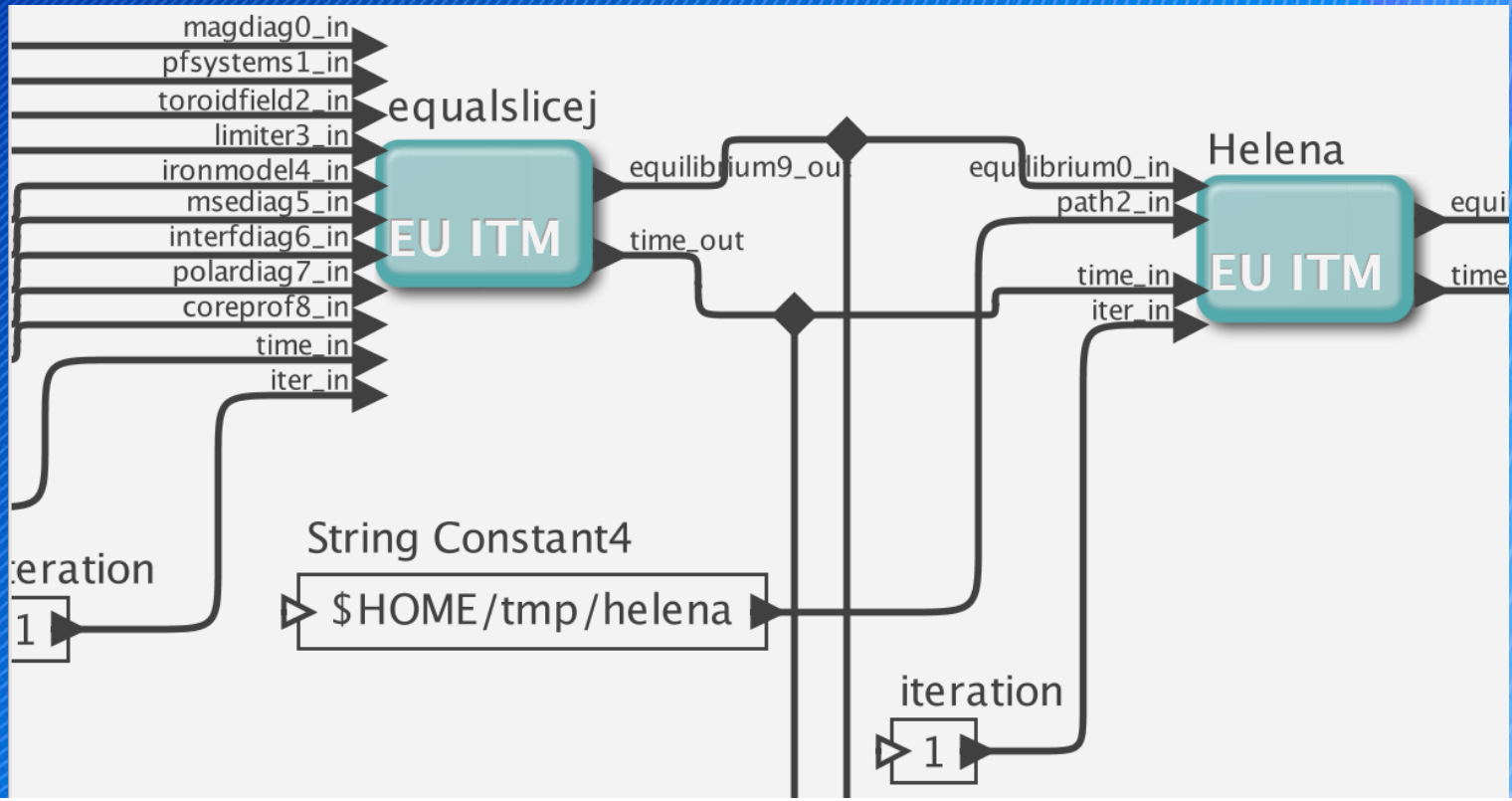
EFDA
EUROPEAN FUSION DEVELOPMENT AGREEMENT
Task Force
INTEGRATED TOKAMAK MODELLING

