Tutorial on using EQSTABIL workflow

R. Coelho (14/12/2015)

CONTENTS

1. Pre-requisites

- Check out the EQSTABIL release from :
  https://gforge.efda-itm.eu/svn/eqstabil/tags/4.10a3_R2.2.0
- Follow the instructions to ensure the correct Kepler installation is used and to install all actors needed :
  http://portal.efda-itm.eu/twiki/bin/view/Main/EQSTABIL_install_410a3
- Create, if necessary, a user database entry for device=test and dataversion=4.10a. Copy all test cases used for the tutorial there :

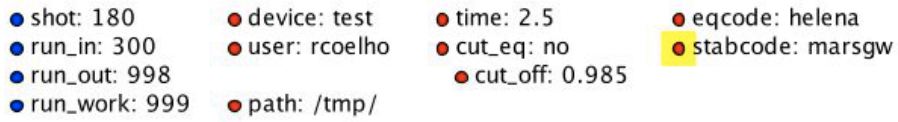
  $>cp /tutorial/case*/euitm* $MDSPLUS_TREE_BASE_0

- Set the working environment for device=test, dedicated Kepler installation and UAL **4.10a.3_R2.2.0** :

  $>source $ITMSCRIPTDIR/ITMv1 kepler test 4.10a.3_R2.2.0

2. Setting workflow variables and actor parameters

As indicated in the TWiki, the EQSTABIL workflow has the following set of parameters :

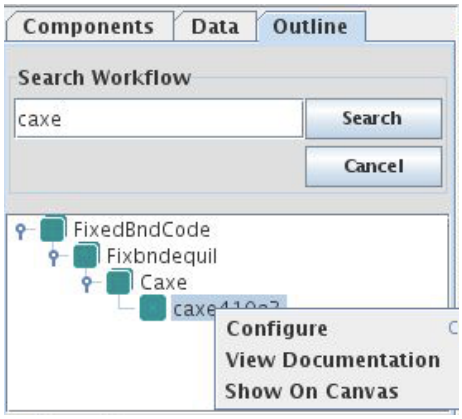


- **shot** : the shot number on the user database (or from another user) where to read the reference equilibrium from (shot/run_in pair)
- **run_in** : the run number where the reference equilibrium is (shot/run_in pair)
- **run_work** : placeholder run for the temporary Kepler CPOs
- **run_out** : run number where the final results of the run will be stored (user running the workflow/shot/run_out). Since the input equilibrium can be a reconstruction that goes beyond the separatrix, 3 occurrences of the equilibrium are saved (original eq., cut equilibrium inside separatrix and corresponding high resolution equilibrium).
- **user** : username. Reading from someone else database is possible but the run_out will naturally be written to personal database only.
- **device** : device database where the input reference data is. MUST BE the same as env variable TOKAMAKNAME
- **time** : time slice (in equilibrium CPO) to be analysed in case the input shot/run_in contains many time slices.
- **path** : temporary folder where to dump the plots generated. Also used to store output files (used by HELENA/ILSA only)
- **cut_eq** :
  - yes : cut the input equilibrium (necessary if high resolution equilibrium code cannot handle separatrix plasma equilibria)
  - no : input equilibrium is used "as is".

- **cut_off** : float]0,1], specifies the percentage of the separatrix flux that will define the poloidal flux of the new plasma boundary.
- **eqcode** : chease/caxe/helena. The equilibrium code to be used
- **stabcode** : ilsa/kinx/marsgw/marsf. The MHD stability code to be used

Actor parameters are set on the actors themselves (not passed by the workflow). To access the actors codeparam the easiest route is to :

1. Click on "Outline" Tab (below the "Pause" button)
2. Type the name of the actor and press "Search" (or Enter)
3. On the final item in the chain of the actor composite, right click and press "Configure". A pop-up panel appears
4. Click on "Edit Code Parameters" and a new window appears
5. Edit the code parameters and Press "Save & Exit"
6. Press "Commit" and setting is completed



3. Case 1 : JET-like equilibrium with Internal n=1 mode

Change directory to the folder where you checked out the software. Make sure to run the ITMv1 script prepared by the CPT (http://portal.efda-itm.eu/twiki/bin/view/Main/UAL_Releases) for the correct UAL and device version. To check just echo from the shell $UAL and $TOKAMAKNAME.

