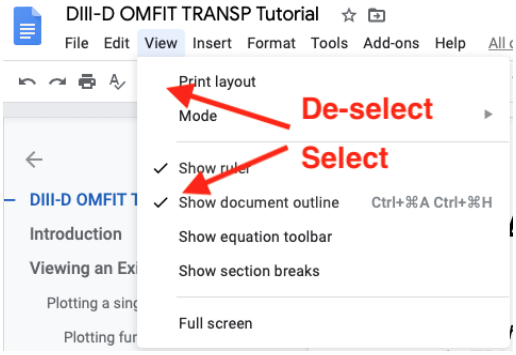
DIII-D OMFIT TRANSP Tutorial

TRANSP is a PPPL PRINCETON PLASMA PHYSICS LABORATORY code

B.A. Grierson for the OMFIT Developers

This is a help document written in tutorial fashion that guides the user through the preparation of a TRANSP run, optional selections, and visualization of the output.
The publication is here: Orchestrating TRANSP Simulations for Interpretative and Predictive Tokamak Modeling with OMFIT

To most effectively use this Google Doc do the following:



A prerequisite to using TRANSP through OMFIT is completing the first time users setup.

Introduction

TRANSP is a comprehensive, time-dependent transport code used for interpretive analysis and predictive simulations of tokamaks. Technical guides are available, as well as the online reference help page and publication on the OMFIT module:

TRANSP Homepage https://transp.pppl.gov
TRANSP Detailed Help http://w3.pppl.gov/~pshare/help/transp.htm
Publication: Orchestrating TRANSP Simulations for Interpretative and Predictive Tokamak Modeling with OMFIT

This document outlines typical use cases for TRANSP, rather than providing an overview of the equations and technical notes. In this sense, this document presents the guiding philosophy and assumptions to answer a scientific question with TRANSP and assumes a basic prerequisite knowledge of tokamak plasma physics and transport in general. Where appropriate, pro tips and examples are provided, as well as further links and reference to details and namelist values.

Instructions for getting access to run TRANSP for DIII-D are in the appendix.

Viewing an Existing TRANSP Run

This is often the best way to get familiar with TRANSP for new users.
Start OMFIT (or use these Instructions in Courseware if you don't know how)
Open the TRANSP GUI and enter the following:

Device = `'DIII-D'`
Shot = `163303`
TRANSP Run = `'A01'`

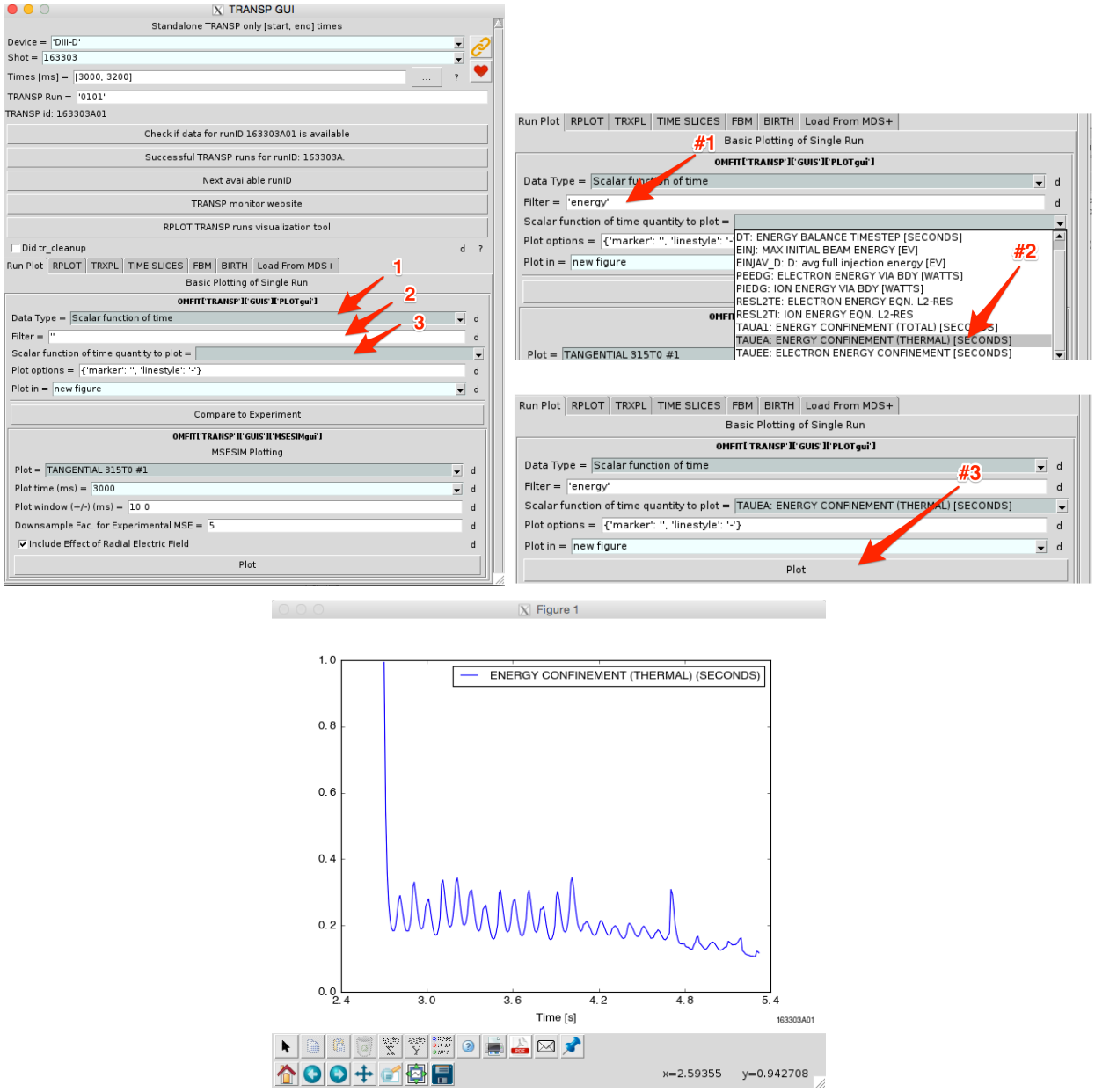
Plotting a single run and finding variables

