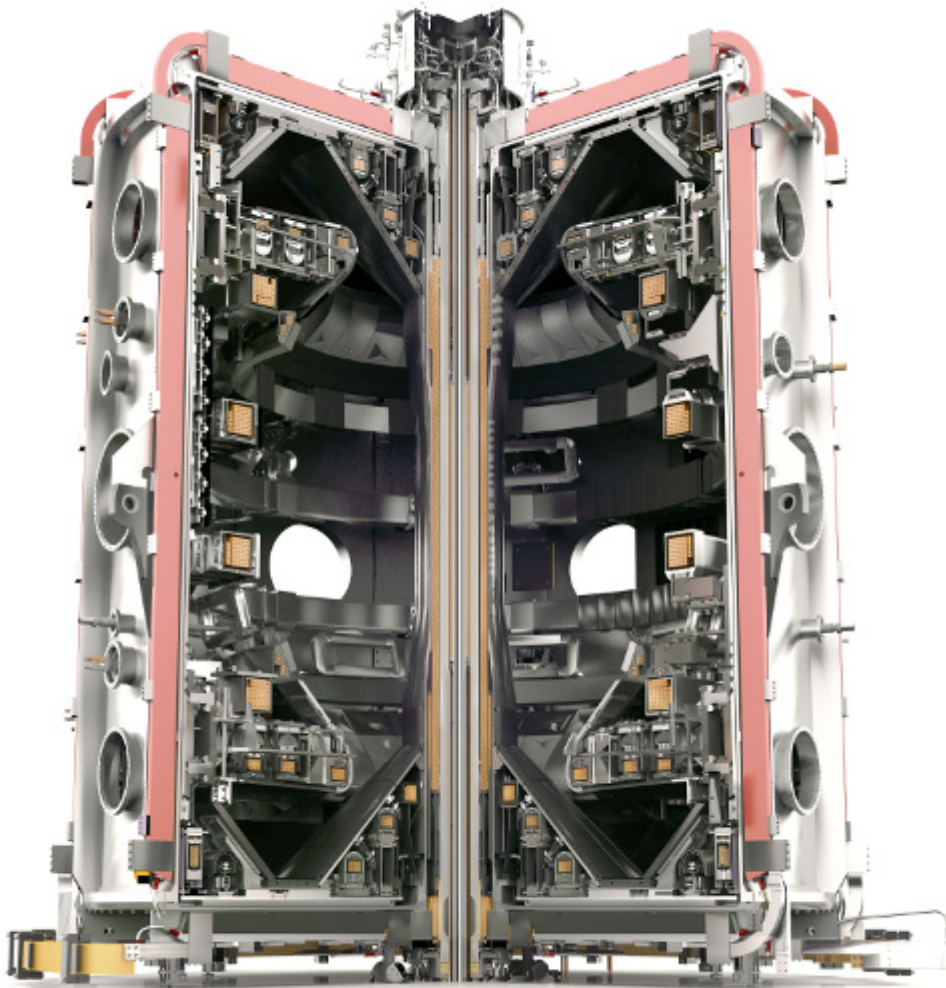


# The MAST-U TRANSP Submission System

---



<b>Training tools</b>	<b>4</b>
<b>Preliminaries</b>	<b>4</b>
Requesting access (1st time users)	4
Acquiring an OSG Certificate	4
<b>Introduction</b>	<b>5</b>
<b>UDA</b>	<b>5</b>
<b>OMFIT</b>	<b>6</b>
FREIA - CCFE	6
PORTAL - Princeton	6
OMFIT Layout	7
<b>kineticEFITtime Module</b>	<b>8</b>
<b>Iteration: 0</b>	<b>9</b>
Generating an Equilibrium	9
Loading in UDA equilibria	9
Running EFIT++	10
Pressure constraint	12
EFIT++ grid	12
Dalpha boundary constraint	12
MSE q-profile constraint	12
Generating gEQDSK files	15
Plotting equilibrium	16
Equilibrium COCOS	19
<b>Iteration: 1</b>	<b>19</b>
Fitting profile data	19
Fetching data	20
ELM filtering	21
Data slicing	24
Symmetrization of Thomson scattering	25
Channel Selection	26
Profile fitting method	28
Electron density fit defaults	30
Electron temperature fit defaults	31
Ion temperature and velocity fit defaults	33
Postfit derived quantities	34
Stretching profiles	34
Storing data	36
Plotting profiles	36

Running TRANSP	37
Time settings	39
Radius	40
Equilibrium	40
Plasma	41
Heating	41
Neutral Beams	42
Low Fidelity	43
Medium Fidelity	43
High Fidelity	43
Anomalous fast ion diffusion	43
ACFILE	44
Sawteeth	44
Pellets	44
Submitting the TRANSP run	44
Analysing the TRANSP run	45
Pressure constrained EFIT++	46
Comparing to the previous iteration	47
<b>Iteration: 2</b>	<b>49</b>
TRANSP analysis run	49
<b>Saving your work and trading runs</b>	<b>50</b>
<b>Useful Advice</b>	<b>50</b>
<b>Appendices</b>	<b>51</b>
First Time Settings	51

## Training tools

A workshop was held and recorded on the 1st February 2022. You can watch the workshop here <https://www.youtube.com/watch?v=5lflr7Fcsjk>

The associated slides can be found here

<https://docs.google.com/presentation/d/1H-54iuLOnex4-p7ZyE5Zs0CdFnPmhjzM/edit?usp=sharing&oid=113002904134993723729&rtpof=true&sd=true>

All the information outlined in the workshop can also be found in the document.

## Preliminaries

### Requesting access (1st time users)

In order to run TRANSP from off-site (not through direct PPPL log-in to portal) you need to obtain a PPPL grid certificate. To obtain a Grid Certificate please follow the directions on the PPPL website: <https://w3.pppl.gov/transp/CILogon/Instructions.html>

Using CILogin requires an account with a valid identity provider. One could use a google account for this. If one does not have such an account or does not wish to use google then we recommend creating a Github account (<https://github.com/>) and using this instead. The advantage here is that the OMFIT source code is stored on Github, consequently if one has a github account one can submit issues and report problems directly to the OMFIT authors. To gain access to the OMFIT source code and submit related issues on GitHub, fill out the [OMFIT user agreement](#), including your GitHub username.

In addition to a FusionGRID certificate one also needs to be a member of the TRANSP users group at CCFE. Please contact the CCFE TRANSP RO requesting membership of the group.

## Introduction

In order to perform a TRANSP simulation two things are required: A TRANSP namelist, which contains a set of switches configuring how the simulation is to be run, and a

dataset, typically containing information such as the electron density and temperature, ion temperature and rotation profiles if available, Z effective and bolometry data etc. This dataset must be provided in a particular file format that TRANSP understands known as a *Ufile*.

The new TRANSP submission system for MAST utilises the OMFIT scientific workflow system <https://omfit.io/>. This provides a graphical user interface and tools which assist users in preparing both the TRANSP namelist and the required input data for a TRANSP simulation before submitting it. The GUI hides a lot of the complexity of understanding the meaning of the numerous namelist switches that can be used to configure a TRANSP run and OMFIT handles the conversion of input data to Ufiles behind the scenes. Once the namelist has been configured and the required data has been read in, the run can be submitted to the FusionGRID at Princeton (PPPL <http://w3.pppl.gov/transp/>), monitored during execution, and fetched on completion.

## UDA

Inter-shot data from MAST is collected and processed in the TRANSP module in OMFIT by using the data access framework tools (UDA) formerly known as IDAM. You will only be able to connect to the external CCFE UDA server if the IP you are connecting from has been white-listed on the CCFE firewall. To get one or more new IP addresses white-listed you should email the required IPs to [Jonathan Hollocombe](#). Once the IP address is white-listed, users can use OMFIT to **access MAST data without needing to remotely connect to the FREIA CCFE Linux servers**.

Once your IP address is white-listed, a local installation of the UDA client is required. On FREIA, a local installation of UDA is loaded into OMFIT via a wrapper start-up script e.g.:

```
export UDA_HOST=idam3.mast.ccf.ac.uk
export UDA_PORT=56565
export PKG_CONFIG_PATH=/common/projects1/UDA/2.1.0/lib/pkgconfig
export LD_LIBRARY_PATH=/common/projects1/UDA/2.1.0/lib:$LD_LIBRARY_PATH
export PYTHONPATH=/common/projects1/UDA/2.1.0/lib/python2.7/site-packages:$PYTHONPATH
```

It is also possible to install the UDA client directly into the OMFIT installation. This has been demonstrated on the Princeton PORTAL cluster (contact [Sterling Smith](#) for further information).

## OMFIT

There are two versions of OMFIT that should be used: the 'master' version and the 'mastu\_develop' (or development) version. The master version is updated every few months and should be quite stable. The mastu\_develop version is less stable but will have more recent updates, bug fixes, and features. Most importantly, it has been updated to handle MAST-U and MAST data. For MAST analysis, it is recommended that users usually use the Master version., but for MAST-U analysis the mastu\_develop module should be used

## FREIA - CCFE

To start OMFIT first log into a machine on FREIA. Then to start the mastu\_develop version of OMFIT type:

```
module purge
```

```
module load omfit/master
```

```
omfit
```

## PORTAL - Princeton

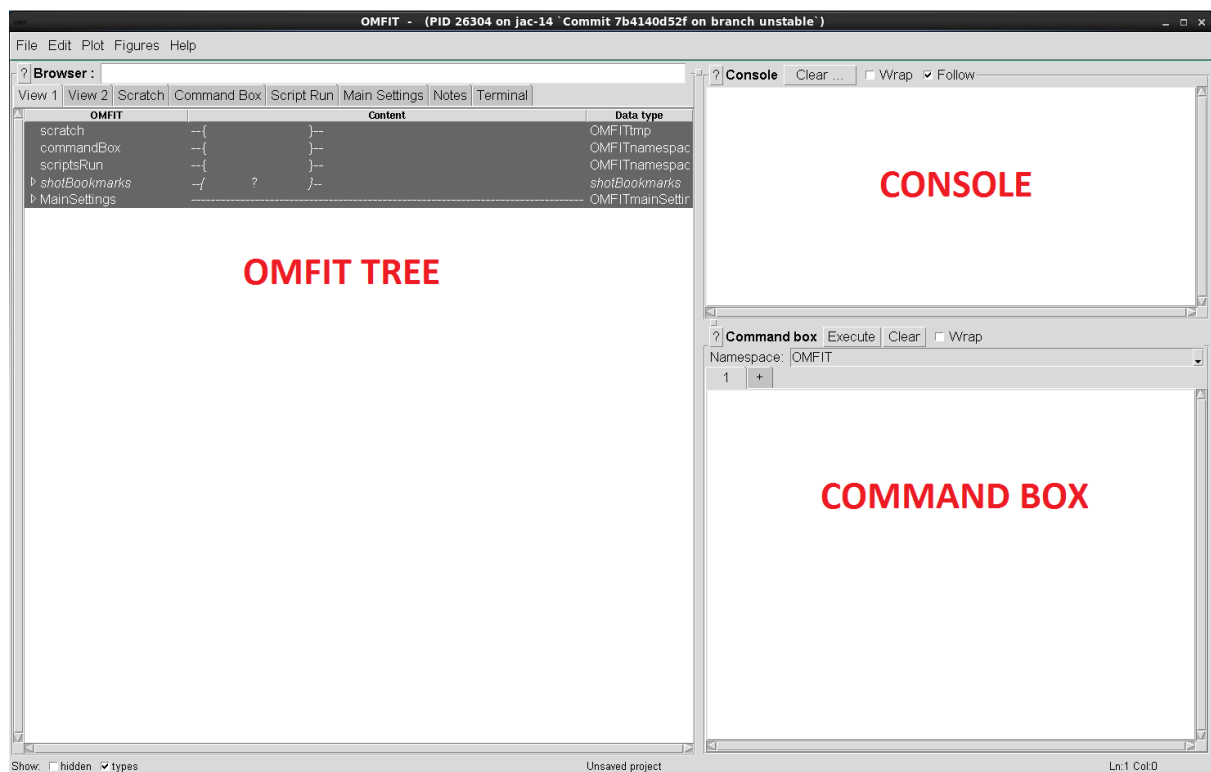
To start OMFIT first log into a machine on PORTAL. The OMFIT installation is maintained in a module which can then be loaded in by typing:

```
module load omfit/unstable
```

```
omfit
```

## OMFIT Layout

These commands will open the main OMFIT window shown below. On the introduction window click on 'Continue to OMFIT >>>' to proceed. If this is your first time using OMFIT see the [First Time Settings](#) section at the end of this document.



The main OMFIT window consists of three main parts, only two of which will be of interest to most users. In the top right of the window is the **console**. This is the area where OMFIT prints out any status information or error messages produced during the preparation of the TRANSP run. You should pay attention to what is printed here. In the lower right is the **command box**. This is a python interpreter and allows the user to execute python commands in OMFIT. This can be useful for directly modifying tree entries but will not be useful to most users and can usually be ignored unless doing development work. Lastly, on the left hand side is the **OMFIT tree**. This is a hierarchical data structure that contains all of the scripts, files, and data items that are used or created during the submission process. Entries in the tree with little arrows next to them contain other sub-entries beneath them. Clicking on the little arrow shows what is contained. Only a few entries in the tree will be of interest to TRANSP users and will be described in this document. For the most part the users will be able to just use the GUI to do things and will not have to care much about the tree structure, however knowing a little about what things are stored and where can be beneficial.

# kineticEFITtime Module

At present the OMFIT tree does not contain very much. As we are interested in doing a TRANSP run we must read in OMFIT's kineticEFITtime module. This will populate the tree with the scripts needed to submit a TRANSP run. To do this use the file menu at the top left to do:

File->Import Module

Then on the dialogue box that appears, scroll down and select the kineticEFITtime module, then click on 'Load'. An entry will now appear at the top left in the OMFIT tree. Clicking on its arrow shows you what's inside! The kineticEFITtime module is actually a collection of OMFIT modules, including EFIT++, OMFITprofiles, and TRANSP. These modules are used to calculate the equilibrium, fit the profile data, and prepare the input files for TRANSP, respectively.

To start preparing a TRANSP run we need to load up the GUI. To do this, double click on the top level kineticEFITtime entry in the tree. You should see the following window, which allows you to set the Device, Shot and time:



OMFIT does not distinguish between MAST and MAST-U. Enter the device as 'MAST' and type the shot number you are interested in. If you make a valid entry the box will turn **green**. **You then need to PRESS ENTER to confirm the entry**. This is always true in OMFIT, you must press enter in entries to set them. A common mistake is to write something in a box, not press enter and then not actually set a value.

In this example, we use the MAST pulse 45425. For the Times, you can enter a python expression, such as: `arange(500,610,10)` which will produce an array from 500 ms to 600 ms with 10 ms intervals.



After entering the shot and time details, the GUI will expand to provide you with new options and tabs. The kineticEFITtime module works on the basis of iterations. In the first iteration, or step 1, the GUI requires you to calculate the equilibrium. In the second iteration, this equilibrium is used to map the profile data onto rho, run TRANSP, and then re-run EFIT++. In the third iteration, this step can be repeated, and so on.

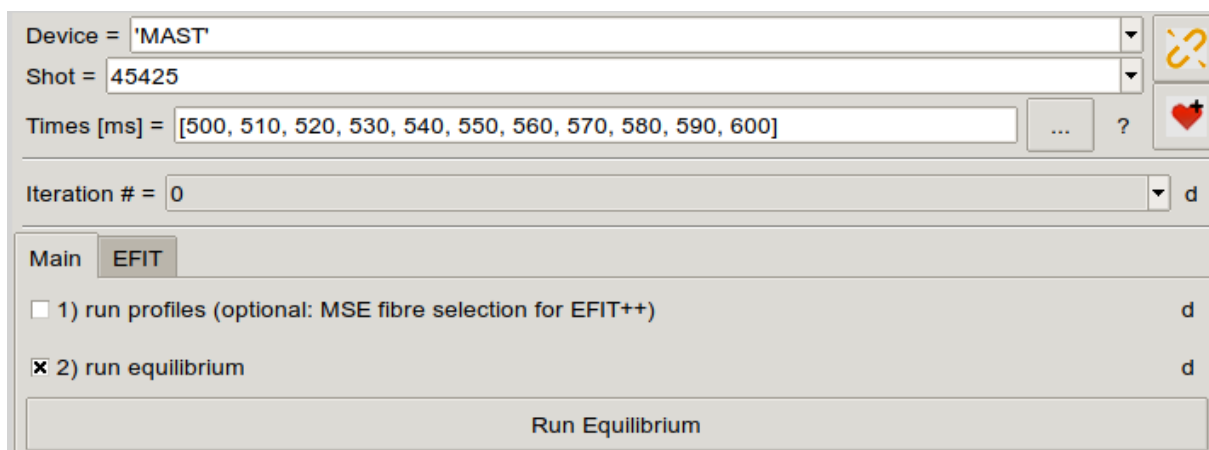
## Iteration: 0

### Generating an Equilibrium

Even when using the TEQ internal equilibrium solver in TRANSP, one must still read in a plasma equilibrium as the plasma boundary is needed. The scripts require a g-eqdsk file. There are two options for producing these files: the first option is to fetch the inter-shot scheduler data using UDA and the second option is to calculate the equilibrium using the EFIT++ code. **Note that only the first option (inter-shot equilibrium data) is possible when running OMFIT on external servers (e.g. PORTAL). The second option is only available for MAST shots, not MAST-U.**

### Loading in UDA equilibria

For MAST-U running EFIT++ is not yet available and the only option at the moment for an equilibrium is to load a magnetics only equilibrium from UDA. Uncheck 1) run profiles and switch to the EFIT tab.