As indicated in the TWiki, this case corresponds to ***device/shot/run*** entry ***test/180/300.***

Step 1: Modify the workflow variables but set a wrong shot or run_in i.e. set shot=1800, run_in=300. Press Play button  . Check the Multi Tab Display window for the messages returned. The workflow automatically stops whenever any of the following three conditions are met :

- No equilibrium CPO on input
- Wrong UAL dataversion set just before starting Kepler
- Mismatch between device set in workspace environment and Workflow.

Step 2: Set shot=180 and run_in=300. Set cut_eq=yes and Run the workflow  . Check the Multi Tab Display window. The workflow stopped because to cut the input equilibrium actually comes in (ψ,χ) flux coordinates. Cutting is available only for equilibrium given in (R,Z) coordinates.

Step 3: Set shot=180 and run_in=300. Set cut_eq=no and Run the workflow. The input equilibrium is shown on a Python plot window. If you confirm this is the equilibrium you wish to work with, close the plot window to proceed to the high resolution equilibrium calculation. If you want to start from another input or wish to verify/reset the code parameters, press the ***Stop*** button  on the workflow ***before*** closing the Plot window.

Step 4: Once high resolution equilibrium is calculated, a new Python window is shown. If satisfied close it to proceed to linear stability calculation. If not then Stop the workflow execution  and then close the Plot window.

Step 5: After concluding the linear stability calculation, two Plot windows are shown : one with the radial coordinate grid and another with the eigenfunction (real and imaginary components) for radial displacement/velocity. Closing both windows, the output CPOs are written to the itmdb and workflow execution finishes. The Multi Tab Display window shows the output_flag for the linear stability code execution. If the run actually converged to an eigenvalue the output_flag is set to 0.

Changing parameters

Step 6: Set the boundary conditions on the MHD codes for a perfect conducting wall at the plasma boundary. Procedure in the codes available :

- ILSA – set **boundary = free boundary**
- MARS – set **nv=4**, **rext=1.001e0**
- MARSF – set **nv=4**, **rext=1.001e0**
- KINX – set **nav=4**, **rext=1.001e0**

Step 7: Change linear stability code to “marsgw” and run 2 different number of harmonics (**m1** – lower limit and **m2** – upper limit) e.g. (m1=-2, m2=12) and (m1=-15, m2=25). N.B. As explained on the actor documentation (TWiki or in the actor itself in Kepler – right-click on it and press Display on the Documentation panel), the definition of the Alfven time used in MARS/MARSF is different and as such the normalized eigenvalue displayed is scaled by **(Rmag/Bmag)*(B0/R0)**. For this particular equilibrium the ratio is 1.19675.

Step 8: Change equilibrium code to “caxe” and linear stability code to “kinx”. Run the workflow and compare the results with those obtained with previous codes. You can repeat with different number of harmonics (changing **nt1** – number of poloidal grid points in “caxe” while using **ipt1=2** in “kinx” to keep the same grid resolution)

4. Case 2 : Circular equilibrium with global n=1 mode

As indicated in the TWiki, this case corresponds to ***device/shot/run*** entry ***test/1/2.***

Step 1: Set the correct shot/run_in variable in the workflow. This case also doesn't qualify to cut the equilibrium so keep cut_eq=no.

Changing parameters

Step 2: Set equilibrium code to "caxe" and stability code to "kinx". Go to "kinx" code parameters and set **al0=-8e-3** (KINX uses negative guesses for squared growth rate of unstable modes) and **NAV=40**, **WAC=3.5**. Run the workflow. Has the run converged in the iterations set in code parameters (NITMAX) ?