A TRANSP run has an incredible amount of information and output, and getting familiar with the myriad of outputs variable names TRANSP provides takes a very long time. Therefore a convenient search tool is provided to explore the TRANSP run.
In the Run Plot tab there is a ComboBox for Data type

- Scalar function of time, i.e. $I_p(t)$
- Function of time and X, i.e. $T_e(x,t)$
- Multigraphs, i.e. all terms in electron energy balance (heating, loss, exchange)

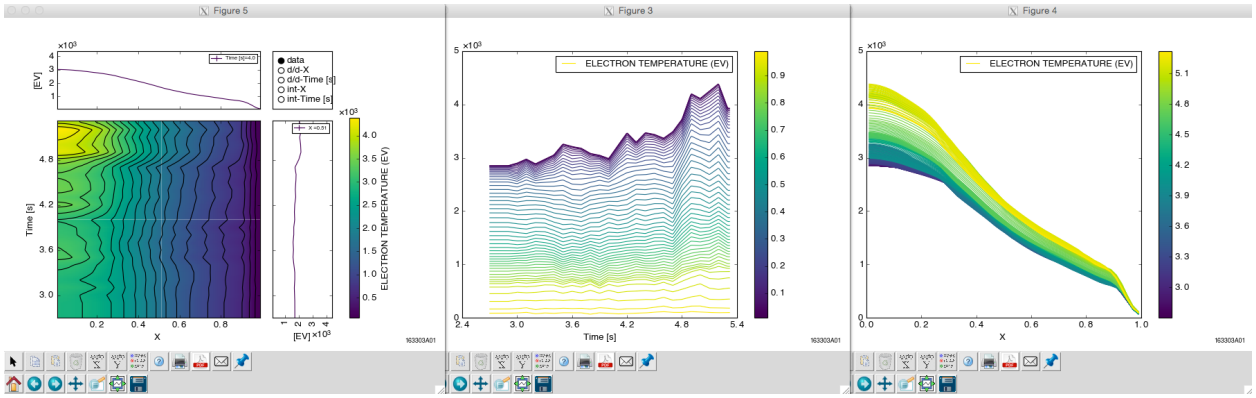
Plotting function of time

Select `Scalar function of time` in the ComboBox
In the 'filter' box type `energy` and [return]
In the 'quantity to plot' ComboBox select `TAUEA: ENERGY CONFINEMENT TIME`
Then click Plot



Plotting function of time and space

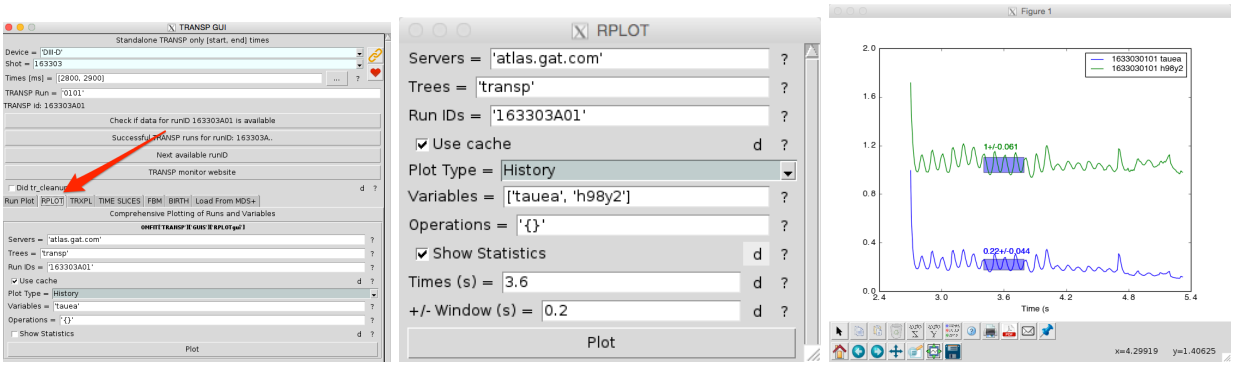
Select `Function of time and X` in the ComboBox
In the filter box type `temperature` and [return]
In the 'quantity to plot' ComboBox select `ELECTRON TEMPERATURE`
Then click Plot for an interactive plot
Select 'slice in' `X` and click Plot
Select 'slice in' `Time` and click Plot



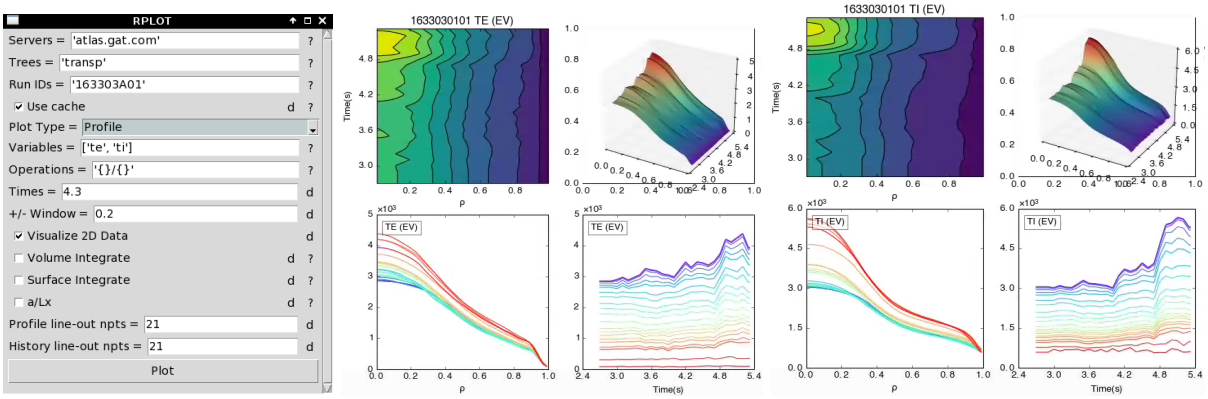
Interactive plotting, slice in X and slice in Time