Device = 'MAST'

Shot = 45425

Times [ms] = [500, 510, 520, 530, 540, 550, 560, 570, 580, 590, 600]

Iteration # = 0

Main EFIT

☐ 1) run profiles (optional: MSE fibre selection for EFIT++)

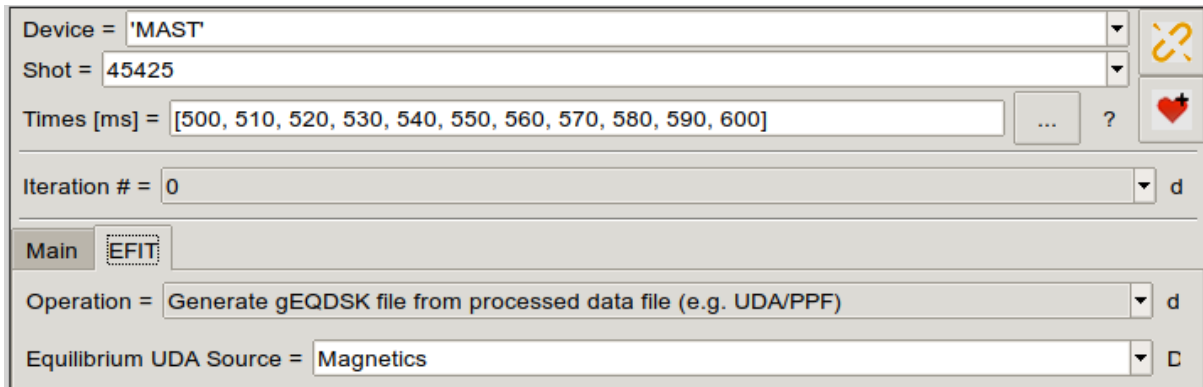
☒ 2) run equilibrium

Run Equilibrium

Under the operation section, select "Generate gEQDSK file from NETCDF file" and put the appropriate path. For shots above 44850 the data is (usually - if not then use the

method above) available on UDA. Under the operation section, select “Generate gEQDSK file from a processed data file (e.g. UDA/PPF)” and select the UDA source as “Magnetics”.

Go back to the main tab and select “Run equilibrium”. This will read the UDA EFIT runs and create the necessary GEQDSK files.



The screenshot shows the 'Main' tab of the kineticEFITtime GUI. The configuration is as follows:

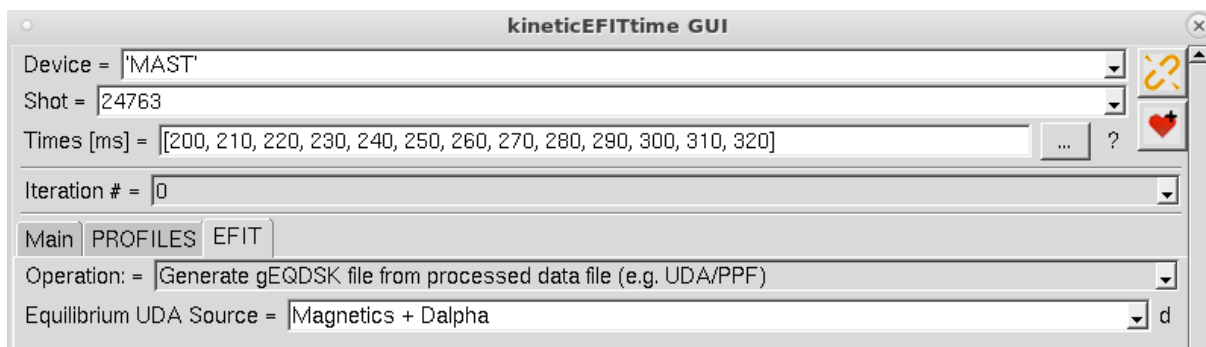
- Device = 'MAST'
- Shot = 45425
- Times [ms] = [500, 510, 520, 530, 540, 550, 560, 570, 580, 590, 600]
- Iteration # = 0
- Operation = Generate gEQDSK file from processed data file (e.g. UDA/PPF)
- Equilibrium UDA Source = Magnetics

For MAST-U simulations you can skip straight to [Iteration 1](#)

## Running EFIT++

Below is an example on how to run EFIT++ for a MAST shot 24763.

The EFIT++ module acts as a wrapper GUI for the EFIT++ scripts that are maintained on the Freia Linux servers at CCFE. This module can be used to run EFIT++ with different run modes (e.g. magnetics, Dalphi, TS, MSE) or it can be used to load in g-files/EFIT++ netCDF files produced by the user. Clicking on the EFIT tab in the GUI will provide you with an operation drop-down box which allows us to either run EFIT++, load equilibrium from UDA, or load in a previously calculated EFIT++ output file. With ‘Generate gEQDSK file from processed data (e.g. UDA/PPF)’ selected, the GUI is very minimal since it takes everything from UDA:



The screenshot shows the 'EFIT' tab of the kineticEFITtime GUI. The configuration is as follows:

- Device = 'MAST'
- Shot = 24763
- Times [ms] = [200, 210, 220, 230, 240, 250, 260, 270, 280, 290, 300, 310, 320]
- Iteration # = 0
- Operation = Generate gEQDSK file from processed data file (e.g. UDA/PPF)
- Equilibrium UDA Source = Magnetics + Dalphi

In this example, we will instead run the EFIT++ code to generate the gEQDSK files. To do this, select 'Generate gEQDSK file EFIT++ run' and the GUI will now offer more options, as shown below.

The screenshot shows the 'kineticEFITtime GUI' window. At the top, there are input fields for 'Device = MAST', 'Shot = 24763', and 'Times [ms] = [200, 210, 220, 230, 240, 250, 260, 270, 280, 290, 300, 310, 320]'. Below these is 'Iteration # = 0'. A tabbed interface shows 'Main', 'PROFILES', and 'EFIT' tabs, with 'EFIT' currently selected. Under the 'EFIT' tab, 'Operation: = Generate gEQDSK file from EFIT++ run' is selected in a dropdown menu. Below this, 'EFIT++ Run Mode = Dalpha + Magnetics + Pressure (TS) + MSE' is also selected in a dropdown menu. An 'Additional options' section contains a title bar 'OMFIT['kineticEFITtime']['EFIT++']['GUIS']['SETTINGGui']' and two buttons: 'Show EFIT++ default run mode options' and 'Reset all options'. Below these are several sub-tabs: 'TRANSP', 'Grid', 'Dalpha', 'Pressure', 'MSE', 'Magnetics', 'Q-profile', and 'Miscellaneous'. The 'TRANSP' sub-tab is active, showing a checkbox 'Use TRANSP pressure constraints' (unchecked), a text field 'Use TRANSP CDF stored on CCFE Linux' with value '2601', a checkbox 'No TRANSP CDF file available' (unchecked), a checkbox 'Mirror LFS/HFS pressure profile' (unchecked), and a checkbox 'Manually edit EFIT++ run command' (unchecked). Each checkbox has a small 'd' icon to its right. The bottom of the GUI is a large empty text area.

The EFIT++ Run Mode box is used to setup EFIT++ with a set of internal defaults based on the constraints defined in the run mode. If you leave blank all of the settings in the lower tabs, then it will use the internal defaults for the selected run mode. However, if you want to tweak the settings, then you may do so using these tabs. Furthermore, for experienced EFIT++ users, you can edit the run command directly by selecting 'Manually edit EFIT++ run command' at the bottom of the GUI.

In this example, we are going to generate the equilibrium with the following constraints: magnetics, Dalphi (to define the LCFS position), pressure (from Thomson scattering/TRANSP), and MSE (for q-profile).

### Pressure constraint

The pressure constraint in EFIT++ can either be set by the electron temperature and density measurements from Thomson scattering, or it can be set from the calculated pressure profile from TRANSP output. Since the Thomson scattering data may underestimate the total pressure in the core due to the presence of fast ions and impurities, we can set this to just include the outer radii of the plasma. To do this, go to the Pressure tab and type '0 0.6 --prange 1.25 2.0' into the box labelled 'Add prange'. This will set the limits to  $R=0-0.6$  m and  $R=1.25-2.0$  m.

If we had a TRANSP simulation for this shot already generated, then we would firstly want this to constrain the pressure over the full profile. In this case, we would set the prange to '0 2.0'. Furthermore, in the TRANSP tab, we can define the TRANSP sequence number and also choose whether or not to mirror the pressure profile onto the HFS. For now, we ignore the TRANSP tab since we have not generated our TRANSP input yet.

### EFIT++ grid

In the Grid tab, select the Flux function type 'Spline'. The default R and Z grid size is 65 conforming to the condition  $2^{N+1}$ . We could increase these R and Z grids to 129, which is often necessary for generating equilibrium for pedestal stability codes, but in this example we'll keep the default setting of 65.

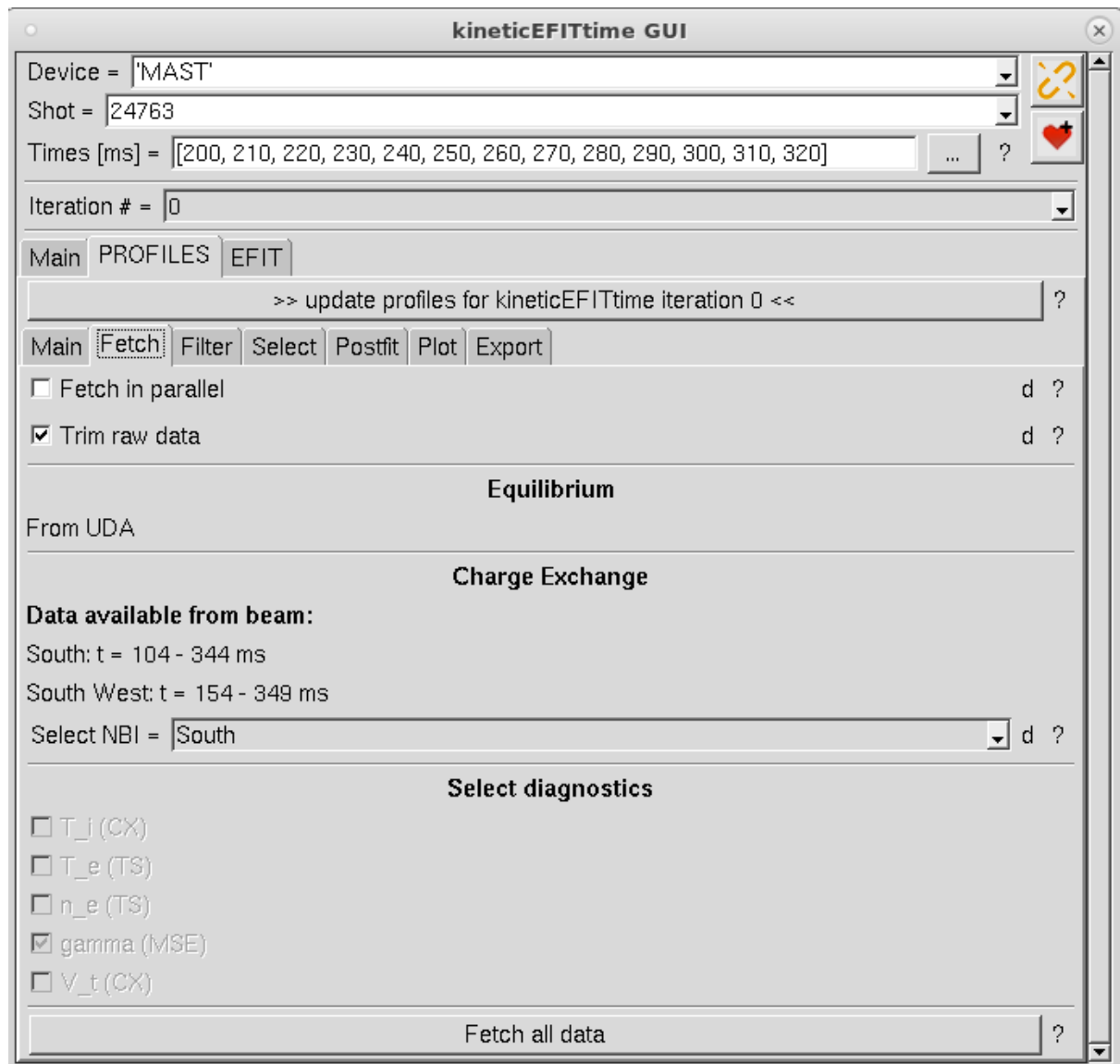
### Dalphi boundary constraint

The Dalphi tab allows us to set the weighting on the boundary condition; we can leave this as the default value.

### MSE q-profile constraint

In the MSE tab, we can define the weight and sightlines which should be skipped by the code. The default option should capture the most commonly used sightlines, however one can use the OMFITprofiles module to automatically set this column. To do this, move to the PROFILES tab (between Main and EFIT) and then to the Fetch tab. Note, this same PROFILES GUI will be used later to fit the profiles, but at this initial phase of

generating the equilibrium, it is only used to assess the MSE measurements. You should see the following GUI:



Click the 'Fetch all data' button to retrieve the raw measurements. At this stage, you can skip to the Select tab. In this window, select 'midplane' from the Subsystem drop down box. This will give you the following options:

**kineticEFITtime GUI**

Device = 'MAST'

Shot = 24763

Times [ms] = [200, 210, 220, 230, 240, 250, 260, 270, 280, 290, 300, 310, 320] ... ?

Iteration # = 0

Main PROFILES EFIT

>> update profiles for kineticEFITtime iteration 0 << ?

Main Fetch Filter Select Postfit Plot Export

Show diagnostics layout

MSE

Subsystem = midplane d

☒ Use Gamma from MSE midplane

Subsample by = 1 d ? Apply Subsampling to subsystem ?