Hint 1: Look at the Multi Tab Display and check the output flag.

Hint 2: Did the eigenfunction Plot window show the eigenvalue (had it converged…) ?

Step 3: Change the "kinx" code parameter to **al0=-3e-3** and re-run the workflow. What's the converged eigenvalue ?

Step 4: Change the workflow setting to use HELENA (eqcode: helena) and ILSA (stabcode: ilsa). Since the mode is unstable (eigenvalue squared is positive) only when the perfect conducting wall is beyond ~x1.6 the minor radius, set in "ilsa" code parameters : **boundary = free boundary** and **rwall=3.50**. The initial guess need not be close to the eigenvalue since ILSA automatically iterates the initial guess when failing to converge. Compare the results with CAXE/KINX.

Step 5: Set **eqcode: chease** and **stabcode: marsf** and vary the wall position (**rext**) from 3.50 up to 1.6 to see growthrate decrease. The user needs to specify the number of vacuum grid points in "marsf" code parameters e.g. **nv=40**.

N.B: close to marginal stability mesh densifications are recommended. Setting position using CHEASE is controlled by choosing the safety factor q(s) where to perform densifications (arrays **qplace**, **qwidth** and scalar **npoidq**)

5. Case 3 : AUG equilibrium (no separatrix) with internal/global n=1 mode

As indicated in the TWiki, this case corresponds to ***device/shot/run*** entry ***test/29100/5.***

Step 1: Set the correct shot/run_in variable in the workflow. This case also doesn't qualify to cut the equilibrium so keep cut_eq=no.

Changing parameters

Step 2: Change equilibrium code to "helena" and stability code to "marsgw". Go to "helena" code parameters and set equidistant=0.0. Go to "marsgw" code parameters and set **rext=1.2.** Run the workflow.

Step 3: To densify the equilibrium grid in HELENA, first take note of the s-grid location of the rational q-surfaces and edge safety factor. These correspond to s=0.7482, 0.861, 0.9367, 1.0. The densification can be more "severe" as we move closer to the edge and weaker on axis. To impose a densification that yields increasingly denser grid as one moves to the plasma edge (*but excluding the rational surfaces)*, one would set :
s_acc=0 0.7 1.0
sig=1.0 0.3 0.1
weights=0.5 0.3 1.0

On top of this set, one can now impose the densification at the rational surfaces, yielding :
imesh=2
n_acc_points=6
s_acc=0 0.7 0.7482 0.861 0.9367 1.0
sig=1.0 0.3 0.01 0.01 0.01 0.008
weights=0.5 0.3 0.02 0.02 0.02 0.2

Step 4: Use CHEASE instead and densify the grid at the rational surfaces (2,3,4 and 4.7).

6. Case 3 : AUG equilibrium (with separatrix) with internal/global n=1 mode

As indicated in the TWiki, this case corresponds to ***device/shot/run*** entry ***test/29100/4.***

Step 1: Set the correct shot/run_in variable in the workflow. This case also qualifies to cut the equilibrium but for now keep ***cut_eq=no***.

Changing parameters

Step 2: Set equilibrium code to “chease” and stability code to “ilsa”. Go to “ilsa” code parameters and set **rwall=1.5** and **upper_bound=2e-4**. Run the workflow and wait for the input equilibrium visualization.

It becomes evident that the plasma has a separatrix and it’s pointless to continue the run so STOP the workflow by pressing the STOP button ***before*** closing the plot windows.

Step 3: Set the workflow parameter **cut_eq=yes** and **cut_off=0.958**. Run the workflow and note the cut boundary (in green) before entering the high resolution equilibrium calculation.

Step 4: Repeat with **cut_off=0.985**. Change to “caxe” and “kinx” or other code pairs as wished.