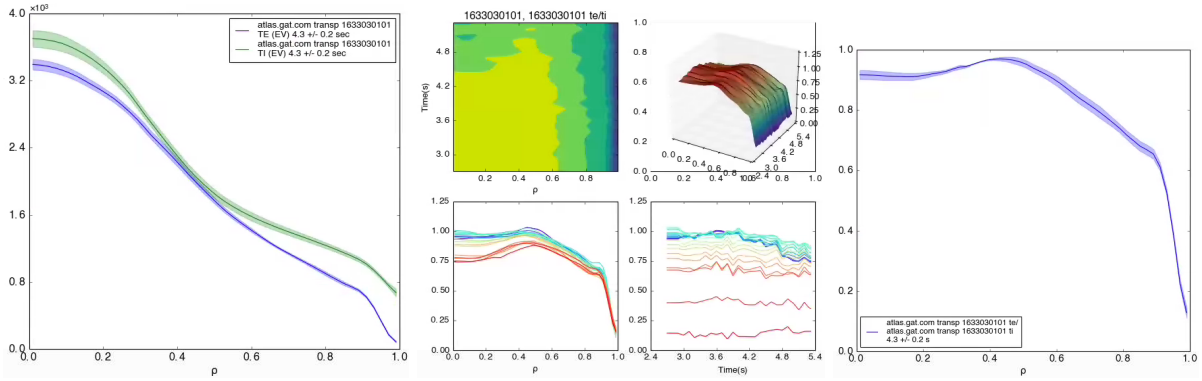
Using RPLOT for comprehensive plotting and calculations

Select the RPLOT tab. The entries will be filled out with the defaults entered previously. Here we can change the TRANSP MDSplus server, MDSplus tree and runID and variable type. Note that these can be lists, so that we can compare multiple runs, multiple machines, multiple variables, etc...

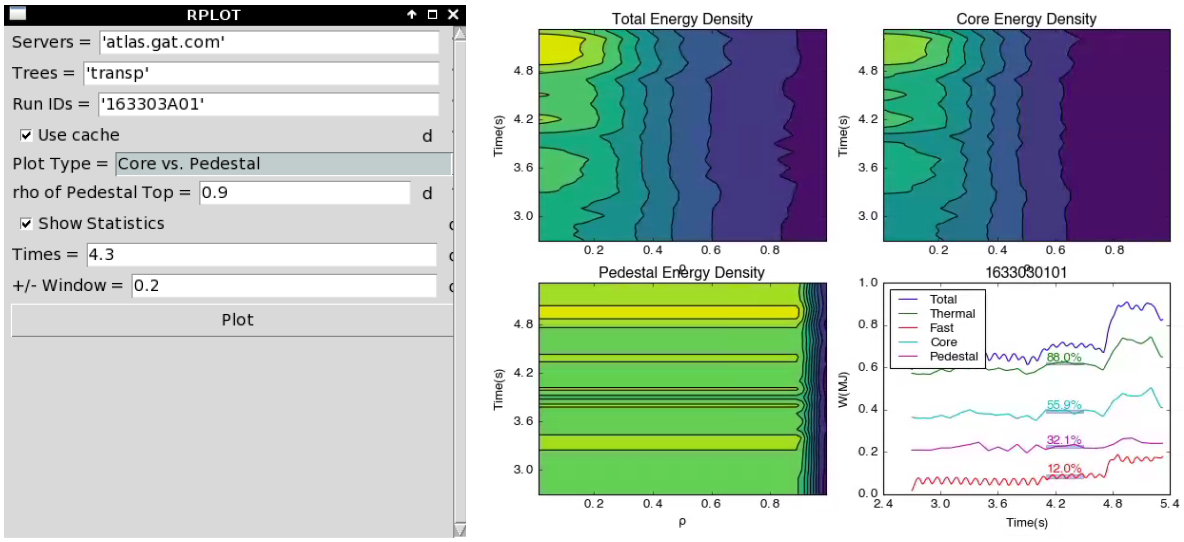


Below are a few examples of the output you can view from TRANSP:

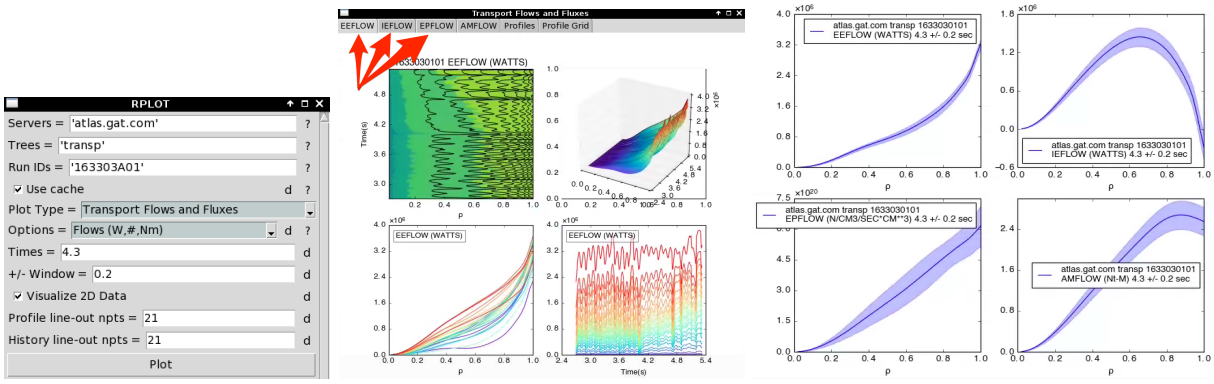




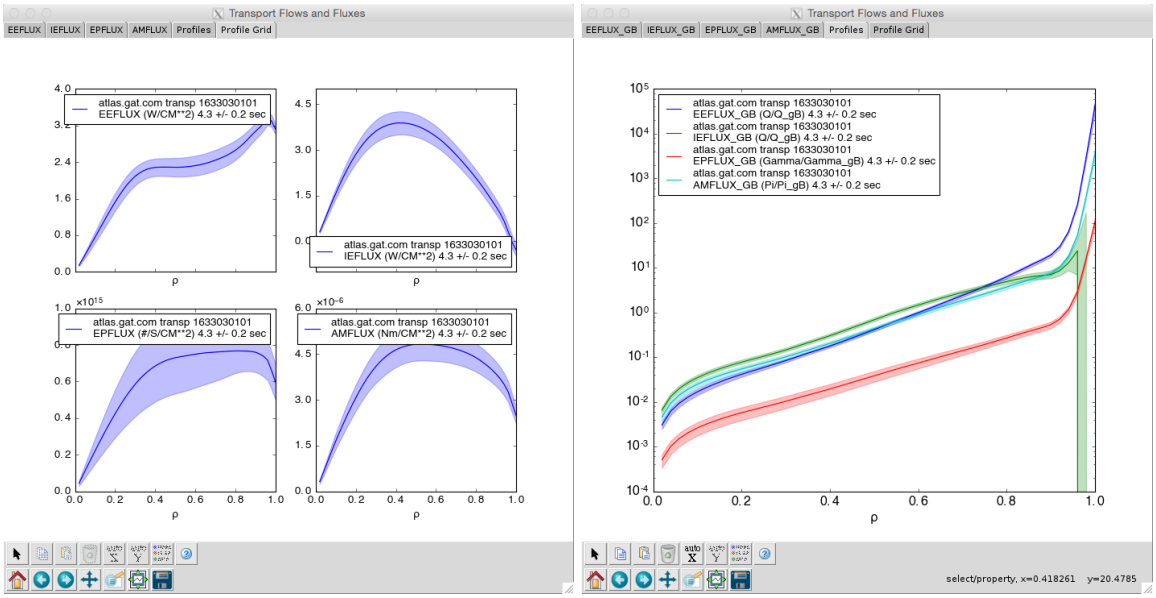
Te, Ti profiles and ratio via “operations”; all appear in a python FigureNotebook



Contributions to total stored energy; thermal vs. fast and core vs pedestal



Power/particle/momentum flows in Watts, particles and Nm

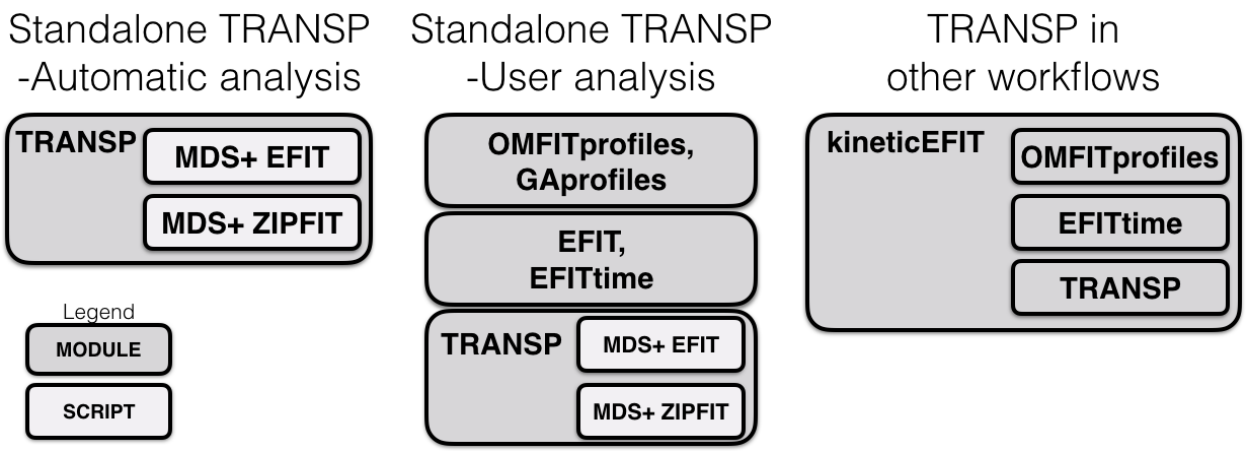


Fluxes in CGS and gyro-Bohm units taken from the [GACODE] standard.

TRANSP for Power Balance Analysis

This is the simplest type of transport analysis and there are a few requirements that come with this workflow. First the user must provide time histories and plasma profiles. Time histories include the plasma current and field, neutron rate, recycling and heating powers. Plasma profiles include the equilibrium quantities such as q, pressure and diamagnetism, as well as the fluid variables ne, Te, Ti and others. TRANSP places these inputs onto a uniform time grid and steps forward in time computing the heating power injected, time derivatives of the fluid and equilibrium profiles and arriving at the power flows. The power flows are the sources minus the sinks minus the d/dt terms and are known as the “observed” transport power densities.

The TRANSP module in OMFIT has taken the role of IDL *autotransp* tool at DIII-D with a greatly expanded capability. There are two very high level differences in the way TRANSP can collect its input data. Either EFITs come from MDSplus, or EFITs come from an OMFIT EFIT module. Similarly, profiles come from between-shot MDSplus ZIPFITs, or profiles come from an OMFIT profile module such as OMFITprofiles, GAprofiles or user ZIPFITs. You can also mix and match (i.e. MDSplus EFIT and GAprofiles). These are illustrated below.



Standalone TRANSP - Automatic Analysis

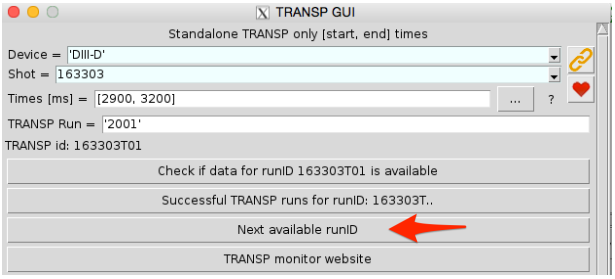
Load the TRANSP module in OMFIT and start the module

Set the device, shot, time and TRANSP run

Open the TRANSP GUI and enter the following:

Device = `'DIII-D'`
Shot = `163303`
Times = `[3000,5000]`
TRANSP Run = `'T01'`

Click the button 'Next Available RunID'