<input checked="" type="checkbox"/> midplane_0	<input checked="" type="checkbox"/> midplane_1	<input checked="" type="checkbox"/> midplane_2
<input checked="" type="checkbox"/> midplane_3	<input checked="" type="checkbox"/> midplane_4	<input checked="" type="checkbox"/> midplane_5
<input checked="" type="checkbox"/> midplane_6	<input checked="" type="checkbox"/> midplane_7	<input checked="" type="checkbox"/> midplane_8
<input checked="" type="checkbox"/> midplane_9	<input checked="" type="checkbox"/> midplane_10	<input checked="" type="checkbox"/> midplane_11
<input checked="" type="checkbox"/> midplane_12	<input checked="" type="checkbox"/> midplane_13	<input checked="" type="checkbox"/> midplane_14
<input checked="" type="checkbox"/> midplane_15	<input checked="" type="checkbox"/> midplane_16	<input checked="" type="checkbox"/> midplane_17
<input checked="" type="checkbox"/> midplane_18	<input checked="" type="checkbox"/> midplane_19	<input checked="" type="checkbox"/> midplane_20
<input checked="" type="checkbox"/> midplane_21	<input checked="" type="checkbox"/> midplane_22	<input checked="" type="checkbox"/> midplane_23
<input checked="" type="checkbox"/> midplane_24	<input checked="" type="checkbox"/> midplane_25	<input checked="" type="checkbox"/> midplane_26
<input checked="" type="checkbox"/> midplane_27	<input checked="" type="checkbox"/> midplane_28	<input checked="" type="checkbox"/> midplane_29
<input checked="" type="checkbox"/> midplane_30	<input checked="" type="checkbox"/> midplane_31	<input checked="" type="checkbox"/> midplane_32
<input checked="" type="checkbox"/> midplane_33	<input checked="" type="checkbox"/> midplane_34	<input checked="" type="checkbox"/> midplane_35

☒ Only plot quantities that will be fit d

Plot MSE (midplane) raw time series

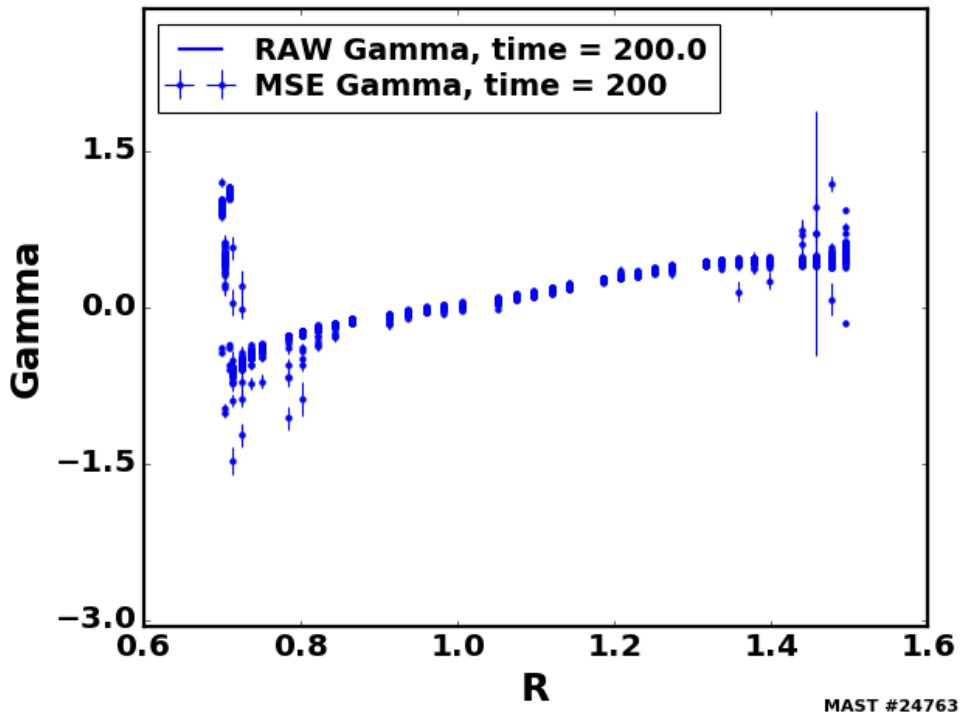
Plot profiles at times = [] d

Select x-axis = R d

Plot MSE profiles

Plot MSE 2D data

Each 'midplane\_##' option corresponds to a radial channel of the MSE diagnostic. To view the radial data, click 'Plot MSE profiles'. This will produce the plot shown below.



This plot is interactive in that you can use the keyboard <- and -> arrow buttons to navigate in time. To avoid the noisy data found in the core and edge, we can deselect the channels `midplane_[0-7]` and `midplane_[34-35]`. Clicking on the Plot MSE profiles button will now update the plot showing data only from the remaining channels.

If you're happy with the selected data, you can now navigate back to the EFIT tab in the GUI, and then back to the MSE tab. There now appears a checkbox button called 'Import deselected channels from OMFITprofiles'. Clicking this then updates the 'Skip sightlines' box. Note, this also deselects channels which are not used by the diagnostic for this shot. If you do not fetch the MSE data from the PROFILES tab, then this import checkbox will not appear and the user will have to either rely on the default condition or set this field manually.

### Generating gEQDSK files

The remaining tabs looking at the magnetics, q-profile, and miscellaneous provide some further options which we can leave for now (trusting the internal defaults).

To generate the gEQDSK files, move back to the Main tab and deselect '1) run profiles (optional: MSE fibre selection for EFIT++)'. Now click 'Run Equilibrium'. The Console window in the top right gives you status updates from the EFIT++ code. Most importantly, this tells us whether the EFIT++ code managed to converge on the defined

time grid. In this example, the code managed to converge on all time steps. Furthermore, if the EFIT++ code has failed for any reason, the error message will appear in red in the Console box.

## Plotting equilibrium

Once the gEQDSK files have been generated, an overview of the data can be plotted using OMFIT. Firstly, to test how well the EFIT++ equilibrium reproduces the input constraints, within the OMFIT tree navigate to kineticEFITtime -> EFIT++ -> OUTPUTS. In this tree, you will see two more subtrees called gEQDSK and CDF. The latter is the direct output from the EFIT++ code, and the gEQDSK tree holds all of the files for each time step.

The screenshot displays the OMFIT application window. The main pane shows a hierarchical tree structure under the 'kineticEFITtime' node. The 'EFIT++' node is expanded, showing 'INPUTS', 'OUTPUTS', 'gEQDSK', and 'CDF'. The 'gEQDSK' node contains a list of files for various time steps (200 to 320), each with a file path and size. The 'CDF' node contains a single file 'efitOut.nc'. The 'Console' pane on the right shows the output of the program, including the path to the local copy of the OMFIT module and the list of times for which gEQDSK files were written. The 'Command box' at the bottom shows the current namespace as 'OMFIT' and the command 'root['kineticEFITtime']['EFIT++']['OUTPUTS']['CDF'].plot(time=0.28)'.

Content	Data type
kineticEFITtime	OMFITmodule
EFIT++	OMFITmodule
INPUTS	OMFITTree
OUTPUTS	OMFITTree
gEQDSK	OMFITcollection
200	FILE: g024763.00200 (85.0kB)
210	FILE: g024763.00210 (85.0kB)
220	FILE: g024763.00220 (85.0kB)
230	FILE: g024763.00230 (85.0kB)
240	FILE: g024763.00240 (85.0kB)
250	FILE: g024763.00250 (85.0kB)
260	FILE: g024763.00260 (85.0kB)
270	FILE: g024763.00270 (85.0kB)
280	FILE: g024763.00280 (85.0kB)
290	FILE: g024763.00290 (85.0kB)
300	FILE: g024763.00300 (85.0kB)
310	FILE: g024763.00310 (85.0kB)
320	FILE: g024763.00320 (85.0kB)
CDF	FILE: efitOut.nc (3.8MB)
SCRIPTS	OMFITTree
GUIS	OMFITTree
LIB	OMFITTree
SETTINGS	OMFITsettings
help	OMFITascii
TRANSP	OMFITmodule
OMFITprofiles	OMFITmodule
EFITtime	OMFITmodule
INPUTS	OMFITTree

```

Writing geqdsK files for times:
[0.2 0.21 0.22 0.23 0.24 0.25 0.26 0.27 0.28 0.29 0.3 0.31 0.32]

Adding eqdsk file: g024763.00200
Adding eqdsk file: g024763.00210
Adding eqdsk file: g024763.00220
Adding eqdsk file: g024763.00230
Adding eqdsk file: g024763.00240
Adding eqdsk file: g024763.00250
Adding eqdsk file: g024763.00260
Adding eqdsk file: g024763.00270
Adding eqdsk file: g024763.00280
Adding eqdsk file: g024763.00290
Adding eqdsk file: g024763.00300
Adding eqdsk file: g024763.00310
Adding eqdsk file: g024763.00320

Finished adding gEQDSK files

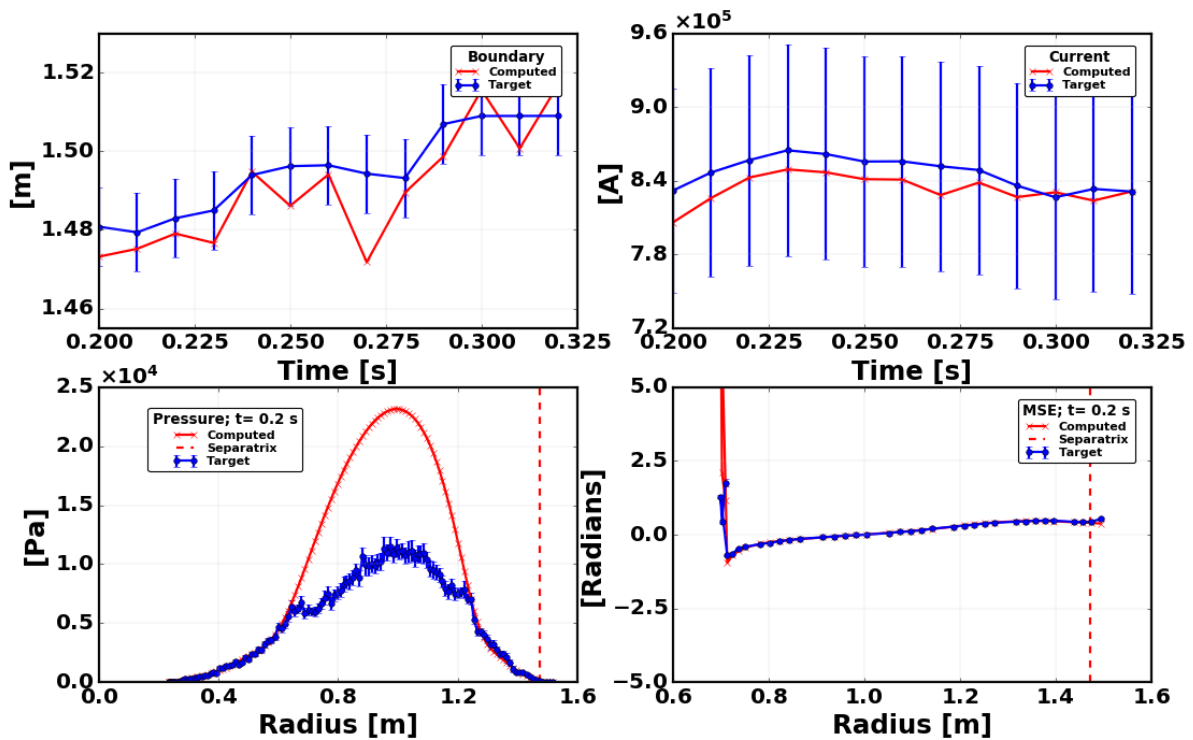
Step 'iteration_0' saved: [EFIT++]
done @ 11/12/2019 10:23:08.710933 took 29.272 s
  
```

```

root['kineticEFITtime']['EFIT++']['OUTPUTS']['CDF'].plot(time=0.28)
  
```

If you double-click on the CDF menu in the tree, it will produce a summary of graphs as shown below.

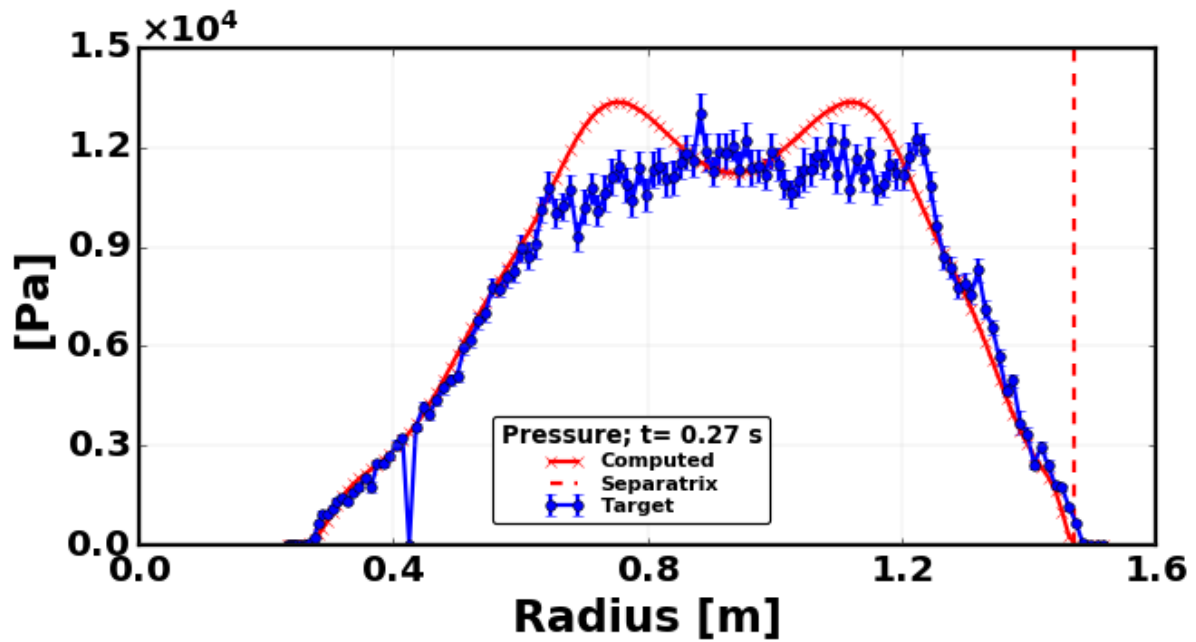




The top two graphs show the boundary and plasma current outputs from EFIT++ and how they compare to the constraints. In the bottom two graphs, you see the radial profile of pressure (left) and MSE angle (right) for the first time slice. If you want to update the time slice, then type into the Command box:

```
root['kineticEFITtime']['EFIT++']['OUTPUTS']['CDF'].plot(time=0.30)
```

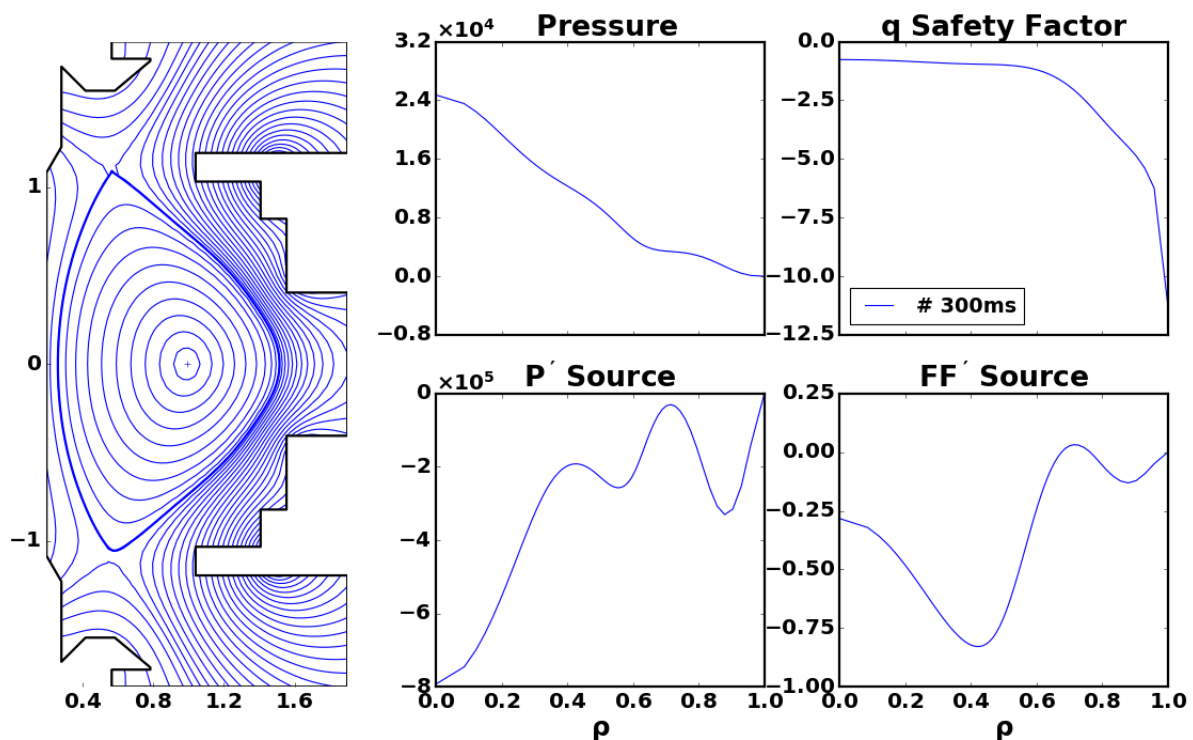
This will update the two lower figures to the new time slice of  $t=0.3$  s. Note that the boundary condition is not met at  $t=0.27$  s. This could indicate that the equilibrium here is not a good solution, despite it converging. Indeed, plotting the pressure for this time slice shows a hollow pressure profile.



If you're not happy with the equilibrium solution, then you can either go back and alter the EFIT++ settings, or you can simply remove this time slice. In this example, we will choose to do the latter. This is simply done by removing 270 from the Times [ms] entry box (remember to click Enter) and then clicking Run Equilibrium. Now you can see that no gEQDSK file was produced for t=0.27s.

To get an overview of the gEQDSK file, open the gEQDSK tree menu and double click on the time slice of interest. This will produce a new set of summary graphs relevant to that

time slice as shown below.



## Equilibrium COCOS

The EFIT++ code outputs equilibria in COCOS=7. For MAST, this means that we have a negative  $q$ -profile and  $\psi$  is decreasing from core to edge. Note that the  $q$ -profile in UDA (for the MAST campaign data) is taken as  $\text{abs}(q)$  which therefore switches the output to COCOS=3.