ITM-TF Code Camp Training, Garching,
12-16 March 2012

Training
Equilibrium and Stability Workflow

W. Zwingmann

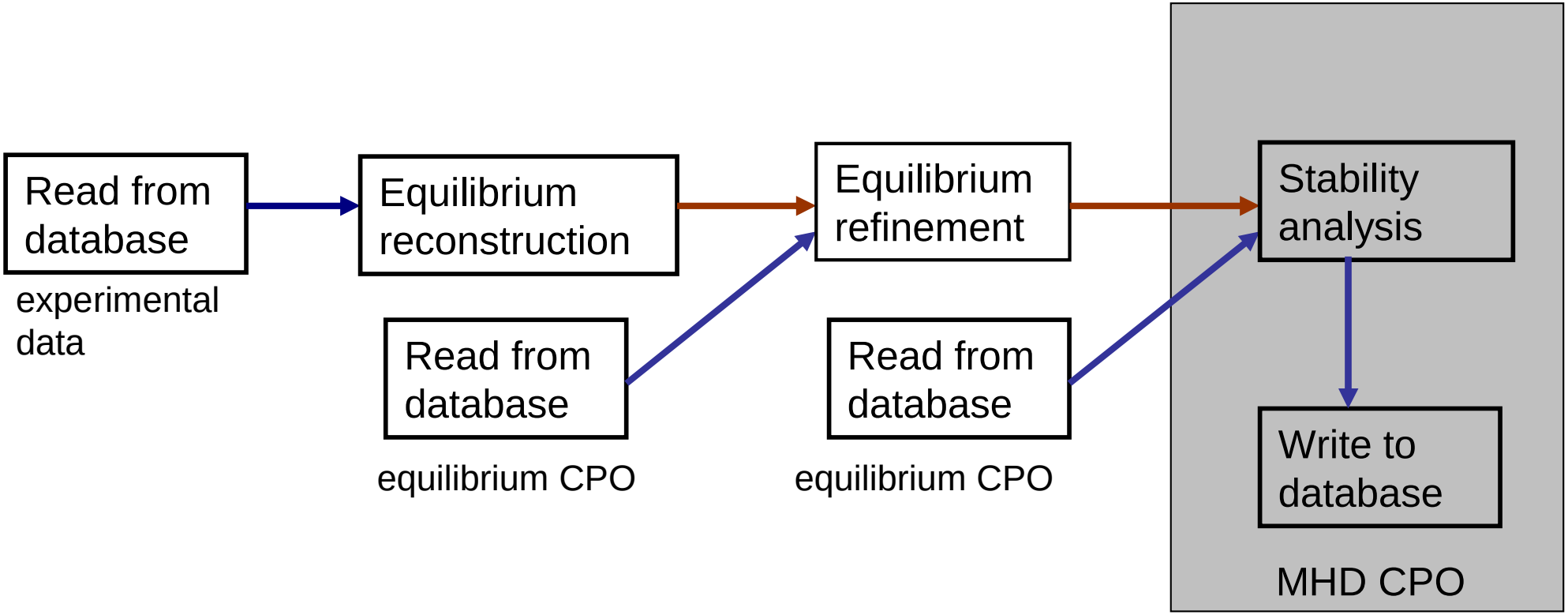


Outline

- Equilibrium and stability workflow
- Equilibrium reconstruction
- Equilibrium refinement
- Use of workflow
- Code parameters
- Use of UALdemux
- Use of kepler plotting routines
- Batch execution (general)
- Python visualisation