Next move on to enter a comment, such as `'Tutorial Example'`

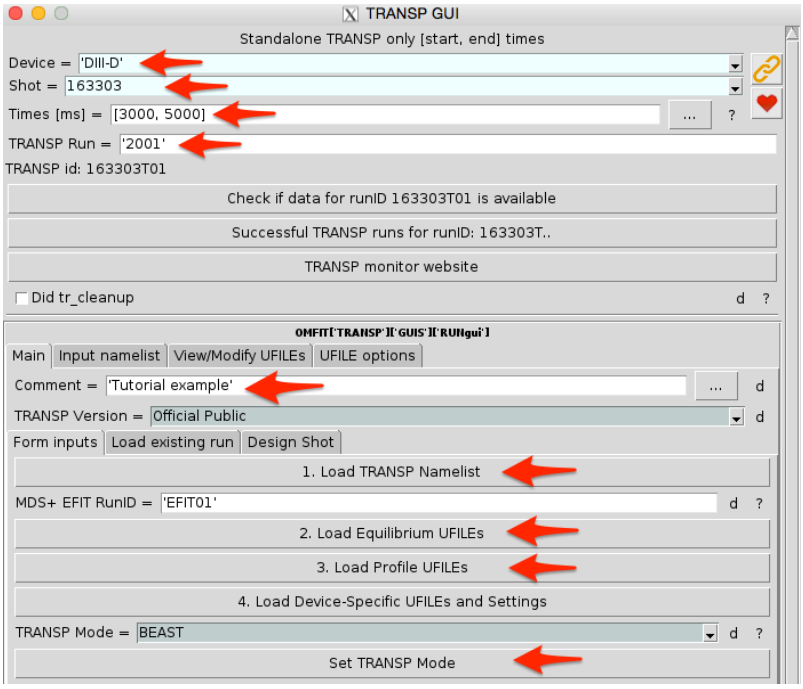
Preparing the inputs

In TRANSP GUI click the following buttons sequentially:

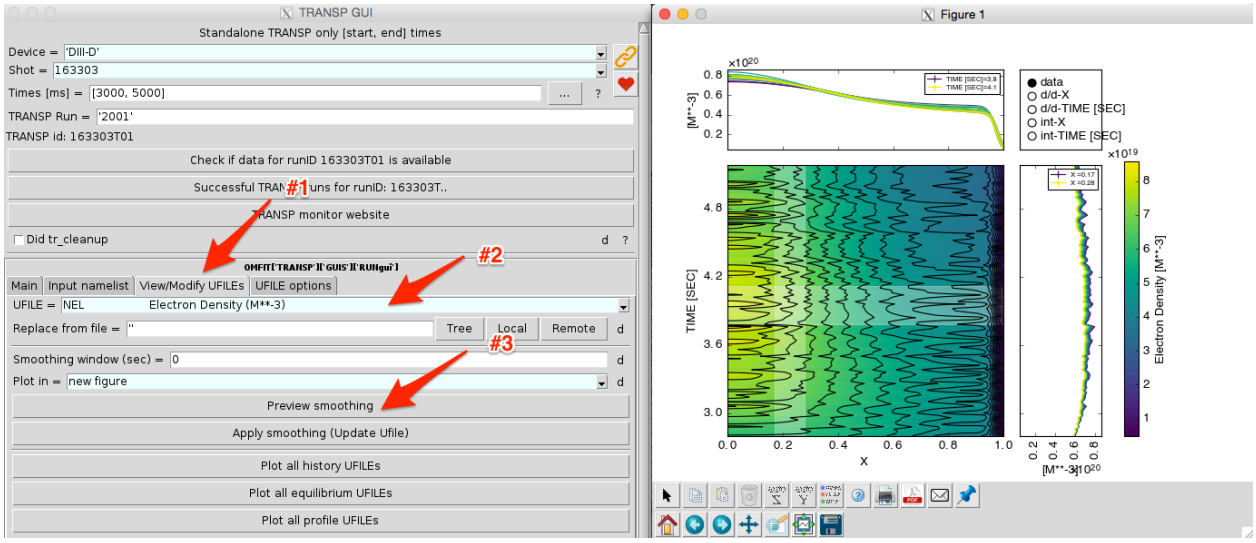
1. Load TRANSP Namelist
2. Load Equilibrium UFILEs
3. Load Profile UFILEs
4. Load Device-Specific UFILEs and Settings

Select TRANSP mode to BEAST

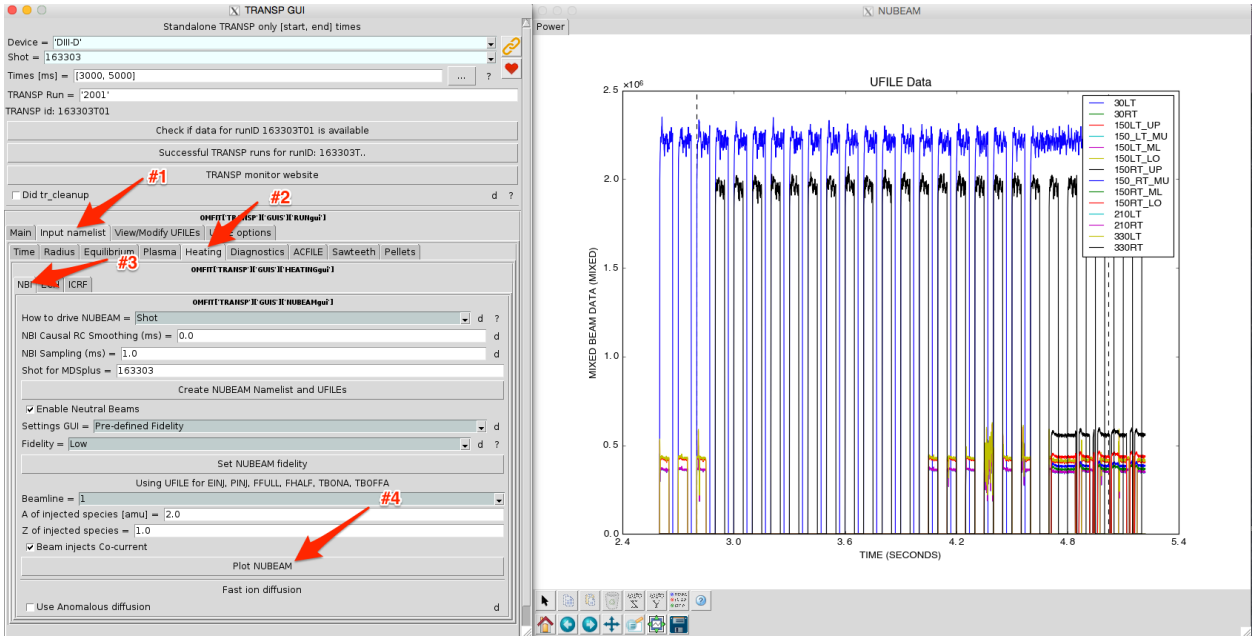
Set TRANSP Mode



You have just prepared all the inputs for TRANSP, and feel free to explore the tab 'Input Namelist', where you can modify at your will. One common set of quantities to view are the input profiles and the NBI and ECH histories, shown below.