The DIII-D convention, and also codes associated with TRANSP, require a positive  $q$ -profile and increasing  $\psi$  function. OMFIT automatically handles this by converting the equilibrium data to COCOS=1.

## Iteration: 1

### Fitting profile data

Now that we have an equilibrium for each time step, we can move onto the next iteration. To do this, type 1 into the 'Iteration #' box near the top of the GUI. This will generate new options in the GUI as shown below. Note, all of the data generated in iteration 0 is now stored in the tree under kineticEFITtime -> OUTPUTS -> iteration\_0.

**This is important for provenance.**

MAST #45425 from 500 to 600 ms

Iteration # =

▼ d

Main
PROFILES
TRANSP
EFIT
Plotting

✕ 1) run profiles
d

✕ 2) run transport
d

Transport Module =

▼ C

✕ 3) run equilibrium
d

Run Profiles + Transport + Equilibrium

## Fetching data

Now that we have our equilibrium, we will use this to fit and map our profile data onto normalised poloidal flux  $\psi_N$ . Move to the PROFILES tab and then click on the Fetch tab (which is the same as mentioned in the previous subsection looking at the MSE diagnostic channel selection). Now you should select  $T_i$ ,  $T_e$ ,  $n_e$ , and  $V_t$  whilst de-selecting gamma (this is auto-deselected for MAST-U). For the CX data, for MAST data you can retrieve the data from either the South beam or the South West beam. This can be set in the Select NBI drop-down box. The timings of the beam are shown under 'Data available from beam'. For MAST-U data only the South beam is available. Note, remember here to ensure that 'Fetch in parallel' is deselected. Finally, click 'Fetch and map all data'.

MAST #45425 from 500 to 600 ms

Iteration # = 1 d

Main
PROFILES
TRANSP
EFIT
Plotting

>> update profiles for kineticEFITtime iteration 1 << ?

Main
Fetch
Filter
Slice
Select
Fit
Postfit
Plot
Export

☐ Fetch in parallel ?

☒ Trim raw data d ?

Equilibrium

From OMFIT['kineticEFITtime']['EFIT++']

Charge Exchange

Data available from beam:

South: t = 189 - 995 ms

Select diagnostics

☒ T<sub>i</sub> (CX) d

☒ T<sub>e</sub> (TS) d

☒ n<sub>e</sub> (TS) d

☐ gamma (MSE)

☒ V<sub>t</sub> (CX) d

Fetch and map data ?

## ELM filtering

Moving to the Filter tab, here you can mask certain time points in the raw experimental data by setting time points which are not considered for fitting. The tab is focussed here on ELMs (and in future also on saw-teeth) and is shown below.

MAST #45425 from 500 to 600 ms

Iteration # = 1 d

---

Main **PROFILES** TRANSP EFIT Plotting Export

>> update profiles for kineticEFITtime iteration 1 << ?

---

Main Fetch **Filter** Slice Select Fit Postfit Plot Export

**Event Synchronization Selection**

☐ event\_synchronization d

---

**Manual Time Selection**

Accepted time intervals [ms] =  C ?

---

**ELM selection/rejection filter**

Signal =  C ?

Timescale [ms] =  C ?

Threshold =  C ?

Accepted phase =  C ?

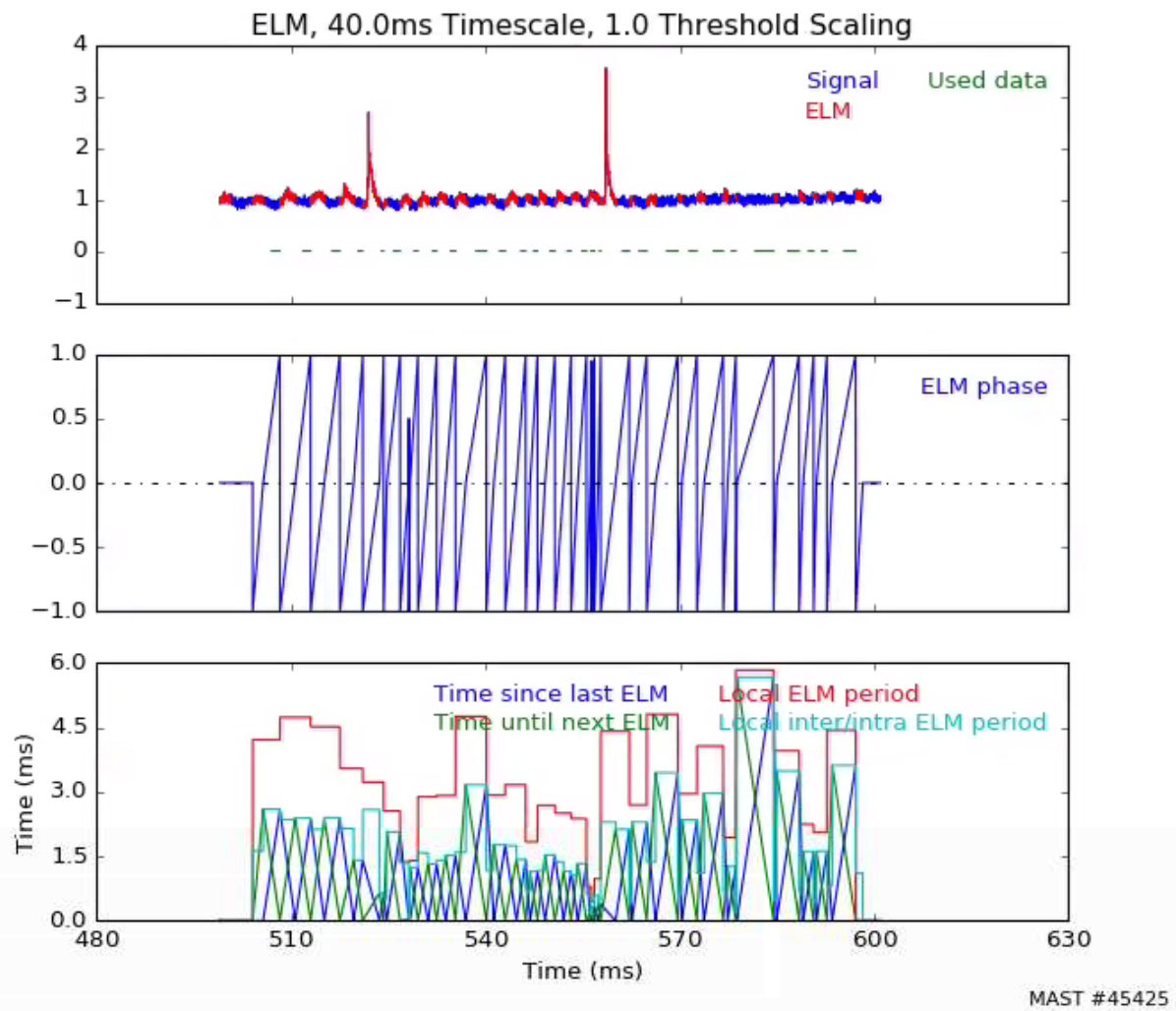
Accept as event-free after [ms] =  C ?

Clearance [ms] =  C ?

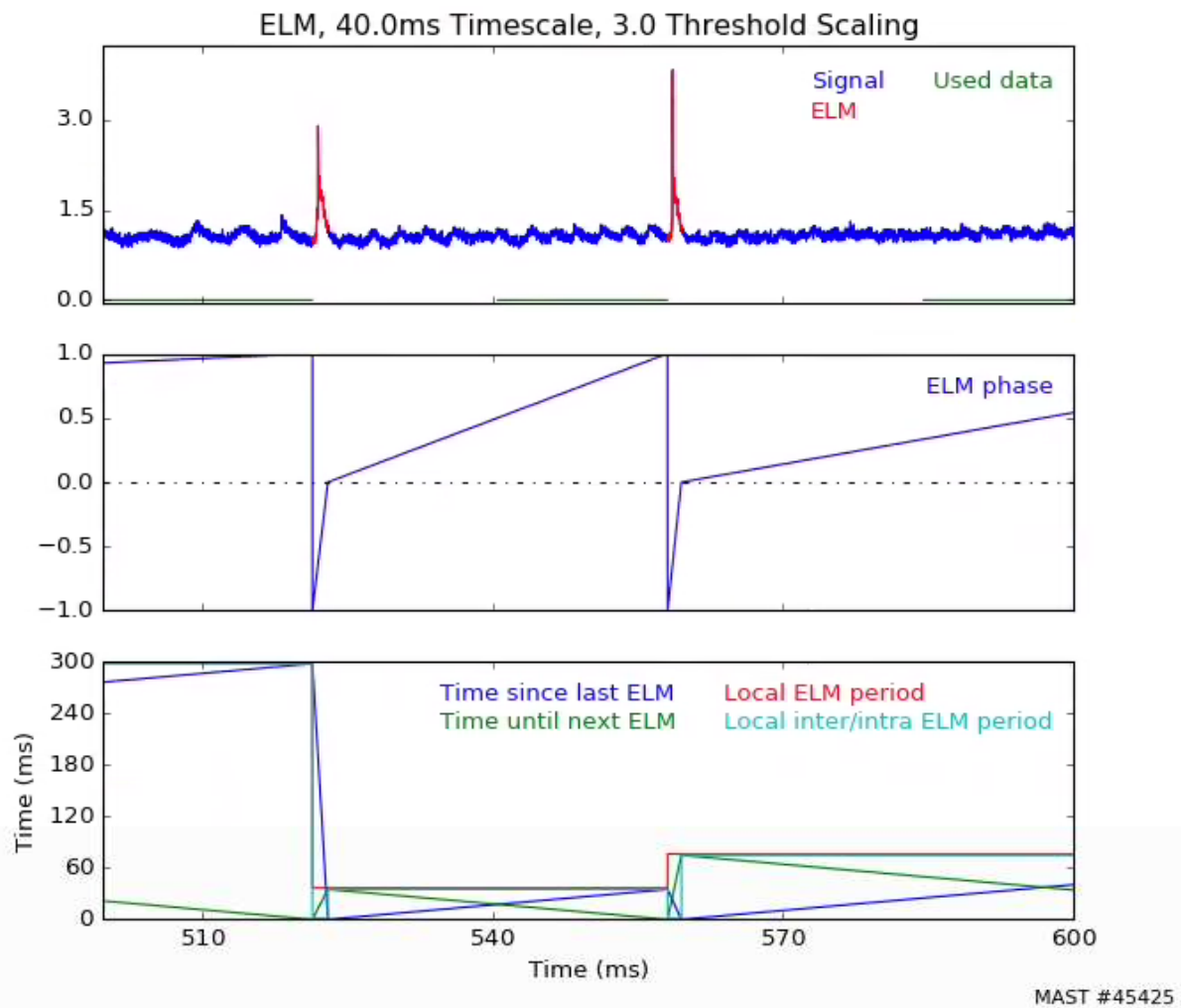
Find & plot ELM events ?

The first part to set is the UDA trace that you want to use to detect the ELMS. For MAST, the default is 'xim\_da/HM10/T\*' which is the midplane Dalphi signal. For MAST-U the default is 'xim/da/hm10/t'.

The Accepted ELM phase details what part of the ELM cycle that you want to focus on. For example, [0,1] looks at the full inter-ELM time window, [0.75,1] focuses only on the last 25% of the inter-ELM phase, while [-1,1] focuses on both the ELM and inter-ELM phase (i.e. no masking is performed). Let's set the accepted ELM phase as [0.0,1] and then click 'Find & plot ELMS'. This produces the figure shown below.



Here, you can see that it has detected the ELMs in the beginning, however it has picked up a bunch of smaller blips as ELMs too. To remove these small bumps, increase the threshold to 3.0. Click again on 'Find & plot ELMs' and now you will see that it has only detected the ELMs. Once you are happy with the filtering settings, go to the GUI and click on 'Reassign ELM filters to raw data'.



## Data slicing

In the next tab called Slice, you now can average the raw data onto the specified time grid. Here you can set the Time averaging [ms] window, but note that if you set, for example, 20 ms then this will take  $\pm 10$  ms. If you select 'Causal window', then it will set the leading half of the time window to zero to avoid generating lag in the filtered signal. In this example, set the Time averaging to 20 ms and select Causal window.

The Window function allows the user to specify the function to use to average the data. For example, this could be set to a Gaussian. The standard function is a hanning function, which we will use in this example.

Selecting Interpolate where data is missing allows you to interpolate any missing data points. This can be important when analysing inter-ELM measurements on MAST, where



there may not be a significant number of data points. In this example, we leave this unchecked.

MAST #45425 from 500 to 600 ms

Iteration # = 1

Main | PROFILES | TRANSP | EFIT | Plotting

>> update profiles for kineticEFITtime iteration 1 <<

Main | Fetch | Filter | **Slice** | Select | Fit | Postfit | Plot | Export

Spatial averaging = []

Time averaging [ms] = 20

Window function = hanning

Convolution error = propagate

☐ Interpolate where data is missing

☐ Causal window

☐ Symmetrize TS

Set separatrix Te = 60.0

Set separatrix Te weight = 0.001

☐ Use LFS only for separatrix Te constraint

☐ Plot symmetrization results

Diagnostic Specific

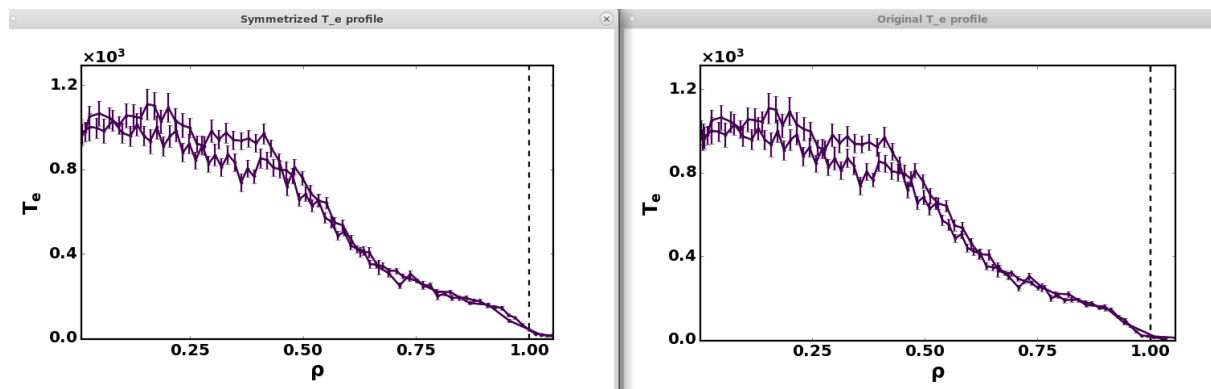
☐ Stitch TS subsystems

Slice all data

### Symmetrization of Thomson scattering

The last point to discuss in this tab is the 'Symmetrize TS' feature shown in the figure above. This feature applies a radial shift function to the Thomson scattering data based on either trying to optimise the match between the LFS and HFS measurements or on the separatrix temperature. You can select the weighting to either the former or the latter (or a mix) by setting the 'Set separatrix Te weight' box. Values of the order 0.001 put emphasis on the matching of HFS and LFS, while values of the order 100 put emphasis on the separatrix temperature. As default, the cost function associated with the separatrix temperature constraint is only applied to both the LFS and HFS data to avoid noisy data in the SOL biasing the shift. However, to only apply this cost function to the LFS data, click the 'Use LFS only for separatrix Te constraint' check box.

To demonstrate the shift, first we can try weighting the shift towards the separatrix  $T_e$  constraint of 40 eV for time slice 250 ms (note that a new instance of the kineticEFITtime module was set up with only one time slice for this particular demonstration). Below, the results from the symmetrization are shown on the left, and the original profile shown on the right. Note that these graphs are produced by clicking the 'Plot symmetrization results' check box. There is some improvement on the symmetrization and now the profiles cross the separatrix at 40 eV (rather than ~15 eV in the original profile).



Now, back to the tutorial example, we will simply work with the raw data and use the Stretch function later to adjust the temperature and density profiles. Therefore unclick the 'Symmetrix TS' box and finally, click 'Slice all data'.

## Channel Selection

Users can deselect radial channels associated with each diagnostic in the Select tab. In this example, we will deselect the HFS core channels from the CX measurements (e.g.  $T_i$  and  $V_t$ ). Go to the CHERS tab and select 'midplane' from the Subsystem drop down menu. Type [20, -1] in "Rangesample by" and click "Apply Rangesampling to subsystem" to select the 20th channel onwards.

CHERS
TS

Subsystem = midplane
C

☒ Use T\_12C6 from CHERS midplane
d

☒ Use V\_tor\_12C6 from CHERS midplane
d

☒ Use omega\_tor\_12C6 from CHERS midplane
d

Subsample by = 1
C
?

Apply Subsampling to subsystem
?

