EFDA

EUROPEAN FUSION DEVELOPMENT AGREEMENT

Task Force
INTEGRATED TOKAMAK MODELLING

23/04/2009

Exercises

F. Imbeaux for the ISIP team

TF Leader : P. Strand,
Deputies: L-G. Eriksson, R. Coelho, M. Romanelli

EFDA CSU Contact Person: D. Kalupin

https://portal.efda-itm.eu/portal/authsec/portal/itm/ISIP
isip@mail.efda-itm.eu

Data entries

- Create your private database : 4.06d, test
- Copy the input example from the public Database (shot = 3, run = 1) to your private database
- Set the UAL environment variables to work on your private database

- Documentation on https://portal.efda-itm.eu/portal/authsec/portal/itm/ISIP/content?action=2&uri=/isip/database/Database_UG.htm

Data entries : solutions

- Create your private database : 4.06d, test
  - /afs/efda-itm.eu/project/switm/scripts/create_user_itm_dir test 4.06d
  - Have a look to it using : ls ~my_username/public/itmdb/itm_trees
- Copy the input example from the public Database (shot = 3, run = 1) to your private database
  - cp /pfs/itmdb/itm_trees/public/test/4.06d/mdsplus/0/euitm_30001.* ~my_username/public/itmdb/itm_trees/test/4.06d/mdsplus/0/.
- Set the UAL environment variables to work on your private database
  - source /afs/efda-itm.eu/project/switm/scripts/set_itm_data_env my_username test 4.06d

Write a physics module for a single time slice

- Write a Fortran physics module that does the following :
- Input : single time slice of a coreprof CPO
- Output : single time slice of an mhd CPO
  - Copy the Time and Psi values from the coreprof CPO to the mhd CPO
  - Mimic the result of an MHD calculation by filling the MHD frequency signal with some values
  - Fill the codeparam substructure : be careful to the specific treatment of strings in Fortran (ITM) : they must be allocated by line of 132 characters

- Start from ~imbeaux/public/training_ISIP/my_physics_program_slice_ex.f90

Test if it works

- Rename your program to my_physics_program_slice.f90
- Copy all files from ~imbeaux/public/training_ISIP
  - Makefile : in the « all » line, remove all mentions to my_physics_program and my_wrapper
  - The folder contains now a standard wrapper that should work with your routine
- Make clean
- Make
- My_wrapper_slice : will execute your program, taking coreprof from shot 3, run 1 and send the output « mhd » to shot 3, run 2
- Test_mhd_get : will check shot 3, run 2 to see if the values are correct

Write a physics module for multiple time slices

- Write a Fortran physics module that does the following :
- Input : multiple time slices of a coreprof CPO
- Output : multiple time slices of an mhd CPO
  - Same functionality as before, but extend to all time slices

- Start from ~imbeaux/public/training_ISIP/my_physics_program_ex.f90

Solutions

- All solutions in ~imbeaux/public/training_backup
- Can be used as examples for developping your own applications (physics subroutines, manual wrappers)

UAL practice

- In normal use, physics users or developers should not manipulate the UAL directly
  - The wrapper around physics modules is doing the UAL calls
- However, it may be useful, for testing purposes, to have some knowledge and practice of the UAL

UAL main functions

- Main UAL functions (see UAL documentation on ISIP portal)
  - Open (shot,run) or create (shot,run) → returns Idx, an identifier of the data entry for the UAL
  - Close (shot,run)
  - GET/PUT : get an entire CPO (with all time slices)
  - GET_SLICE : get a single time slice of a CPO (three interpolation methods are allowed)
  - PUT_SLICE : appends a single time slice to an existing CPO. If starting from an empty CPO, must be initialised by a PUT_NON_TIMED (will write time-independent information only)