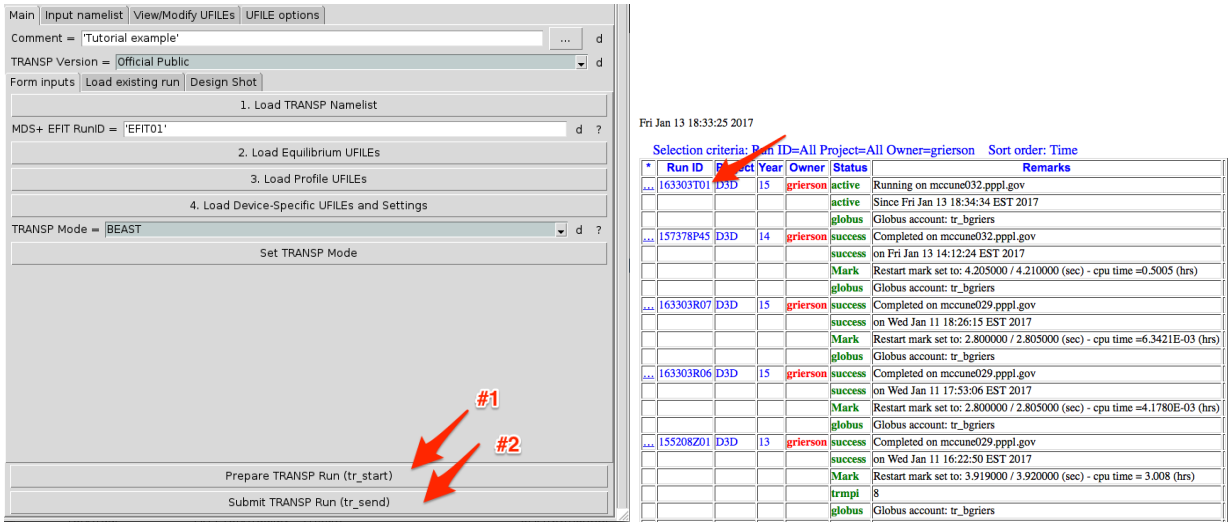


Previewing input density



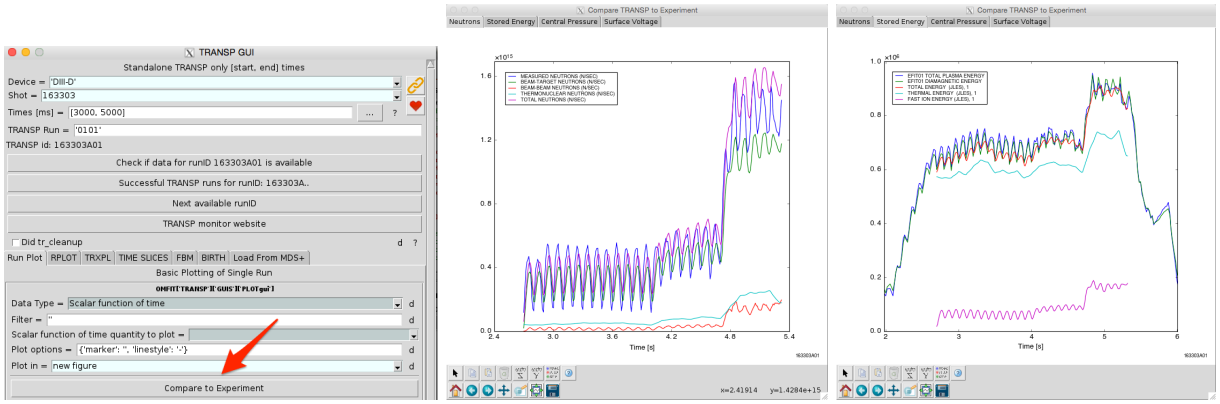
Previewing input NBI heating

Next prepare and then submit the run, shown below. The `tr_start` and `trdat` logs can be viewed in `TRANSP['OUTPUTS']['tr_start.log']` through your text editor.



In BEAST Mode, this run will take 40 min to cover the run, or 20 min / tokamak second. In analysis mode, this run will take 6.4 hours, or 3.2 hours / tokamak second.

When the run is completed, click the button ‘Compare to Experiment’ to view the neutron rate and stored energy comparison to measurements and EFIT.



Compare your results to the experiment for data consistency

This run T01 was also repeated in Analysis Mode (T02), and kineticEFIT Mode (T03). We can compare the accuracy of the two modes by using RPLOT and displaying the fast pressure, neutrons, heating and torque.

Between and Amongst Shots TRANSP (BEAST)