Material on: Gateway ~zwingman/public/GARCHING2012

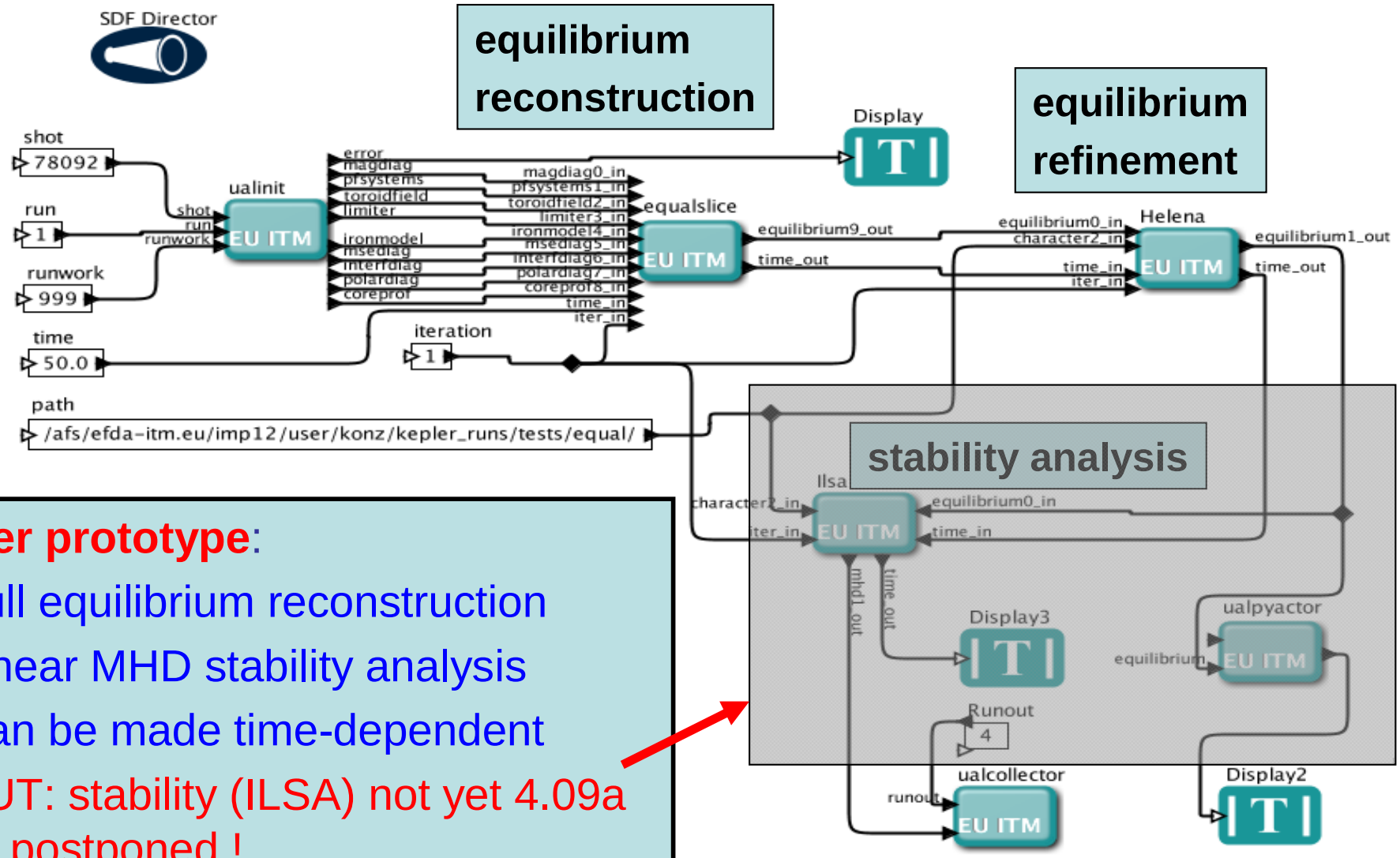
Equilibrium and Stability Chain



e.g.	But also:
❑ JET data	❑ AUG, Tore Supra, FTU, …
❑ EQUAL	❑ CLISTE, FIXFREE, EQUINOX, …
❑ HELENA	❑ CHEASE, CAXE, CLISTE …
❑ ILSA	❑ KINX, MARS-F

INTEROPERABILITY

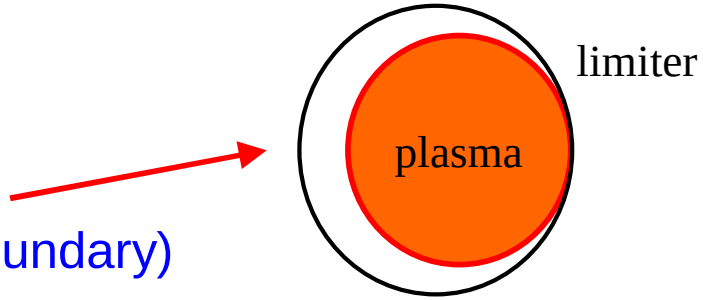
Equilibrium and Stability 4.08b



Kepler prototype:

- Full equilibrium reconstruction
- Linear MHD stability analysis
- Can be made time-dependent
- BUT: stability (ILSA) not yet 4.09a
  ⇒ postponed !

Equilibrium solvers



Assumptions:

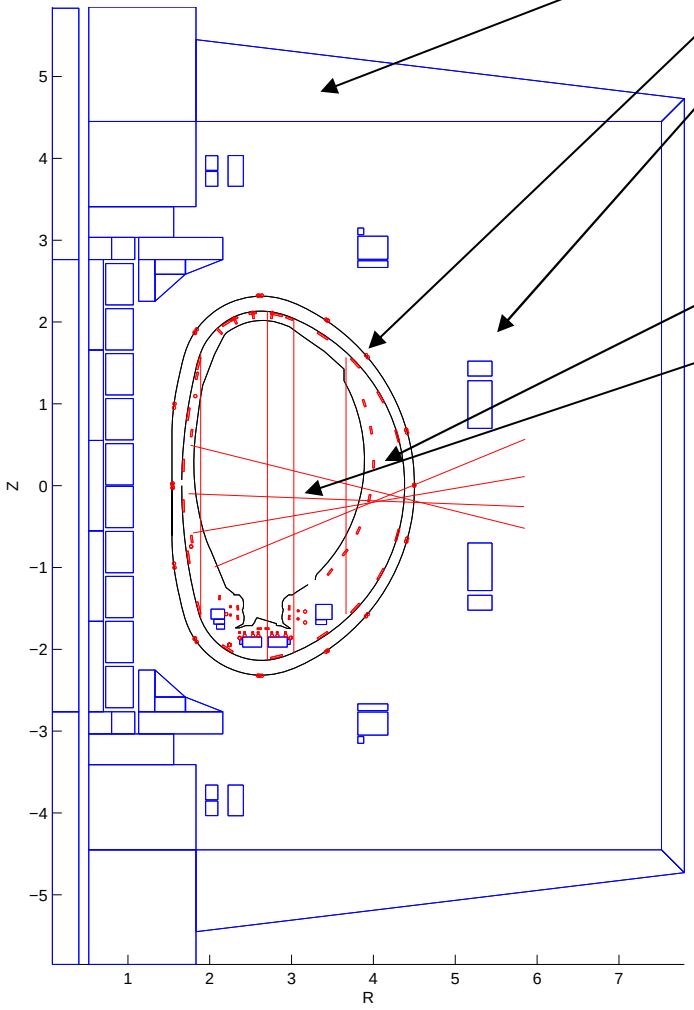
- d/dt = 0
- plasma boundary determined by code (free boundary)

Code name	Op	**v**≠0	3D	An-isotropy	GForge	Tokamak independent
CEDRES++	D				yes	yes
FIXFREE	D					
CREATE-NL	D				yes	
EQUAL	R			(yes)	yes	yes
EFIT++	R					
EQUINOX++	R				yes	yes
CLISTE	R,D					
FLOW	D	yes				
VMEC	D		yes	yes		

Op: R := reconstruction, current profile from measurements
D := direct (or predictive), current profile known

Equilibrium Reconstruction

JET from machinedescription



External fields:

Iron transformer model

Passive structures

Poloidal field coils

(can be replaced by simpler model, see equinox)