Rangesample by = [0, -1]
C
?

Apply Rangesampling to subsystem
?

☒ midplane\_0
☒ midplane\_1
☒ midplane\_2

☒ midplane\_3
☒ midplane\_4
☒ midplane\_5

☒ midplane\_6
☒ midplane\_7
☒ midplane\_8

☒ midplane\_9
☒ midplane\_10
☒ midplane\_11

☒ midplane\_12
☒ midplane\_13
☒ midplane\_14

☒ midplane\_15
☒ midplane\_16
☒ midplane\_17

☒ midplane\_18
☒ midplane\_19
☒ midplane\_20

☒ midplane\_21
☒ midplane\_22
☒ midplane\_23

☒ midplane\_24
☒ midplane\_25
☒ midplane\_32

☒ midplane\_33
☒ midplane\_34
☒ midplane\_35

☒ midplane\_36
☒ midplane\_37
☒ midplane\_38

☒ midplane\_39
☒ midplane\_40
☒ midplane\_41

☒ midplane\_42
☒ midplane\_43
☒ midplane\_44

☒ midplane\_45
☒ midplane\_46
☒ midplane\_47

☒ midplane\_48
☒ midplane\_49
☒ midplane\_50

☒ midplane\_51
☒ midplane\_52
☒ midplane\_53

☒ midplane\_54
☒ midplane\_55
☒ midplane\_56

☒ midplane\_57
☒ midplane\_58
☒ midplane\_59

☒ midplane\_60
☒ midplane\_61
☒ midplane\_62

☐ midplane\_63

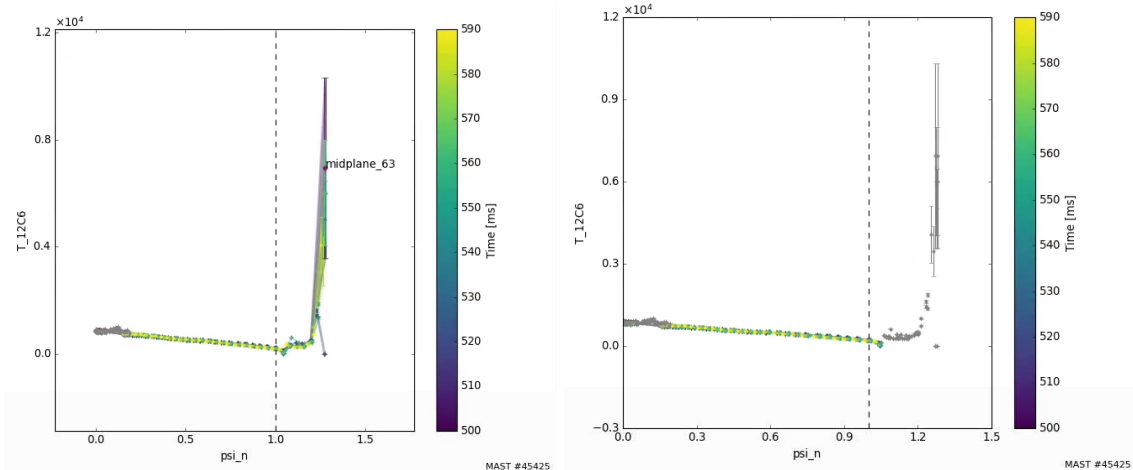
☒ Only plot quantities that will be fit
C
?

Plot CHERS (midplane) raw time series

Plot CHERS sliced data

Plot CHERS 2D sliced data

Click on "Plot CHERS sliced data" and you get the image shown below on the left. Clearly there are a few channels that have spurious data. You can click on the data point to find which channel it is. We can deselect these channels by applying Rangesampling with [20, -6]. Doing this results in the figure on the right.



For the Thomson scattering measurements, we will again deselect the HFS points. To do this, go to the TS tab and select 'midplane' from the Subsystem drop down box. Now deselect the channels midplane\_[0-65] and some of the end channels [121-129].

## Profile fitting method

Now that we have data averaged on to the desired times, we can fit the profile data. Go to the fit tab which appears as shown below.

MAST #45425 from 500 to 600 ms

Iteration # =  d

Main PROFILES TRANSP EFIT Plotting

>> update profiles for kineticEFITtime iteration 1 <<

Main Fetch Filter Slice Select **Fit** Postfit Plot Export

Fit coordinate =  d ?

Fit range =  C ?

Output resolution =  C ?

Load fit settings from exported OMFITsettings file =  File C ?

Fit all with current settings

n\_e omega\_tor\_12C6 T\_12C6 T\_e V\_tor\_12C6

Transform =  C ?

Symmetry =  C ?

Fitting method =  ? w

Bases in psi\_n =  C ?

Space-time correlation ratio =  C ?

☒ Bias psi\_n by smoothed derivative at each time C ?

Additional edge bias =  C ?

Copy Fit Settings to all Fits

Plot 1D sliced data

--> FIT and plot 1D <--

The first part to set is the Fit coordinate. We can choose between normalised poloidal flux or rho. Here we select psi\_n for normalised poloidal flux. Note that rho is defined as:

$$\varphi = \int q d\psi; \rho = \sqrt{2\pi\varphi/\pi B_T}; \rho_N = \rho/\max(\rho)$$

The fit range specifies the radial range. We allow for some points into the SOL by setting this as [0,1.1].

Now we can select a fitting method for each diagnostic measurement. First, move to the n\_e tab. The Transform option allows the user to transform the data using e.g. log, sqrt,

or hyperbola. This can be useful to ensure positive fits and can also help to improve the overall fit. Of course, if the measurement can go negative (e.g. velocity) then the Transform should be left as None.

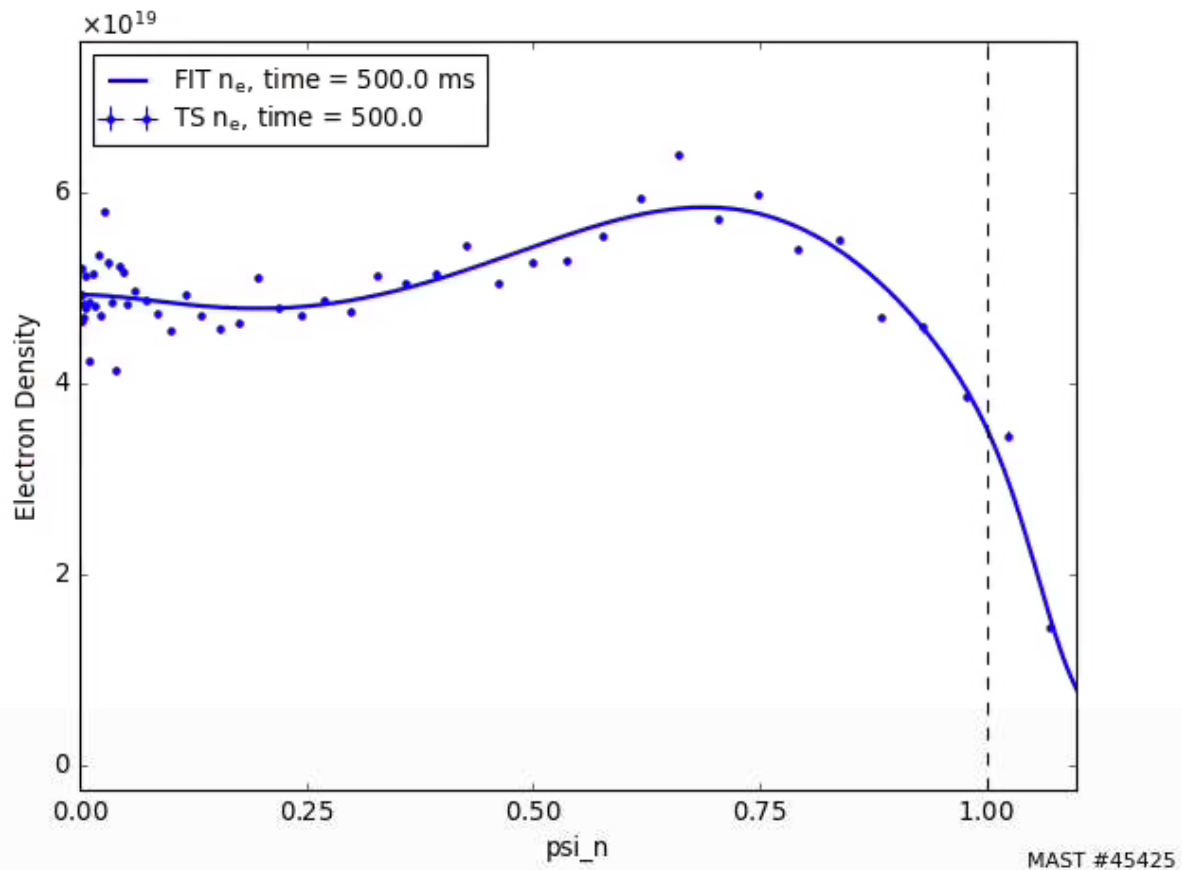
There are many different Fitting methods to choose from. For an overview of each one, it is recommended to read the [OMFITprofiles module tutorial document](#).

### Electron density fit defaults

For H-mode scenarios, where there is a pedestal, it is typically best to use 'Core polynomial & edge exponential (mtanh) blend' as the fitting method. The fit settings used in this example are shown below .

n_e		omega_tor_12C6	T_12C6	T_e	V_tor_12C6
Fit times =					
Transform =	log				
Symmetry =	Even				
<input type="checkbox"/> Uniform radial resampling of data prior fitting					
Fitting method =	Core polynomial & edge exponential (mtanh) blend				
Core polynomial order =	3				
Min pedestal width =	0.001				
Max pedestal width =	0.05				
Min core-pedestal blend width =	0.04				
MTANH sym. guess =	0.98				
MTANH sym. min =	0.96				
MTANH sym. max =	1.1				
<input checked="" type="checkbox"/> Constrain fit to have zero on-axis gradient					
On-axis value constraint (eV) =	None				
<input checked="" type="checkbox"/> Fit edge (x > 1) values					
Edge value constraint (eV) =	None				
Maximum # of iterations =	2000				
Copy Fit Settings to all Fits					
--> FIT and plot 1D <--					
Plot 1D		Plot 2D			
Plot residuals and chi-squared					
Fit parameter =	A	> Plot parameter			
Save current fits as =	'save'				
Store as problematic profile					

Clicking the '→ FIT and plot 1D ←' button will perform the fit on the density profile and plot the results. Using the left and right arrows lets you flick through the different time slices. An example of the fit is shown below for a single time slice.



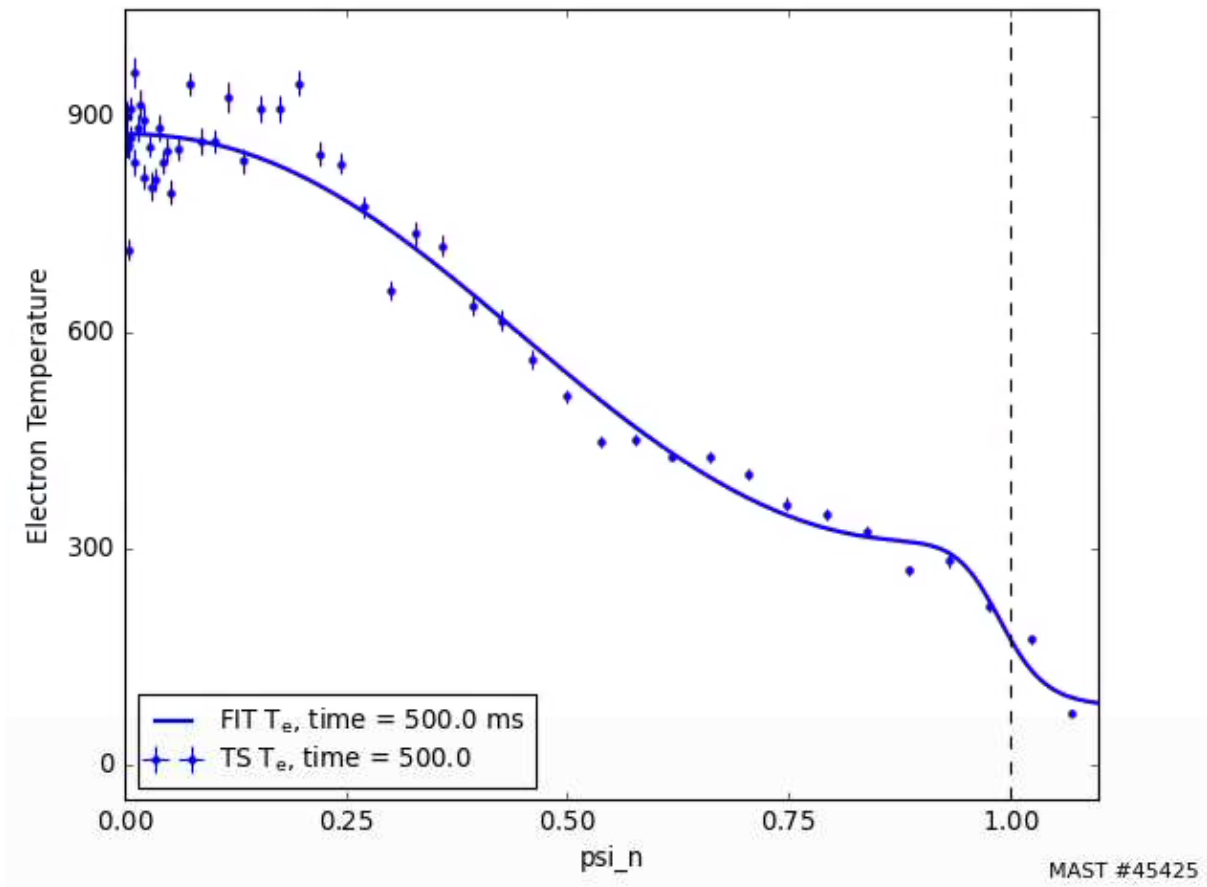
### Electron temperature fit defaults

For the electron temperature, we use the following settings.

Clicking the '→ FIT and plot 1D ←' button will perform the fit on the density profile and plot the results. An example of the fit is shown below for a single time slice.

n_e	omega_tor_12C6	T_12C6	T_e	V_tor_12C6
Fit times = <input type="text" value="0"/>				
Transform = <input type="text" value="None"/>				
Symmetry = <input type="text" value="Even"/>				
<input type="checkbox"/> Uniform radial resampling of data prior fitting				
Fitting method = <input type="text" value="Core polynomial &amp; edge exponential (mtanh) blend"/>				
Core polynomial order = <input type="text" value="3"/>				
Min pedestal width = <input type="text" value="0.001"/>				
Max pedestal width = <input type="text" value="0.05"/>				
Min core-pedestal blend width = <input type="text" value="0.01"/>				
MTANH sym. guess = <input type="text" value="0.98"/>				
MTANH sym. min = <input type="text" value="0.96"/>				
MTANH sym. max = <input type="text" value="1.0"/>				
<input checked="" type="checkbox"/> Constrain fit to have zero on-axis gradient				
On-axis value constraint (eV) = <input type="text" value="None"/>				
<input checked="" type="checkbox"/> Fit edge (x > 1) values				
Edge value constraint (eV) = <input type="text" value="None"/>				
Maximum # of iterations = <input type="text" value="2000"/>				
Copy Fit Settings to all Fits				
--> FIT and plot 1D <--				
Plot 1D		?	Plot 2D	
Plot residuals and chi-squared				
Fit parameter = <input type="text" value="A"/>			> Plot parameter	
Save current fits as = <input type="text" value="'save'"/>				
Store as problematic profile				

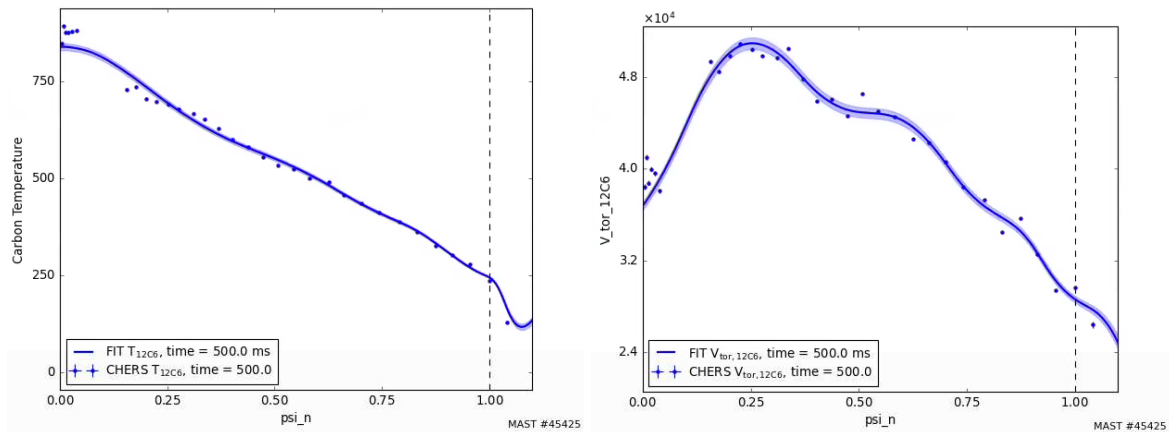




**Users interested in pedestal widths and gradients should contact the Thomson scattering diagnostic RO for further analysis. This OMFIT analysis tool does not currently provide a means for instrument function deconvolution.**

Ion temperature and velocity fit defaults

The ion temperature can generally be fitted using the default Radial basis functions fitting method. An example of the fit is shown below using 10 radial basis functions.