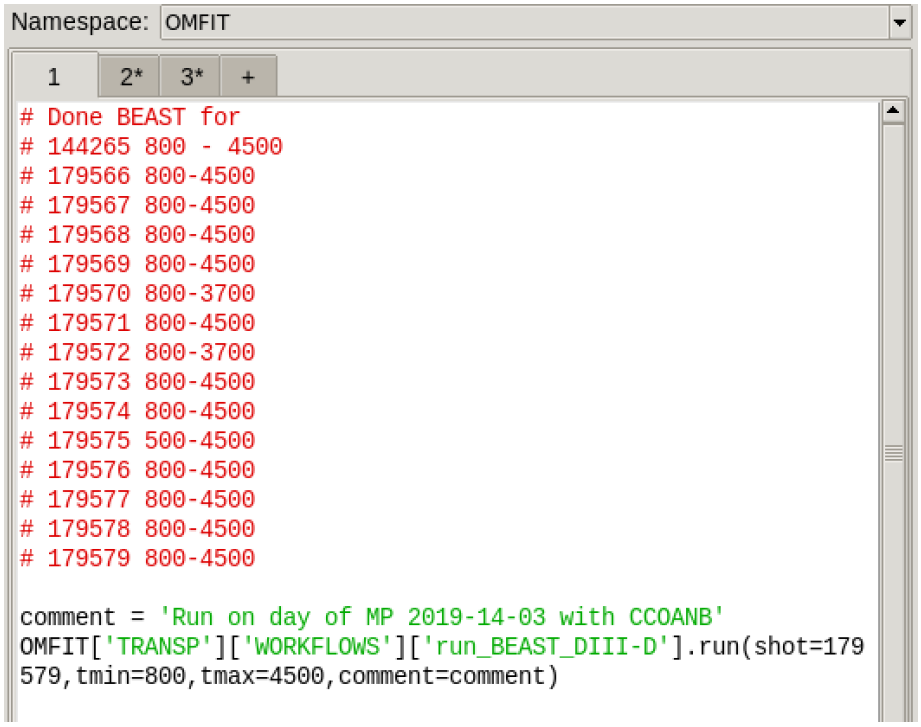
A very simple means of running TRANSP for a “control room” or “survey” level of analysis is with a predefined TRANSP script TRANSP[‘WORKFLOWS’][‘run_BEAST_DIII-D’]. This is executed from the OMFIT command box after importing a TRANSP module with a few options and uses TRANSP runid `O` for “Ops” or “Operations” meaning control room operations. The runid `O01`, `O02`, etc… will be incremented automatically.

- Specifying a comment for that particular run day
- Specifying the shot number
- Specifying the start and end time
- Running both an interpretive “analysis” run as well as a magnetic field diffusion “magdif” run.

An example set of commands in the command box are as follows:

```
comment = 'BEAST for experiment <blah>'
shot = 163303
tmin = 400.; tmax=5500.
OMFIT['TRANSP']['WORKFLOWS']['run_BEAST_DIII-D'].run(shot=shot,
tmin=tmin, tmax=tmax, analysis=True, magdiff=False, comment=comment)
```

This run should complete in the fastest possible time.
Best practices have yielded the use of the command box as the notes for verifying which shots have been submitted with which settings as shown below.
Then save the project based on the MP number like `2019-14-03_BEAST.zip` to encapsulate the settings used for this experiment.



Standalone TRANSP - User Analysis

User analysis assumes that you have prepared either offline or within your OMFIT project EFITs and/or Profiles. A common workflow is to use GAprofiles or OMFITprofiles on an existing MDSplus EFIT, which is shown next. Then, we show how to make your own EFITs or load existing from disk. The single guiding principle is that if `EFITtime`, `GAprofilestime`, or `OMFITprofiles` are in the OMFIT project, then TRANSP will try to use that data, otherwise TRANSP falls back to the defaults of MDSplus EFIT and ZIPFIT.

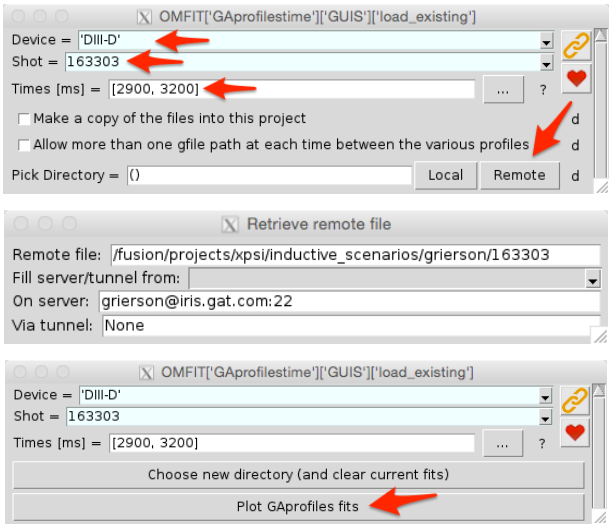
User GAprofiles

Load modules TRANSP and GAprofilestime into OMFIT
Open the GAprofilestime GUI and enter the following:

Device = `'DIII-D'`
Shot = `163303`
Times = `[2900,3200]`

Click the button 'Remote'

Enter the information for where the GAprofiles fits are located (a common XPSI area for this example) and then [return]



And you can now plot the fits.
TRANSP “knows” that the GAprofiles module is loaded in OMFIT through the TRANSP module dependencies.
Open the TRANSP GUI and enter the following:

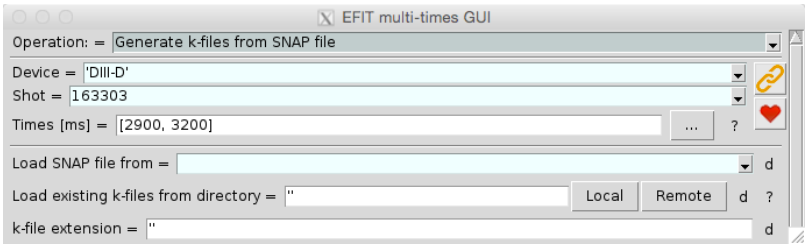
Device = `'DIII-D'`
Shot = `163303`
Times = `[2900,3200]`
TRANSP Run = `'T01'`

Then continue as in the Automatic Analysis - Preparing the inputs above and TRANSP will run with an MDSplus EFIT of your choosing and your GAprofiles fits.