Measurements:

magnetic sensors

Polarimetry/Interferometry

MSE

…

Grad – Shafranov equation

$$-\Delta^{*}\Psi = \mu_0 R\left(\sum_p c_p J_p^{plasma}(\Psi) + \sum_k I_k J_k^{ext}\right)$$

interleaved with least – squares

$$\chi^2 = \left\| \frac{1}{\sigma}\left(F^{synthetic}(\Psi; c_p, I_k) - F^{measured}\right)\right\|^2$$

EQUAL: machine geometry

Information obtained with exp2itm, usually stored in shot with run# 1

Magnetic diagnostics	magdiag%flux_loops%setup_floops%position%r, z, phi magdiag%bpol_probes%setup_bprobe%position%r, z polangle
Limiter	limiter%limiter_unit%closed, r, z
Poloidal field system, coils etc.	pfsystems%pfcoils%desc_pfcoils%nelement … %pfelement%pfgeometry%npoints *… %pfelement%pfgeometry%rzcoordinate%r, z* … %pfelement%pfgeometry%rzdrdz … %pfelement%pfgeometry%type, turnsign
Toroidal field coils	torfield%bvac_r, r0
If present on tokamak: Faraday diagnostics	 polardiag%setup_line%pivot_point%r, z polardiag%setup_line%second_point%r, z
Interferometry	interfdiag%setup_line%pivot_point%r, z interfdiag%setup_line%second_point%r, z
Motional Stark Effect (MSE)	msediag_in%setup_mse%rzgamma%r msediag_in%setup_mse%rzgamma%z msediag_in%setup_mse%geom_coef
Ferromagnetic transformer	ironmodel%desc_iron%geom_iron%npoints ironmodel%desc_iron%geom_iron%rzcoordinate%r, z

Machine descriptions for: JET, Tore Supra (tested with EQUAL)

AUG, MAST, TCV (incomplete or not yet tested)

EQUAL: input signals, data

Information obtained with exp2itm, usually stored in shot with run# 1

magdiag%flux_loops%measure%XXX
magdiag%bpol_probes%measure%XXX
magdiag%ip%XXX
magdiag%diamagflux%XXX
pfsystems%pfcoils%coilcurrent%XXX
torfield%bvac_r%XXX

polardiag%measure%XXX
interfdiag%measure%XXX

XXX: value, abserror, relerror (usually vectors)

If abserror=0 and relerror=0 for one measurement, signal is taken out of fit (e.g. faulty sensors)

Further fine tuning with weights in code parameters
1: normal
0: taken out of fit

Warning: number of sensors different for each tokamak
⇒ one set of code parameters per machine
Code itself remains identical !

Equilibrium Refinement: HELENA

High resolution fixed boundary equilibrium modules:

CHEASE, HELENA, CAXE, FIXFREE, …

Equilibrium CPO
e.g. from equilibrium reconstruction

High resolution equilibrium module

HELENA : FEM 3rd order Hermite elements
(EQUAL : finite difference, 2nd order)

Calculation of geometric coefficients

Equilibrium CPO
e.g. for MHD stability module

Input for helena

equilibrium%eqgeometry%boundary%r, z
equilibrium_in(1)%eqgeometry%geom_axis%r
equilibrium_in(1)%eqgeometry%a_minor

equilibrium_in(1)%global_param%psi_ax, psi_bound
equilibrium_in(1)%global_param%toroid_field%r0, b0
equilibrium_in(1)%global_param%i_plasma

equilibrium_in(1)%profiles_1d%psi
equilibrium_in(1)%profiles_1d%pprime
equilibrium_in(1)%profiles_1d%ffprime

Equilibrium codes: present status

Codes: EQUAL

ITM code, SVN tag 0.88, rev. 171

HELENA

http://solps-mdsplus.aug.ipp.mpg.de/repos/HELENA/branches/ets

SVN rev. 403

ILSA: to be clarified

Documentation:

EQUAL: ~zwingman/public/GARCHING2012/doc/EQUAL_refman.pdf

Doxygen type manual solely generated from comments inside Fortran source (contains T_EX-style equations)

HELENA: ~zwingman/public/GARCHING2012/doc/manual.pdf (2007)

Contained in SVN repository

Off-topic: Doxygen and Fortran

Notes on using doxygen for Fortran programs in
~zwingman/public/GARCHING2012/doc/doxygen_for_fortran

1. Requires newer version of doxygen (1.7.6.1 on gateway OK)
2. Doxyfile must be taylored for Fortran programs (OPTIMIZE_FOR_FORTRAN = YES)
3. Basic rules, found by trial and error:

- Comment with !> and !< are processed by doxygen
- Precede subroutine, function and module with comment using !>
- Comments within program text should use !<
  e.g. `real(R8), pointer :: rc(:)=>null()  !< Radial coordinate`
- Useful directives \par, \author etc are explained in the doxygen documentation
  http://www.stack.nl/~dimitri/doxygen/manual.html

Equilibrium and stability chain: present status

Actors: ~zwingman/public/GARCHING2012/equal-helena
equalslicej (code paramaters taylored for JET)
Helena (derived from C.Konz)

Workflow: ~zwingman/public/GARCHING2012/equal-helena/equal_helena_01.xml
(derived from C.Konz)