## Postfit derived quantities

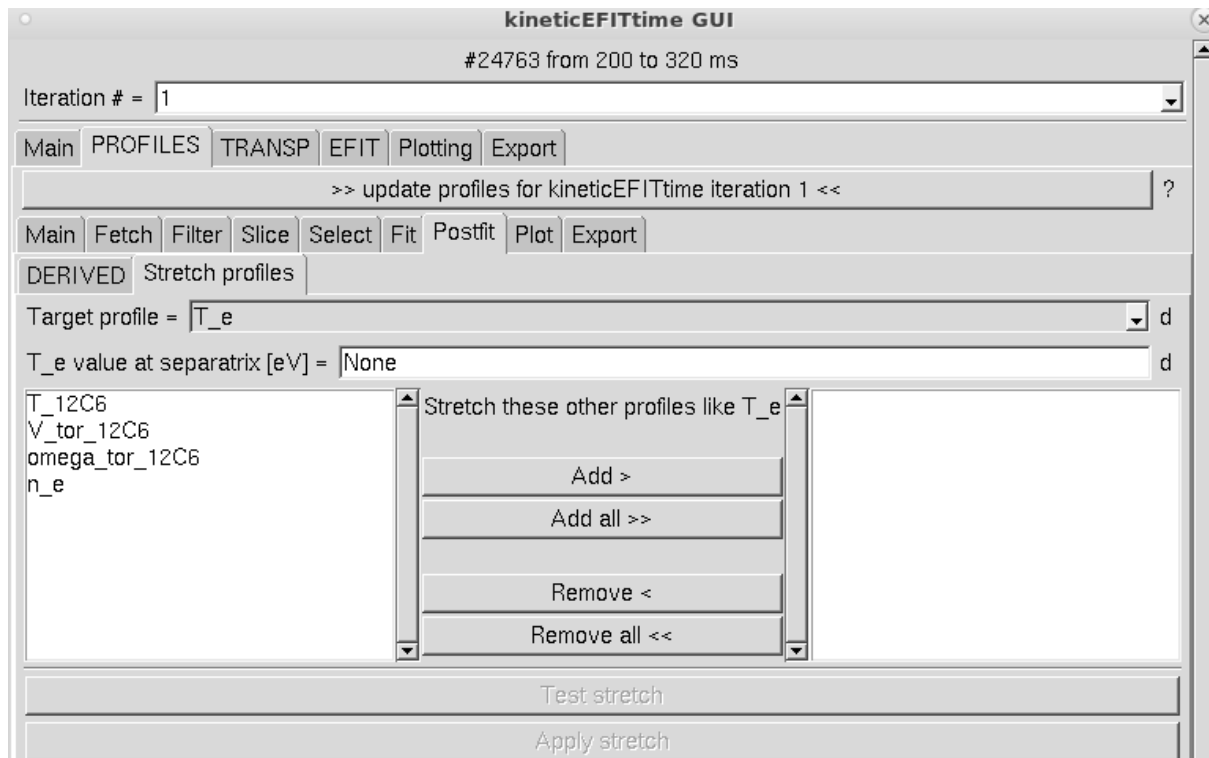
Once we have fitted the profiles, we can now calculate the derived quantities from the fits by going to the Postfit tab. For example, if we are interested in the pedestal, we can calculate the width and height by selecting 'Pedestal width and height'. Furthermore, we can calculate the scale lengths by selecting 'Gradients and scale lengths'. Lastly, since we currently don't have a regular diagnostic output for Zeff, we can assign a DZeff based on different impurities. For MAST, the impurity options are carbon and helium.

The derived Zeff will simply produce a flat Zeff profile and an associated impurity density profile based on the electron density profile fits. In this example, we set the carbon delta Zeff to 0.5 which gives a total Zeff of 1.5. Now we click on the 'Calculate derived quantities' button.

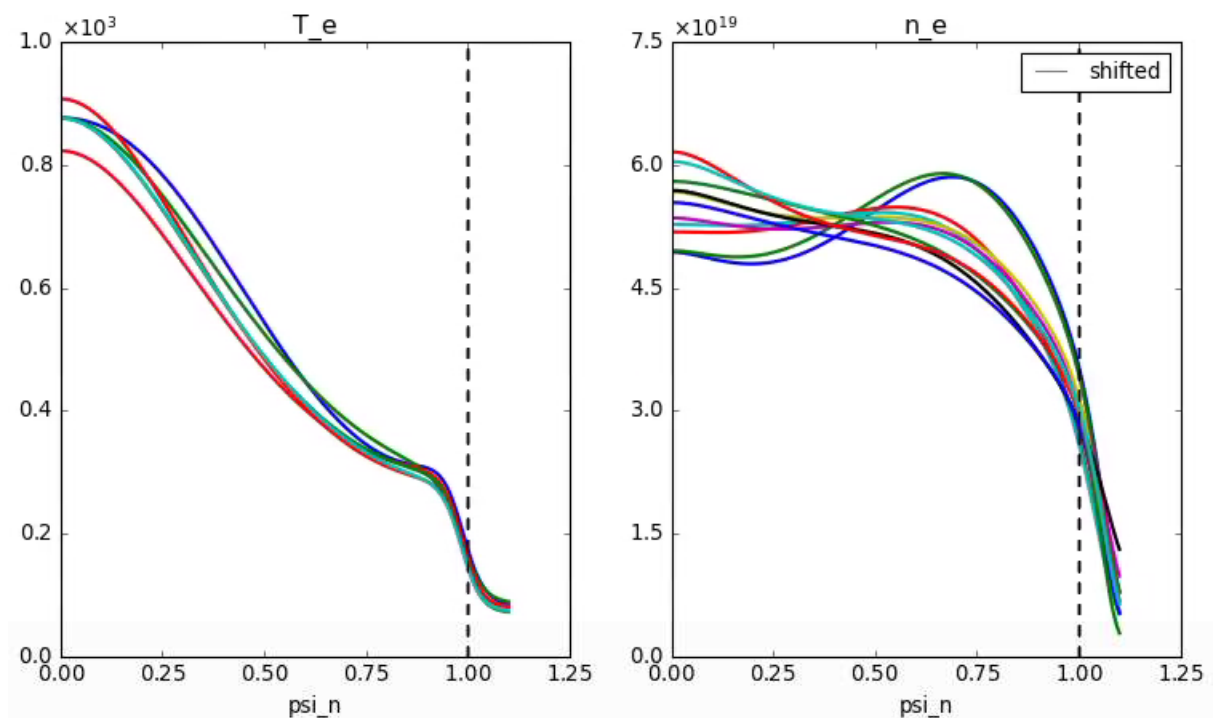
## Stretching profiles

If we are not happy with the position of the separatrix, for example if the equilibrium boundary is not accurate, then we can correct this based on an input separatrix  $T_e$  by going to the Stretch profiles tab.

In this tab, we select the target profile as  $T_e$ . This produces the options as shown below.



Now we can set the separatrix  $T_e$  as 40 eV by updating the 'T\_e value at separatrix [eV]' entry box. Underneath this, we can specify what profiles are to be stretched by introducing this constraint. We can click the  $n_e$  profile and then click on the 'Add >' button. Clicking on the 'Test stretch' button will produce the graphs as shown below.



If you're happy with the stretched profiles, we can then click on the 'Apply stretch' button.

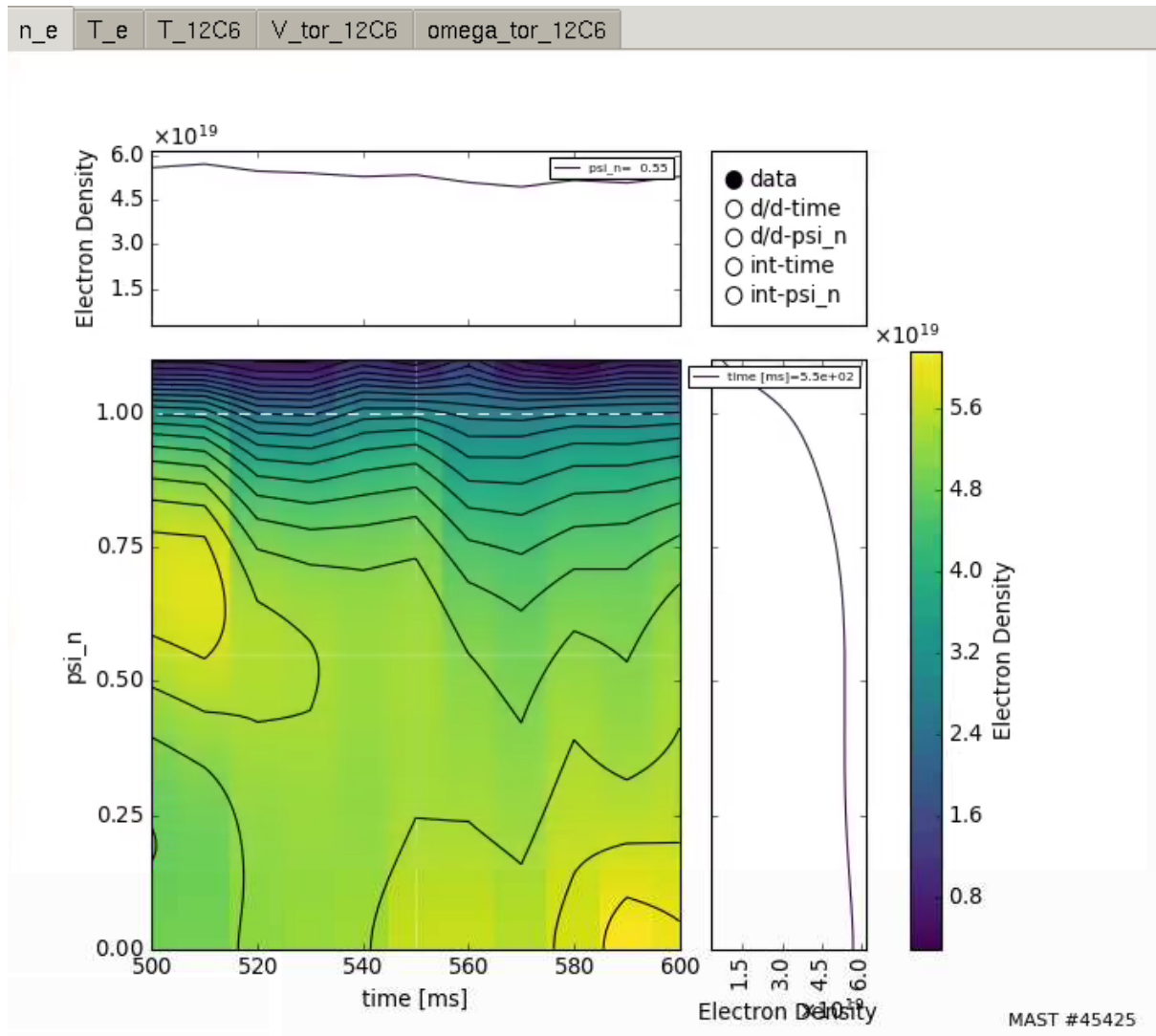
## Storing data

Once you are finished with the profile fitting, you can plot the fits and raw data in the Plot tab. However, note that if you have applied the stretch then this may cause an offset in the fits in comparison to the raw data.

Finally, to move the data into the correct location in the OMFIT tree, i.e. kineticEFITtime -> OUTPUTS -> iteration\_1 you must now click on the '>> update profiles for kineticEFITtime iteration 1<<' button near the top of the GUI.

## Plotting profiles

Moving over to the plot tab, we can plot the full 2D profiles of all the data by clicking on "Plot 2D interactive"



## Running TRANSP

The next tab to move to is called TRANSP. On this tab, to begin with, there is only the option to enter your TRANSP Run id. In this box, enter your run id, for example for Q01 type '1701'. Next, click the 'Setup TRANSP for kineticEFITtime' button. This will automatically run through the steps of generating a default TRANSP namelist and writing the equilibrium, profile and device-specific UFILES. The GUI should now look like the familiar stand-alone TRANSP GUI that some users may already be familiar with.

Setup TRANSP for kineticEFITtime

TRANSP Run = '1701'

TRANSP id: 45425Q01

Output Type = CDF d

TRANSP monitor website

☐ Did tr\_cleanup
 C ?

OMFIT['kineticEFITtime']['TRANSP']['GUIS']['RUNgui']

Main

Input Namelist

View/Modify UFILES

UFILE Options

TRANSP Version = Official Public C

Form Inputs

Look at Runing Job

RPLOT

Load Existing Run

Scan

Design Shot

Cleanup Run

1. TRANSP Namelist

Load TRANSP Namelist

2. Equilibrium Selection

OMFIT Equilibrium Module = [EFIT++] -- OMFIT['kineticEFITtime']['EFIT++']
Unlink

Load Equilibrium UFILES

3. Plasma Profiles Selection

Profiles Module = [OMFITprofiles] -- OMFIT['kineticEFITtime']['OMFITprofiles']
Unlink

Load Profile UFILES

4. MAST Specific Inputs

Load MAST-Specific UFILES and Settings

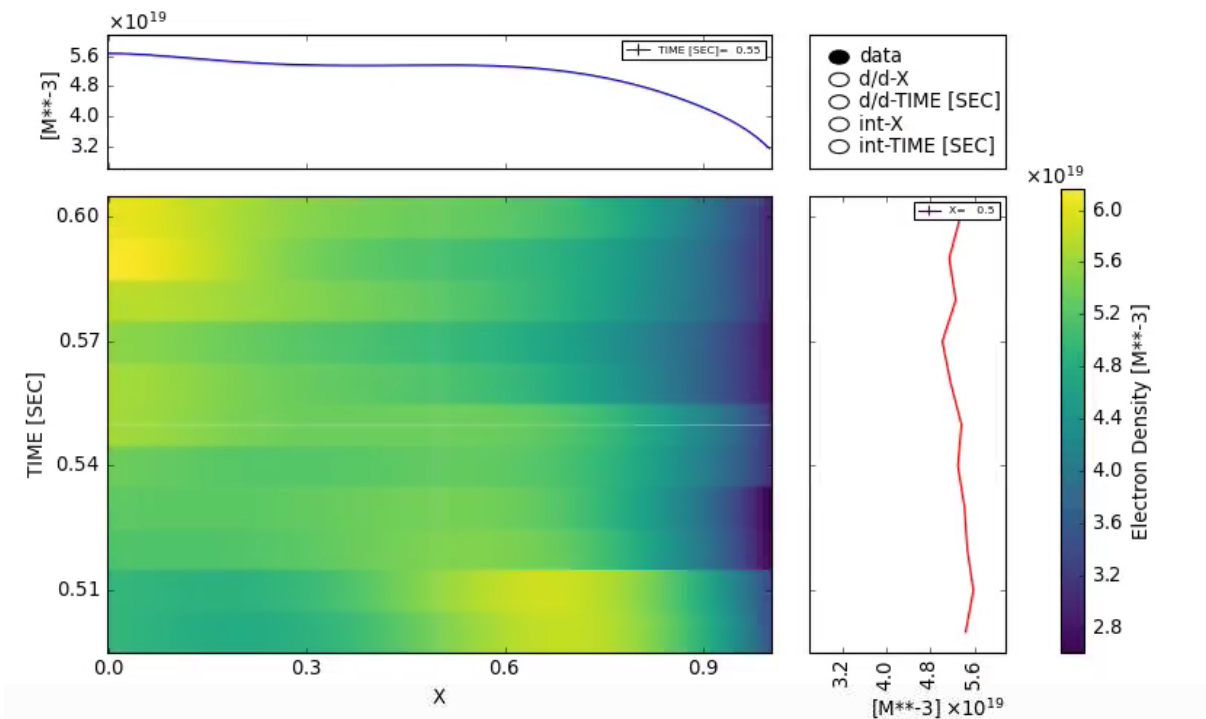
5. Selection of TRANSP Mode

TRANSP Mode = kineticEFIT C ?

