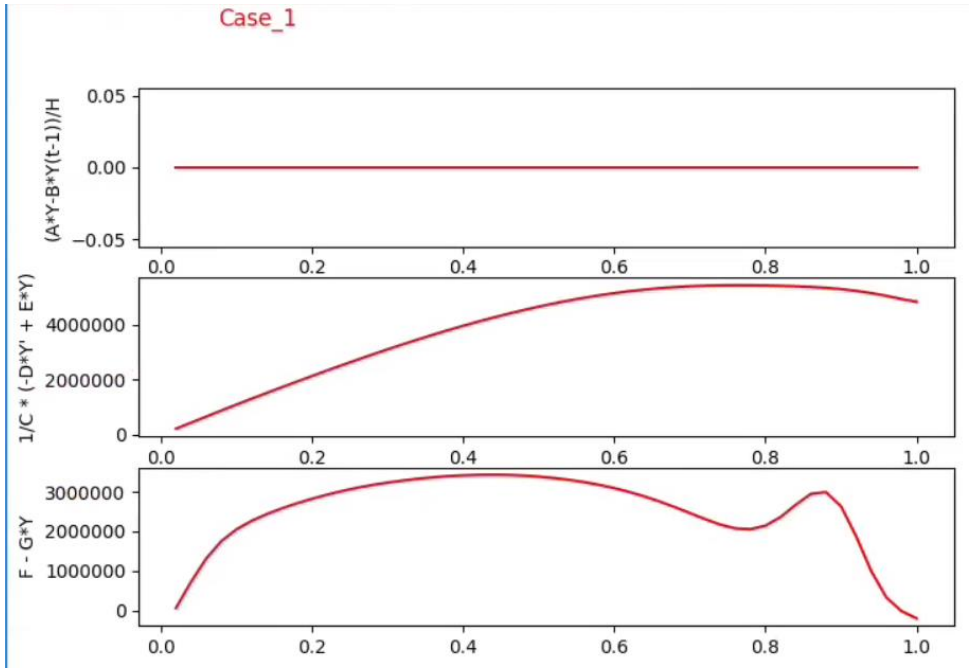
Example: default

To prepare:

- load paramfile (starting from imas_kepler folder)
  param/jet/param_benchmark_ets5_psi_evolution_ip_nclass.txt
- activate dump_solver_data_to_file parameter
- run the workflow

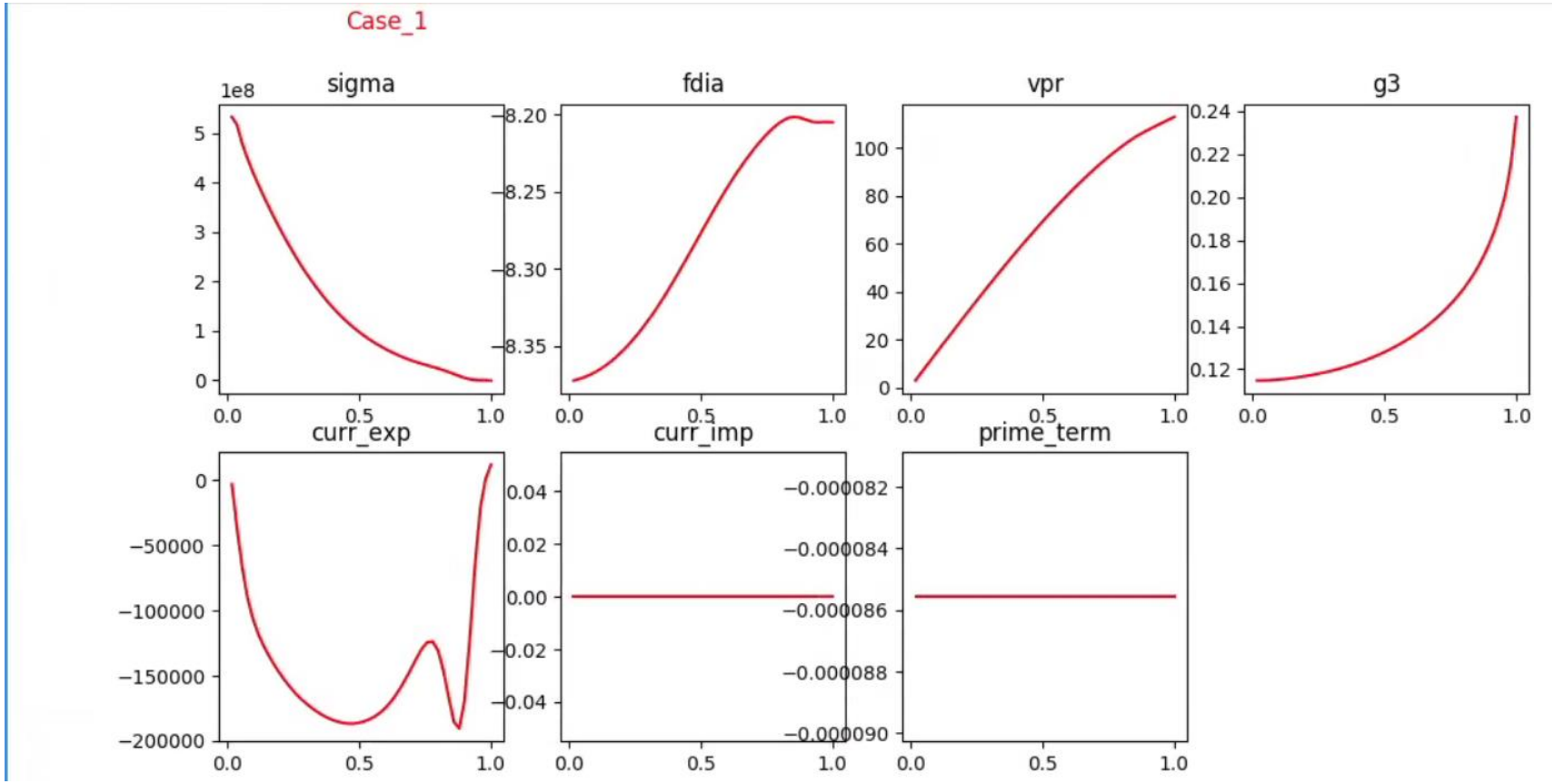
To execute: ./diagout.py in the kplots folder





Example: mode=profiles_psi

To execute:
./diagout.py --mode profiles_psi



Example: compare ETS5/6 profiles_psi

Prepare:

- run correspondent ETS5 case (ask me how if you interested)
- copy diagout files for both versions to the one folder (~/public/data/diagout_ets in my case)

Execute:

./diagout.py --folder ~/public/data/diagout_ets --filenames diagout_profiles_psi_ets5 diagout_profiles_psi_ets6 --labels ets5 ets6 --mode profiles_psi