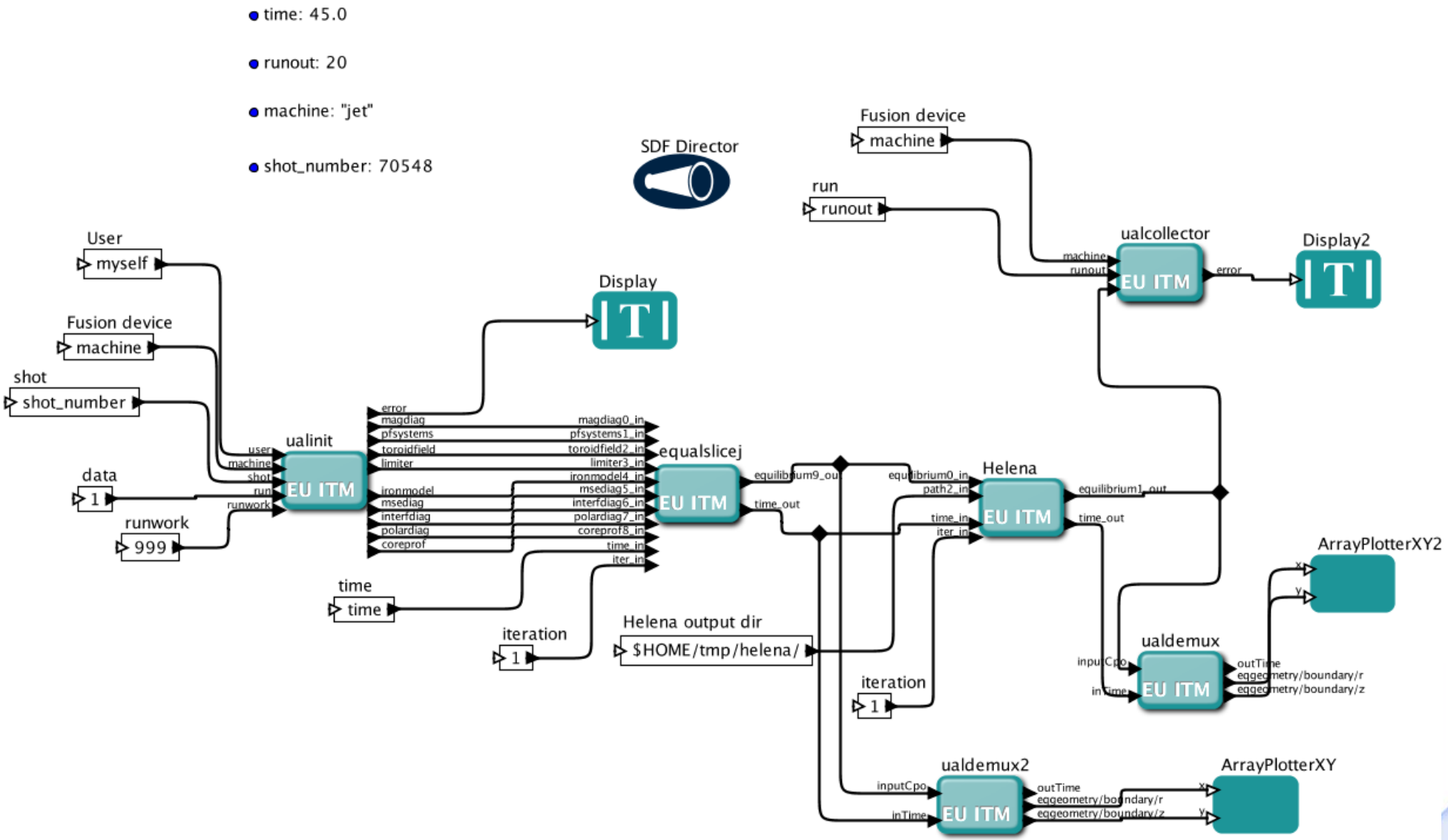
HELENA/ILSA might be replaced by different codes, since:

- not yet ITM codes
- code author(s) no longer work for ITM

$\Rightarrow$ CHEASE, MARS, …

Equilibrium workflow 4.09a

~zwingman/public/GARCHING2012/equal-helena/equal_helena_01.xml



EQ-WF: installation

Actors and Workflow

directory: ~zwingman/public/GARCHING2012/equal-helena/
actors: import_actor [path/]equalslicej
import_actor [path/]Helena
workflow: Copy equal_helena_01.xml to $HOME/KeplerData/workflows/My Workflow

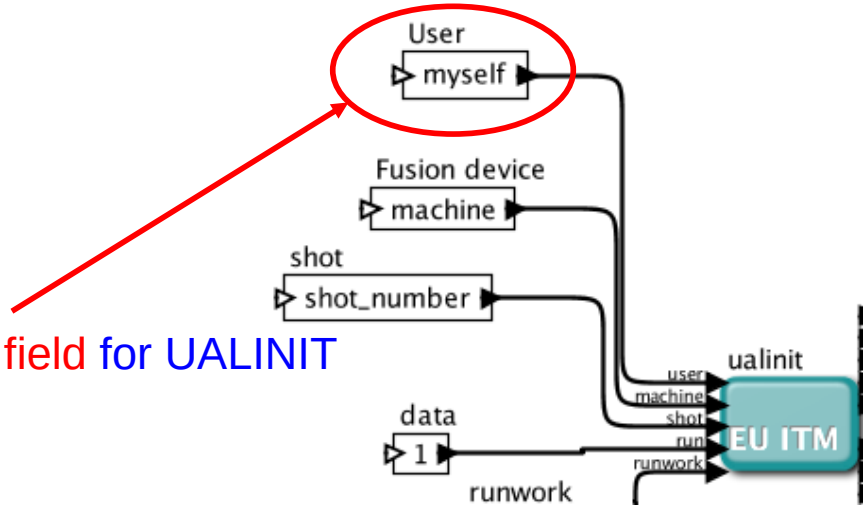
Data

Set UAL (4.09a) and machine (jet) :
source /afs/efda-itm.eu/project/switm/scripts/ITMv1 kepler jet 4.09a