Set TRANSP Mode

When using the TRANSP module standalone, the user would have to go through pressing buttons '1. Load TRANSP Namelist', '2. Load Equilibrium UFILES', '3. Load Profile UFILES', and '4. Load Device-Specific UFILES and Settings'. However, in clicking the 'Setup TRANSP for kineticEFITtime' button, these four steps have already been carried out.

To view the UFILES, navigate in the tree menu to kineticEFITtime -> TRANSP -> UFILES. Double clicking on any UFILE will produce a plot of the parameters. For example, when double clicking on the NEL UFILE (which is the electron density) we get the plot below.



At this stage, the user can modify the namelist settings for their own study. The default setup is based on the kineticEFIT TRANSP mode which is designed to provide a fast estimate of the total pressure profile to then feed back into the EFIT++ calculation.

In this example, we only make changes in the Equilibrium section and Anomalous fast ion diffusion sections. **Please skip other sections if you do not need further information on each namelist tab.**

## Time settings

The main features of interest in this section are the entries for setting the start and end times of the simulation. They will have been defaulted to the values set in the 'Times' entry at the top of the GUI window but if you want to change the simulation time window after reading in the namelist you need to use these entries.

This section also contains tabs allowing the user to set the time resolution of input and output data, the time steps used when solving the heat and particle balance equations, the time steps used by the TEQ equilibrium solver if it is being used, and the timesteps used by the heating codes. Note also that the neutral beam fidelity settings (discussed later in the [heating section](#)) will set a value for the beam time step. In this tutorial we leave this section as set by the defaults.

OMFIT['kineticEFITtime']['TRANSP']['GUI']['TIMEgui']

TRANSP Time Range

Start time (s) = 0.49 d ?

End time (s) = 0.61 d ?

Input Output Balance Equilibrium Heating

Input Time Resolution Settings

Scalar Time Resolution (s) = 0.01 C ?

Profile Time Resolution (s) = 0.01 C ?

## Radius

These settings allow you to set the number of radial zones used by the main TRANSP code and for the neutral beam distribution function. This can be left as default.

## Equilibrium

Under the '*Geometry*' tab one can choose whether to read in a pre-existing full EFIT equilibrium or whether to use the TEQ inverse solver in TRANSP to solve for the equilibrium using the plasma boundary and q profile. In both cases the EFIT equilibrium still needs to be read in; however, in the latter case only the q profile and boundary information are used.

There are pros and cons to both cases. If one is using for instance TS or CX data which needs to be mapped onto a flux coordinate prior to input (as it is not taken along the midplane), this mapping will usually be performed using the EFIT equilibrium so using a different equilibrium for the TRANSP run is not entirely consistent. However, using TEQ allows the fast particle pressure from TRANSP to be included in the equilibrium calculation. It is for the user to decide which is best for their run.

Under the magnetics tab is a checkbox to use current diffusion. If this is selected, TRANSP will solve the poloidal field diffusion equation to determine the evolution of the q profile. In this case it is necessary to select a resistivity model and a bootstrap model. This should only be selected when using TEQ, an input EFIT equilibrium will already have a consistent q profile. **In this example, deselect this check box.**



## Plasma

This section allows the user to configure the composition of the plasma, the fusion product model settings and whether to take the ion temperature from data or set it as a given multiple of the electron temperature. The composition section allows the user to set the bulk plasma and impurity species as well as a value of  $Z$  effective to use if input data is not being used. Currently, there is no  $Z_{\text{eff}}$  diagnostic for MAST and therefore a constant  $Z_{\text{eff}}$  value is the default.

## Heating

This section is the most complicated and is where the user sets up the beam information. The information on the beams must either be read from the UDA system or manually entered by the user. The process for doing so is detailed below.

## Neutral Beams

The screenshot shows a software interface for configuring neutral beams. At the top, there are tabs: 'Main', 'Input Namelist', 'View/Modify UFILES', and 'UFILE Options'. Below these are sub-tabs: 'Time', 'Radius', 'Equilibrium', 'Plasma', 'Heating', 'Diagnostics', 'ACFILE', 'Sawteeth', and 'Pellets'. The 'Heating' sub-tab is active, showing a section for 'NBI' (Neutral Beam Injection) with sub-tabs 'ECH', 'ICRF', and 'LSC'. The 'LSC' sub-tab is selected, displaying the 'OMFIT[\'kineticEFITime\'][\'TRANSP\'][\'GUIS\'][\'NUBEAMgui\']' configuration panel. This panel includes a dropdown menu 'How to drive NUBEAM =' set to 'Shot', a text input 'Shot for MDSplus =' with the value '45425', a 'Create NUBEAM Namelist and UFILES' button, a checked checkbox 'Enable Neutral Beams', a dropdown 'Settings GUI =' set to 'Pre-defined Fidelity', a dropdown 'Fidelity =' set to 'Low', a 'Set NUBEAM fidelity' button, a text label 'Using UFILE for EINJ, PINJ, FFULL, FHALF, TBOA, TBOFFA', a dropdown 'Beamline =' set to '1', text inputs for 'A of injected species [amu] =' (2.0) and 'Z of injected species =' (1.0), a checked checkbox 'Beam injects Co-current', a 'Plot NUBEAM' button, a 'Fast ion diffusion' section with an unchecked 'Use Anomalous diffusion' checkbox, and a 'Fishbone loss' section with an unchecked 'Use Fishbone loss' checkbox. Each configuration item has a 'd' or 'C' icon to its right, likely for documentation or help.

The neutral beam GUI is shown above. Information on the configuration of the beams (powers, energies, full and half energy fractions) needs to be provided and this can be done either by reading the required information from the UDA system or by manually entering it. This choice is made using the drop-down box labelled '*How to drive NUBEAM*'. This has two options: '*shot*' which indicates that the information will be read from the UDA or '*Namelist Only*', which indicates that the information will be manually set. Note that the beam geometry is currently hard-coded; future updates will provide a GUI to enter this information. Once the desired option has been chosen pressing the button labelled **Create NUBEAM Namelist and UFILES** will either automatically read the information and create the required Ufiles in the former case (the created Ufile has extension .NB2) or bring up a GUI allowing the user to manually configure the beams in the latter case.

The only other thing to check here is the NUBEAM Fidelity. This sets several namelist switches which govern how accurate the beam simulation is. The options are as follows:

#### Low Fidelity

Intended for control room analysis for time to solution, and kineticEFITs

Monte-carlo particles NPTCLS=2000

Deposition tracks NDEP0 = 500

Beam timestep DTBEAM = 0.01

Numerical acceleration GOOCON = 5.0

MPI - 4 cores for 250 particles / core

#### Medium Fidelity

Intended for standard particle and power balance calculations and reasonable torque density

Monte-carlo particles NPTCLS=32000

Deposition tracks NDEP0 = 5000

Beam timestep DTBEAM = 0.005

Numerical acceleration GOOCON = 10.0

MPI - 8 cores for 4,000 particles / core

#### High Fidelity

Intended for detailed momentum transport and fast-ion distribution function evolution

Monte-carlo particles NPTCLS=256000

Deposition tracks NDEP0 = 5000

Beam timestep DTBEAM = 0.001

Numerical acceleration GOOCON = 20.0

MPI - 64 cores for 4,000 particles / core

#### Anomalous fast ion diffusion

Near the bottom of the GUI, there is a checkbox labelled 'Use Anomalous diffusion'. **For this tutorial, we will switch this on.** In both the minimum and maximum entry boxes, enter 3.5E4. This will set a constant diffusion rate of 3.5 m<sup>2</sup>/s for the fast ions.

## ACFILE

Clicking the checkbox here allows the user to specify if they wish the TRANSP run to output the beam fast ion distribution or neutral density. Note that the ACFILE does not copy into the Archive folder - it will only remain in the results directory.

## Sawteeth

This section allows the user to either set a sawtooth extrapolation window such that TRANSP will not use data within a given window of a sawtooth event but will extrapolate from outside the sawtooth window to the sawtooth time or to model sawteeth based on the q profile.

## Pellets

This section allows the user to either set pellet times and extrapolation windows such that TRANSP will not use data within a given window of a pellet but will extrapolate from outside the pellet window to the pellet time or set times to model injected pellets.

## Submitting the TRANSP run

If you have completed the above steps then you should have all the data needed to submit a TRANSP run. To do this, go back to the Main tab next to the PROFILES tab, and deselect '1) run profiles' and '3) run equilibrium'. Now click the button called 'Run Transport'.

MAST #45425 from 500 to 600 ms

Iteration # = 1 d

Main TRANSP Plotting Export

☐ 1) run profiles d

☒ 2) run transport d

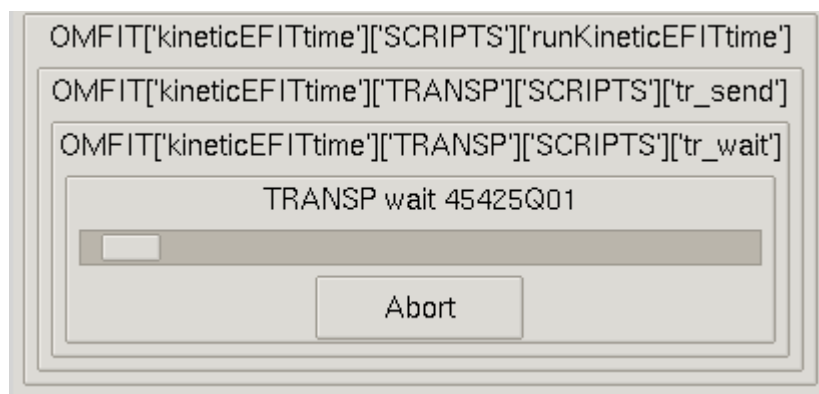
Transport Module = TRANSP C

☐ 3) run equilibrium d

Run Transport

This now goes through the process of running the NT codes called tr\_start, tr\_send and tr\_fetch. If the run was successfully submitted to the FusionGRID at PPPL, you can view the status of it on the website: [https://w3.pppl.gov/transp/transpgrid\\_monitor](https://w3.pppl.gov/transp/transpgrid_monitor).

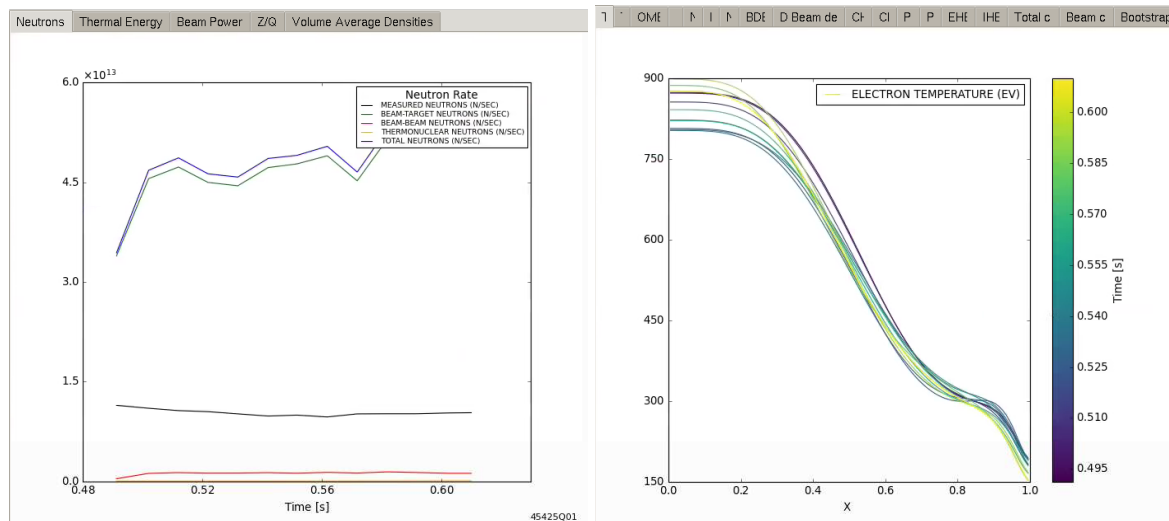
Omfrit also employs a waiting function, which will periodically query the database to check if the run is completed. On the screen this looks like the image shown below.



In this example, the run should take around 5 minutes to complete. After waiting this time, the run should be automatically fetched from the grid and placed into the TRANSP tree in kineticEFITtime -> OUTPUTS -> iteration\_1 -> transport\_module -> OUTPUTS -> TRANSP\_OUTPUT.

## Analysing the TRANSP run

Once a TRANSP run has been fetched, it can be viewed within OMFIT. For an overview of the outputs, navigate to kineticEFITtime -> TRANSP -> PLOTS -> MAST and then double click on 'summary'. This will produce a number of plots as shown below.



The main point to notice here is the agreement between the total calculated neutrons (in blue) and the measured neutrons in black. On MAST, the neutron rates are effectively all driven by beam-target collisions, with a smaller contribution from beam-beam collisions and a negligible contribution from thermonuclear neutrons. The calculated neutrons over-predict the measurement by a factor of 4, though the fission chamber measurements are still to be calibrated.

## Pressure constrained EFIT++

**NOT AVAILABLE FOR MAST-U - Below is for MAST 24763**

With a reasonable level of agreement, the next step is re-running with the pressure constrained equilibrium.

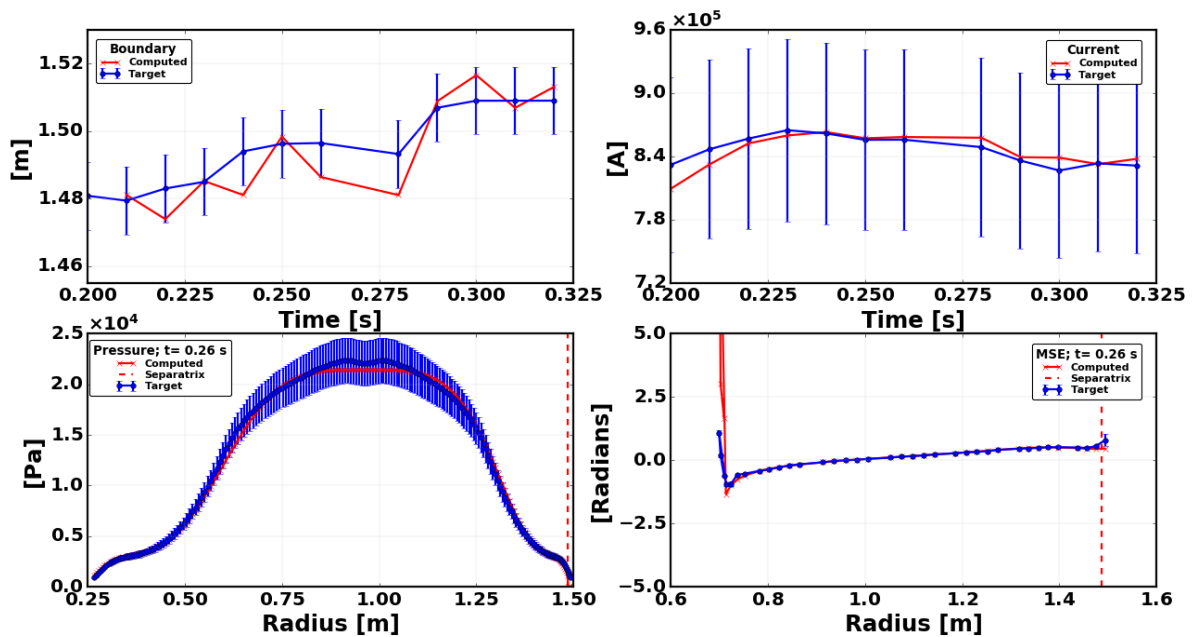
Now that we have our TRANSP output, we can use the pressure profile to constrain our EFIT++. In the Main tab, deselect the '1) run transport checkbox' and select '3) run equilibrium'. Next, move to the EFIT tab and you will notice in the TRANSP tab near the bottom of the GUI that the 'Use TRANSP pressure constraints' checkbox is now highlighted. Note the difference in font color which means that the user cannot deselect this tab.

The TRANSP seq has been automatically updated to the sequence number specified in the previous section, and the GUI shows 'TRANSP CDF file loaded: 24763S01.CDF'. Underneath, there is a checkbox 'Mirror LFS/HFS pressure profile'. TRANSP returns the

pressure profile as a function of  $\rho$ , and therefore EFIT++ can either map this to the LFS or to both the LFS and HFS. In this example, we select 'Mirror LFSHFS pressure profile' .