Create directory for jet in your userspace
/afs/efda-itm.eu/project/switm/scripts/create_user_itm_dir jet 4.09a

```
cp \
~zwingman/public/GARCHING2012/itmdb/itm_trees/jet/4.09a/mdsplus/0/euitm_705480001.* \
      $MDSPLUS_TREE_BASE_0
```

EQ-WF: installation



Data access

- Enter your own username into workflow's user field for UALINIT

ArrayPlotterXY

ArrayPlotterXY is not among the usual Kepler tools, but it is very useful.
It works well in time-dependent workflow (updated every step).

If for some reason, it does not appear in your workarea:
Menu → Tools → Instantiate components → Enter **ptolemy.actor.lib.gui.ArrayPlotterXY**
→ actor will appear on workarea

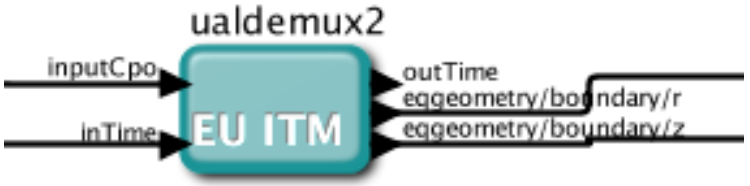
EQ-WF: installation

UALdemux

Very useful tool to extract signals from workflow
→ Right-click on actor → Configure ports → add or modify name

e.g. input: any CPO (must be a CPO !)
output: **global_param/li**

Name must correspond to an existing component of input,
otherwise error in workflow



EQ-WF: code parameters

Parameter editing

Editing of code parameters with default XMLParamForm:

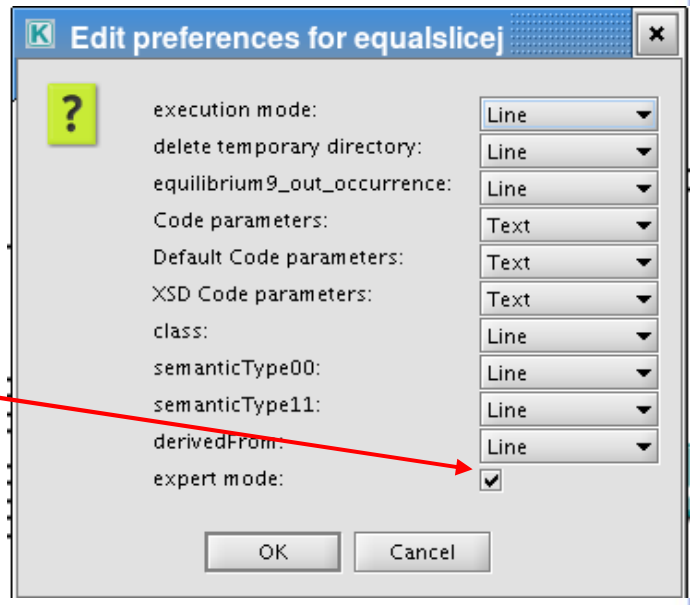
Desirable but editing of arrays a problem, e.g. how to edit weight on magnetic sensor 37 ?

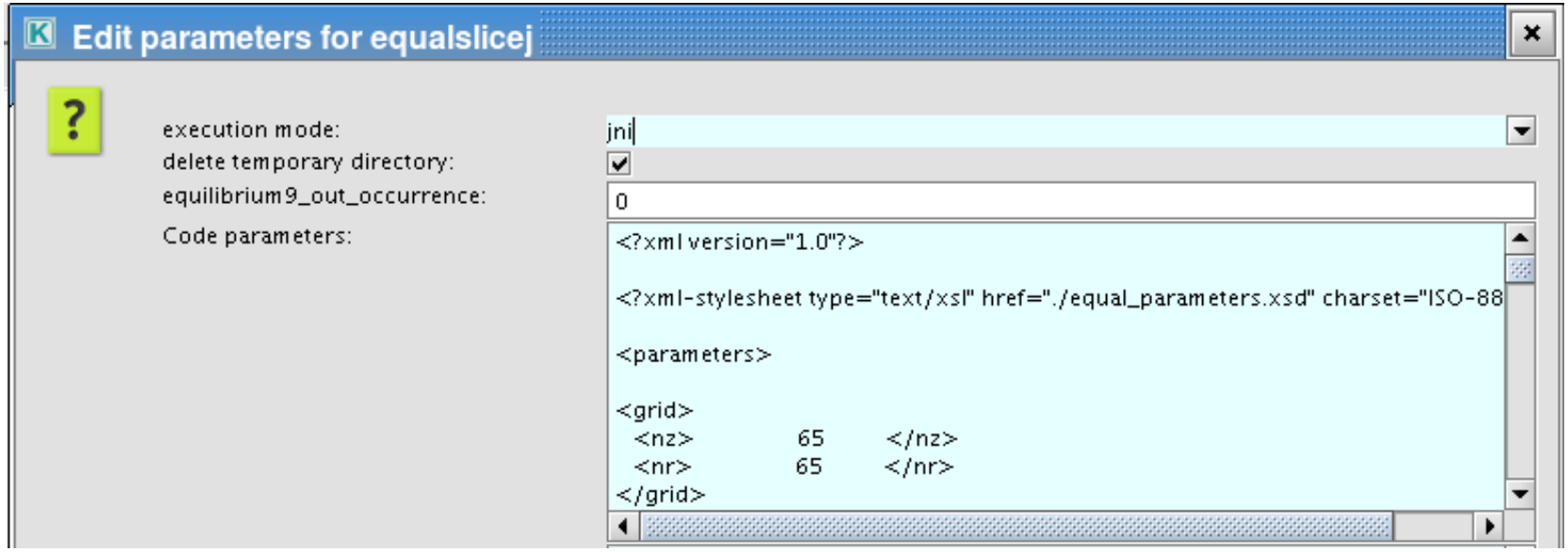
⇒ Enable old way of editing

Click on actors and switch in preferences to expert mode

-> Code parameters editable as text

Try yourself !





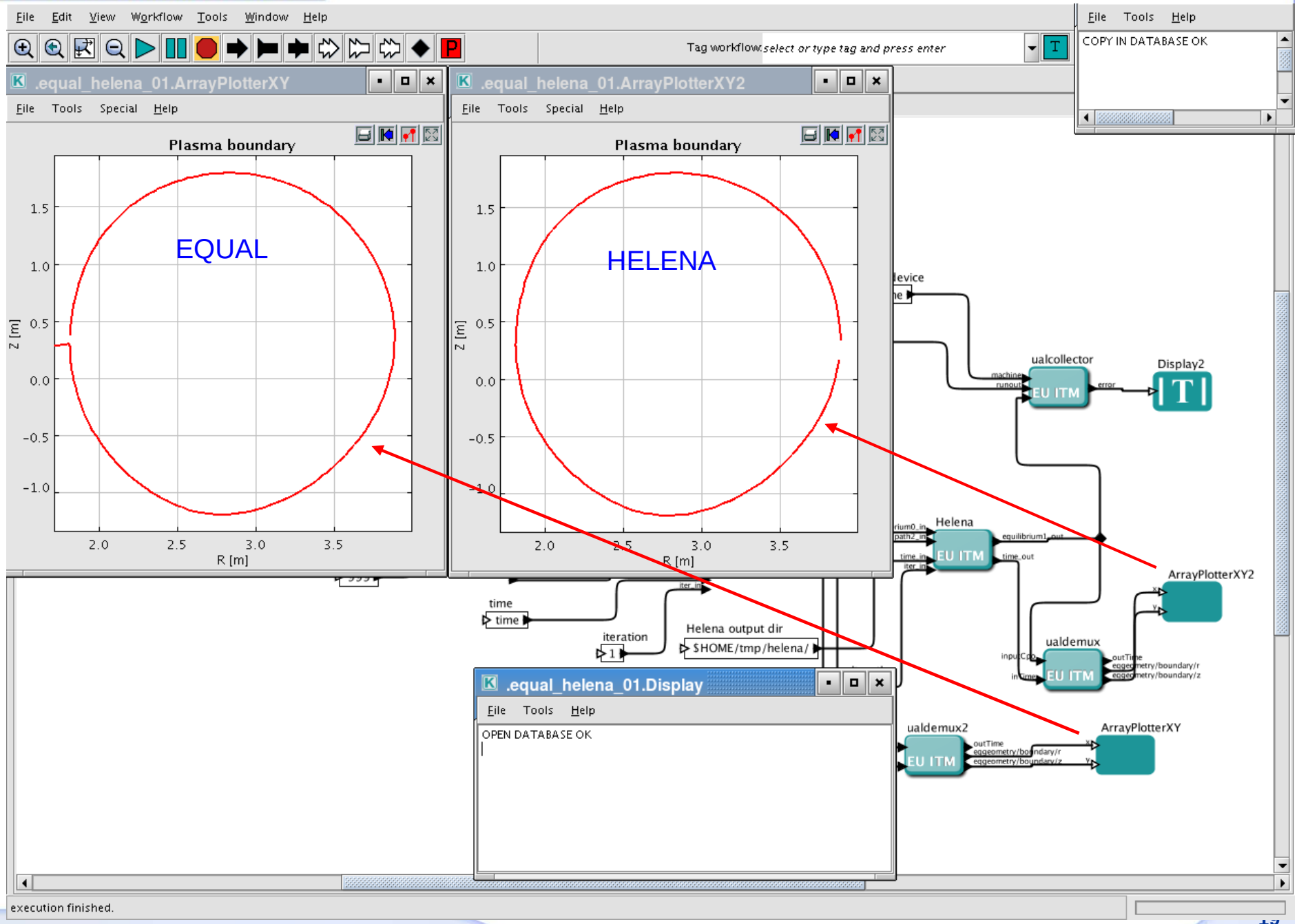
EQUAL code parameters

Some important parameters

```
<grid>
      nz    65        gridsize in vertical direction
      nr    65        gridsize in radial direction

<weight>
      flux            1.0  1.0  1.0  1.0  1.0    1.0  1.0  1.0  1.0  1.0  ….
      bpol            1.0  1.0  1.0  1.0  1.0    1.0  1.0  1.0  1.0  1.0  ….
      faraday         1.0e-8  1.0e-8  1.0e-8  1.0e-8  1.0e-8  1.0e-8  1.0e-8  1.0e-8
      ne              0.1  0.0  0.1  0.1  0.1      0.1  0.1  0.1
<profile>
      npprime          2
      nffprime         2
      constraint_weight  10.1
```