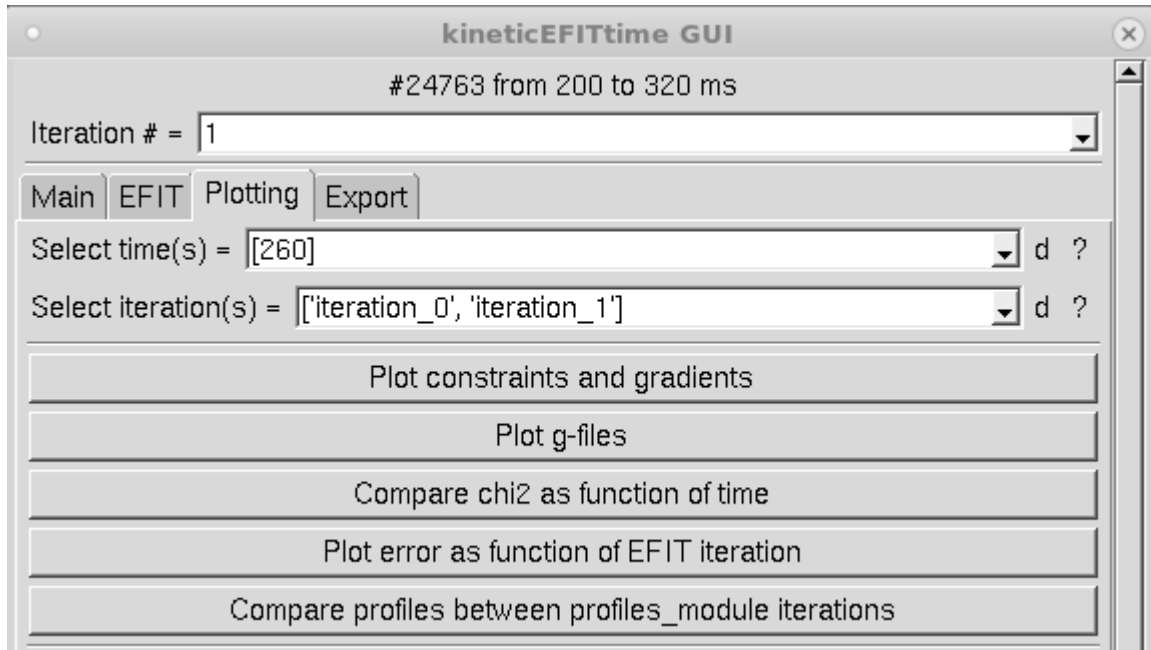
In the Pressure tab, we can now change 'Add prange' to '0 2.0' to constrain the pressure profile over the entire radial profile. Furthermore, in this example we will select 'Allow finite  $p'$  and  $f\bar{f}'$  at boundary'.

Now go back to the Main tab and click the 'Run Equilibrium' button. We can again view the comparison between the EFIT++ output and input constraints by entering in the Command box: `root['kineticEFITtime']['EFIT++']['OUTPUTS']['CDF'].plot(time=0.26)`.

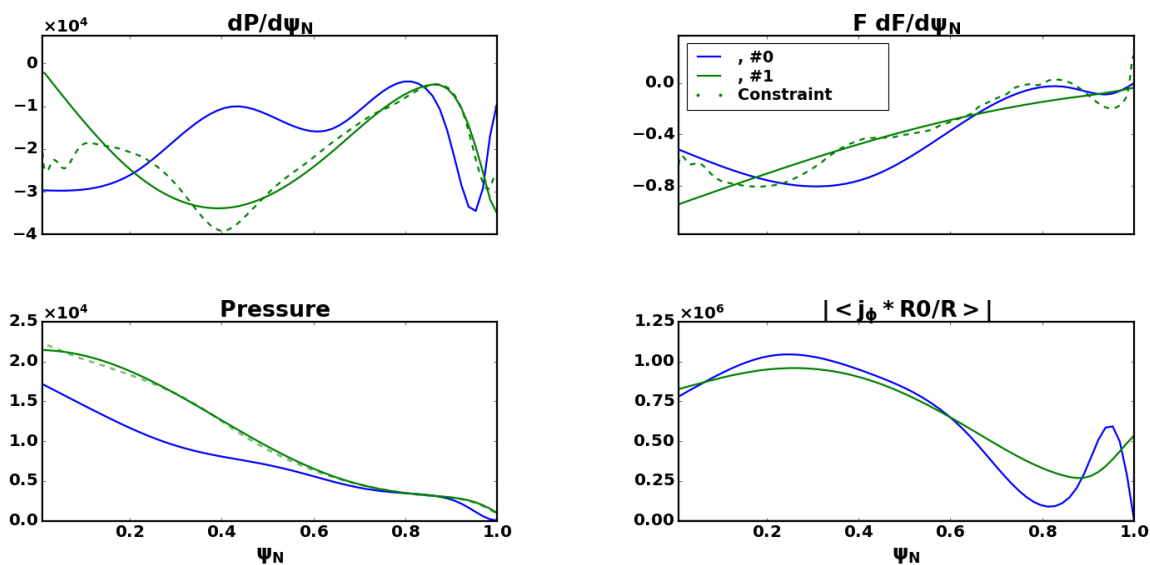


## Comparing to the previous iteration

A handy tool in the kineticEFITtime GUI is the feature to compare equilibrium between iterations. To use this, go to the Plotting tab in the main GUI.



In the Select time(s) dropdown box, select the time slice (s) you'd like to assess. In this case, let's select 260 ms. In the Select iteration(s) dropdown box, select ['iteration\_0', 'iteration\_1']. Then select 'Plot constraints and gradients'. This produces 4 plots as shown below.



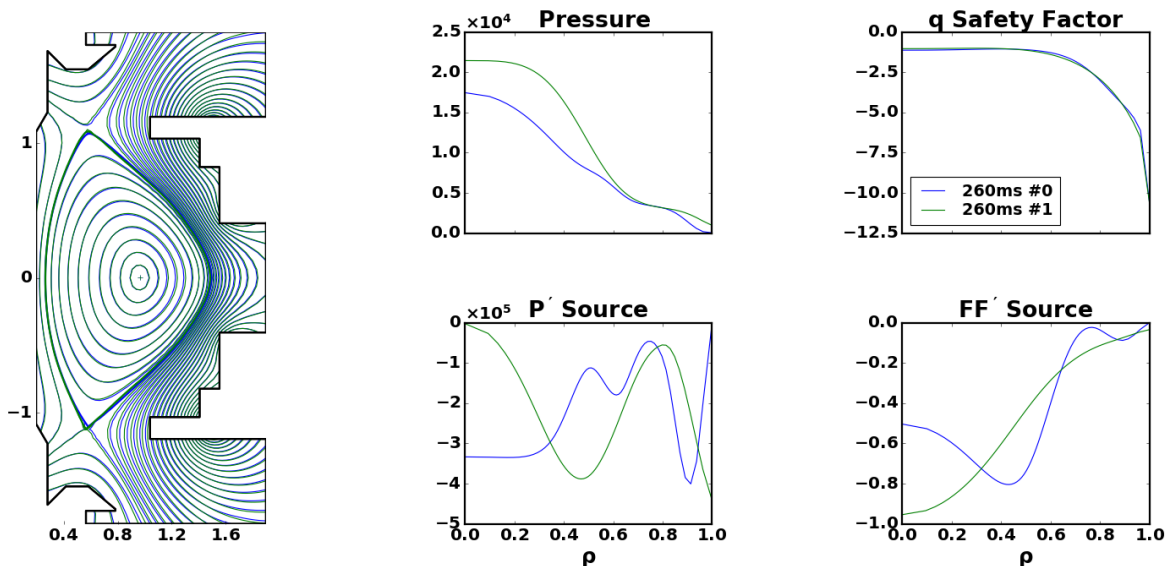
MAST #24763

The bottom left graph shows the difference between the pressure profiles in the first and second iteration. The pressure in the core and near the edge is moderately higher than the EFIT++ solution using only the Thomson scattering constraint. In the edge, this



could be attributed to the inclusion of the Ti measurement. For the core, if we recall, the pressure profile was only constraining the edge of the plasma and therefore EFIT++ was unconstrained in the core with regards to the pressure profile.

We can also view the difference between the gEQDSK files for the two iterations by clicking the 'Plot g-files' button.



On the left plot, the blue and green contours show the difference in the magnetic flux surfaces between the two iterations. In this example, there is a minor difference to the X-point location which has a knock on effect on the strike-point position. However, the plasma centre and shape does not look to have been significantly altered.

## Iteration: 2

### TRANSP analysis run

Now we have, in principle, an equilibrium that we are happy to use to run a more detailed TRANSP simulation. That is, we may now use more test particles in the NUBEAM calculation and increase the time resolution. However, before we do this, we must refit the profiles using the new equilibrium.

As we have already gone through the settings in each of the subtabs in the PROFILES we can now simply deselect the '2) run transport' and '3) run equilibrium' options and click on 'Run Profiles' in the Main tab. This will now go through each of the steps, e.g.

mapping, filtering, slicing, fitting and post-fitting analysis. Note, it is always good practice to double check the saved settings in each of the PROFILE tabs just to ensure it is doing what you expect. Furthermore, The Stretch profiles feature will need to be done manually.

Once you are happy with the new fits, you can now select '2) run transport' in the Main tab and then move to the TRANSP tab. Update your TRANSP run id to a new run and then click 'Setup TRANSP for kineticEFITtime'. Now, you can automatically increase the namelist settings for an analysis run by choosing 'Analysis' in the TRANSP mode drop down box. Once it is set, click 'Set TRANSP mode'.

Now you are ready to submit your TRANSP run.

## **Saving your work and trading runs**

To save your work click on

File->Save Project As

You can then give the project a name (I would advise using the run ID of the TRANSP run) and leave some useful comments about it before saving it. Note that the entire OMFIT environment is saved here, all your Ufiles are saved, the namelist is saved and the settings you currently have are saved. When you reopen the project everything will look just as it did when you saved it.

If you want to copy someone else's run then it is simply a question of copying the run from their projects directory to your projects directory and renaming it something appropriate. If you followed the [First Time Settings](#) at the bottom of this document your projects directory will be:

```
/common/transp_shared/Data/work/<user>/OMFIT/projects
```

Once copied over, the project can be opened in an OMFIT session as if it was your own and will look exactly as it did for the person that created it when they saved the project.

You are then free to modify the sequence number (to use your own letter identifier) and resubmit the run with any changes you wish to make.

## **Useful Advice**

- If you are not sure what something does, see if it has a **?** mark next to it on the right. If so you can click on this for information about the entry.
- If you can't find a GUI after pressing a button, it may have appeared behind the main TRANSP GUI, try moving this.
- The first thing that must happen when preparing a run is to read in a namelist. After that you can read the data in any order you like although it is suggested you follow the order laid out. Once all data you want is read in, you can submit the run.
- If you wish to manually edit the namelist this is possible. In the **INPUTS** sub-tree, right click on the TRANSP entry (the namelist) and select '*Open in Editor*'. This will open the namelist for editing. Once you have finished editing it, save it and then right-click on the TRANSP entry again and select '*Reload from File*'. It is strongly advised that any manual edits of the namelist are made immediately before submitting the run after all other preparation has been done. Otherwise there is a small possibility that a user could add an entry then use a GUI function which adds the same entry in a different location in the tree leading to the same entry existing twice in the namelist. It is also recommended that users attempt to keep any manually added entries in appropriate subsections. The subsections are structured to match the sections on the [transp.help](http://transp.help) on the PPPL website.

## Appendices

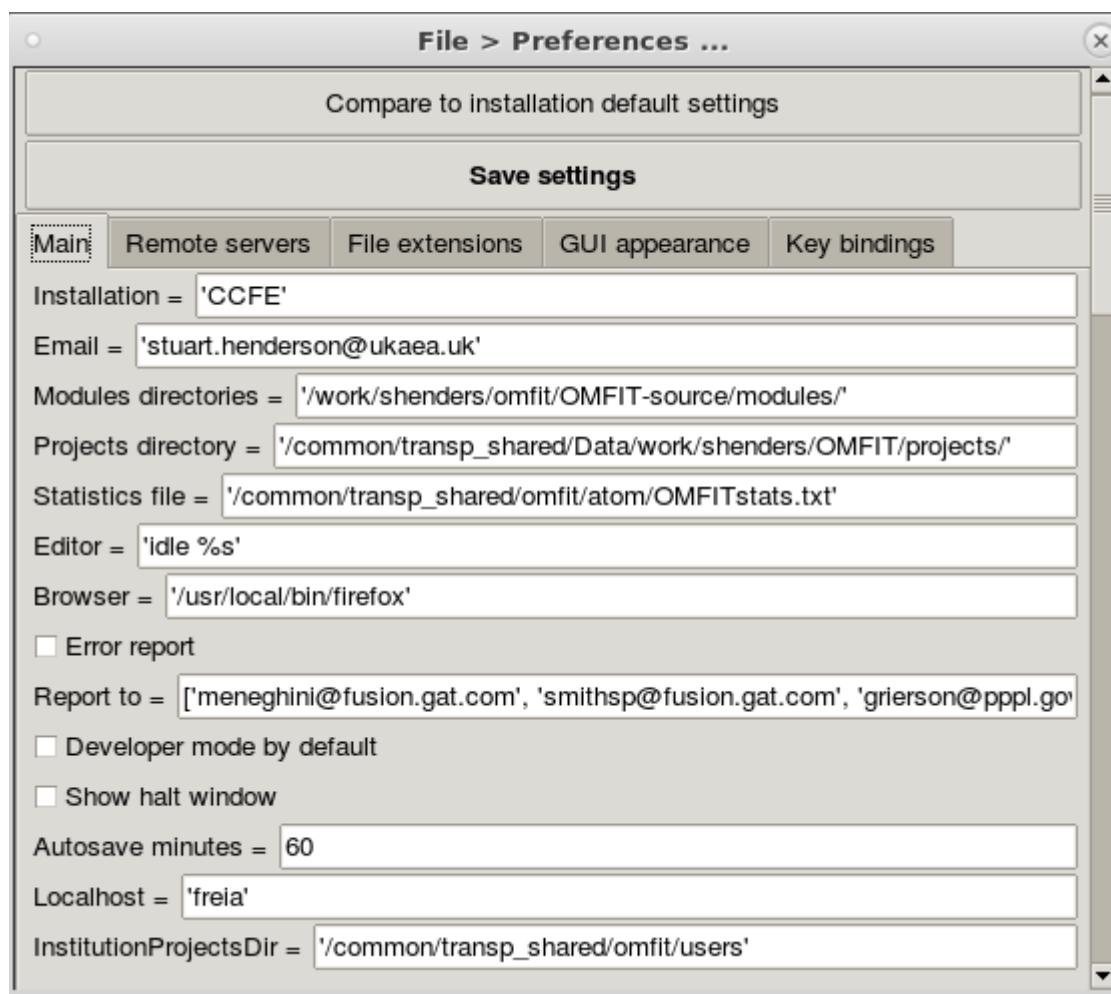
### First Time Settings

There are a few settings that you will need to configure the first time you run OMFIT.

After starting the main window perform:

File->Preferences

To bring up the user preferences window:



Under '*Email*' enter your email address so that you can be contacted about any errors etc. Change the '*Projects Directory*' entry so that it matches:

/common/transp\_shared/Data/work/<user>/OMFIT/projects

Where <user> is your CCFE username. This is where your saved OMFIT projects will be stored. Then click on the **Remote Servers tab**:

Compare to installation default settings	
Save settings	
Main	Remote servers
ASIPP username =	<input type="text"/> C
CCFE username =	<input type="text" value="'bpatel2'"/> C
CEA username =	<input type="text"/> C
EPFL username =	<input type="text"/> C
GA username =	<input type="text"/> C
GF username =	<input type="text"/> C
GITHUB username =	<input type="text"/> C
IFERC username =	<input type="text"/> C
IPPCZ username =	<input type="text"/> C
IPPMPG username =	<input type="text"/> C
ITER username =	<input type="text"/> C
ITM username =	<input type="text"/> C
KFE username =	<input type="text"/> C
MIT username =	<input type="text"/> C
NERSC username =	<input type="text"/> C
NFRI username =	<input type="text"/> C
PPPL username =	<input type="text" value="'bpatel'"/> C
RFX username =	<input type="text"/> C
SWIP username =	<input type="text"/> C
TCV username =	<input type="text"/> C
UCSD username =	<input type="text"/> C
USPBR username =	<input type="text"/> C
Default server =	<input type="text" value="'localhost'"/> d
Edit server =	<input type="text" value="freia"/> d
Server =	<input type="text" value="'bpatel2@freia023:22'"/> C
Tunnel =	<input type="text"/> C
Copy `freia` to new server =	<input type="text"/> C

Your username should already be correct but you should check that it is. Change the *'Server'* entry to '*<user>@freia<n>:22*' where *<user>* is your username on Freia and *n* is a valid node number. From above you can see that I have chosen freia023. Lastly make sure the *'Tunnel'* entry is an empty string. Then click on **Save Settings** at the top of the GUI and restart OMFIT.