EQUAL-Helena workflow



EQ-WF: use and exercises

1. Run
2. Change time
3. Add an actor to display q(psi)
4. Try to change parameters, e.g. gridsize

Batch workflow execution

For production runs, workflows can be run in batch without GUI
Following steps can be applied to any workflow

- Take out display actors etc.
- Write a batch script, see below (…GARCHING2012/equal-helena/equal_helena.sh)
- Run with bsub –o out equal_helena.sh
- Postprocess

```
#!/bin/csh
source /afs/efda-itm.eu/project/switm/scripts/ITMv1 kepler jet 4.09a
cd $HOME/kepler
kepler.sh -runwf -nogui -nocache \
        $HOME/KeplerData/workflows/MyWorkflows/equal_helena_03.xml >& \
        $HOME/keplerbatch/jobout
```

Python visualisation

Python code can be used to display workflow results with a standalone program
e.g. ~zwingman/public/GARCHING2012/python/plotq.py

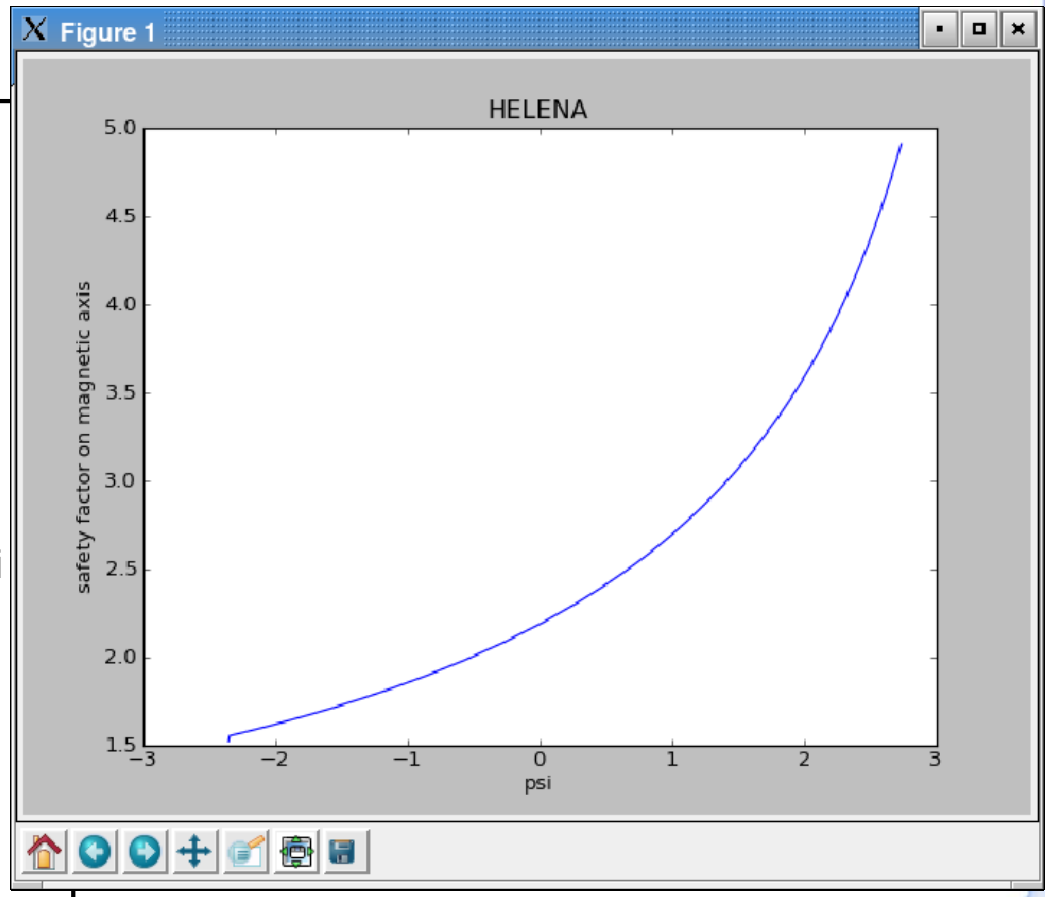
- source /afs/efda-itm.eu/project/switm/scripts/ITMv1 4.09a jet
- python plotq.py

```
import numpy
import pylab
import scipy.io
import ual

equil=ual.itm(70548,20,70548,0)
equil.open()
equil.equilibriumArray.get()

psi = equil.equilibriumArray.array[0].profiles_1d.psi
q   = equil.equilibriumArray.array[0].profiles_1d.q

pylab.plot(psi,numpy.abs(q))
pylab.title('HELENA')
pylab.xlabel('psi')
pylab.ylabel('safety factor on magnetic axis')
pylab.show()
```



To be done and Homework

1. Run for different tokamak (Tore Supra)

Complete tests with exp2itm and try EQUAL

2. Replace HELENA with e.g. CHEASE

Get CHEASE actor
Replace HELENA in chain and try

3. Add ILSA or equivalent stability code

Clarify code ownership (ILSA)
Find input fields for ILSA (or equivalent)
Create actor and try

4. Code parameters

5. Verification and Validation