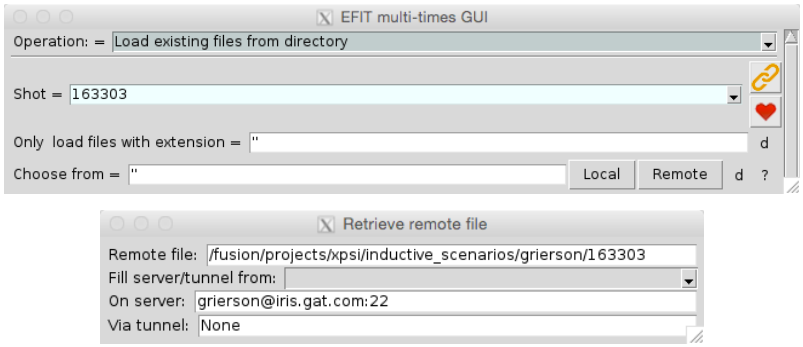
User EFITs (non-MDSplus)

Load modules TRANSP and EFITtime into OMFIT

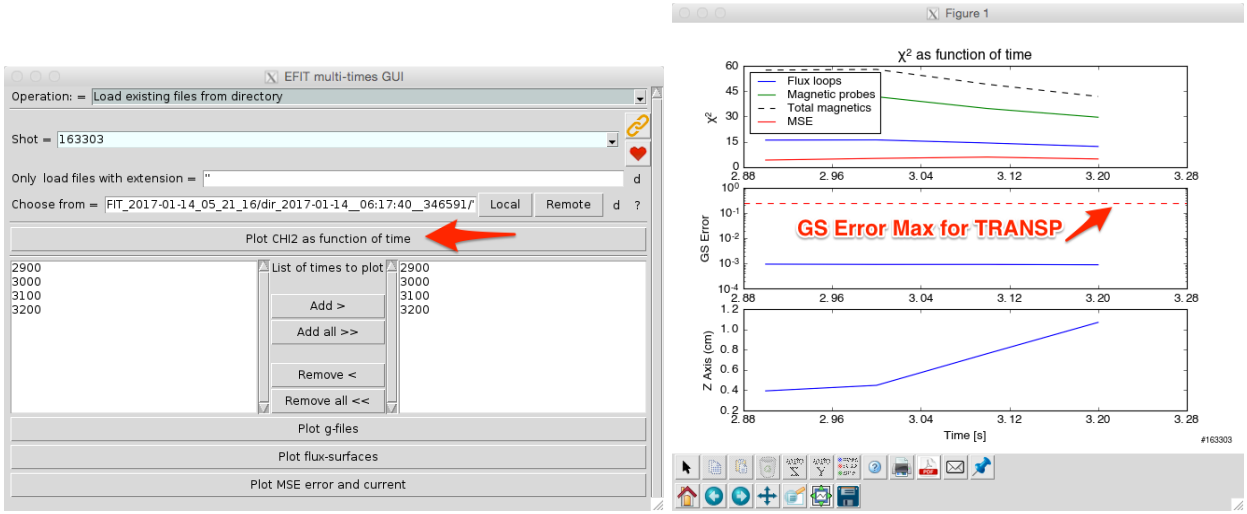
If you would like to create your own EFIT, then execute the common workflow of generating k-files from a snap file, and run EFIT



If you have prepared EFITs on disk, then change the ‘Operation’ to `Load existing files from directory`
If you have created your files with an extension, such as ‘`_run1`’, or ‘`_kinetic`’ then enter that information, but this has not been done for this tutorial.
Click ‘Remote’ and enter the information below, followed by [return]



The GUI reforms to plotting. Here you should plot the chi2, error and Z Axis to make sure that your EFITs aren't crazy. If the Grad-Shafronov error is too high then it can crash TRANSP.



Open the TRANSP GUI and enter the following:

Device = `'DIII-D'`
Shot = `163303`
Times = `[2900,3200]`
TRANSP Run = `'T01'`

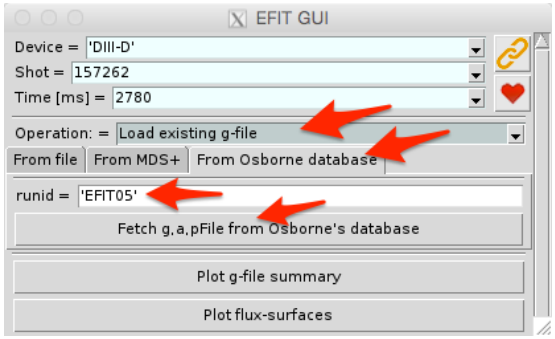
Then continue as in the Automatic Analysis - Preparing the inputs above and TRANSP will run with your user EFITs and ZIPFITs if you do not have GAprofilestime or OMFITprofiles in the project, otherwise it will use GAprofilestime or OMFITprofiles.

User Osborne Profiles

Load modules TRANSP and EFIT into OMFIT. We will use the OMFIT EFIT module to load the EFIT files (gEQDSK, aEQDSK) and profiles (pFile)
Open the EFIT GUI and confirm the following:

Device = `'DIII-D'`
Shot = `157262`
Time = `2780`

Change the operation from “Generate g-file” to “Load existing g-file”, and move to the “From Osborne database” tab.
Enter `'EFIT05'` into the runid entry.
Click “Fetch g, a, pFile from Osborne’s database”



We have now loaded the Osborne EFIT and profiles into the OMFIT EFIT module.
Note that we must know the timestamp on the gEQDSK file, and we must run TRANSP across this time with at least a 1.0 ms buffer, as this is the time that will be used to define Ip, Bt, etc…
Open the TRANSP GUI and enter the following:

Device = `'DIII-D'`
Shot = `157262`
Times = `[2779, 2781]` ← Must be time in gEQDSK file as [time-1, time+1]
TRANSP Run = `'T01'`

and then click “Next Available Runid”
Continue onto steps 1,2,3,4, prepare and submit the run like above in the Automatic Analysis - Preparing the inputs above and TRANSP will run with Osborne EFIT and pFile profiles.

TRANSP and NUBEAM Modes

The number of options for configuring TRANSP for a specific experiment is vast. However, there are some common workflows for which TRANSP is used, ranging from between-shot quick analysis to highly detailed time evolution of a short neutral beam pulse. TRANSP in OMFIT has three basic modes of operation for quick and BEtween and Amongst Shot TRANSP analysis (BEAST), standard power balance analysis (Analysis), and fast resistive current diffusion mode (kineticEFIT). More can be added as needed.

BEAST

BEAST mode is intended to be for fast and control-room level of analysis, but not so fast that the results are meaningless. This should only be used to check things like neutron rates and stored energy for diagnostic consistency due to the coarse spatial grids and time steps and low NUBEAM fidelity. This can introduce time lags on some quantities if the heating sources are modulated. BEAST mode for DIII-D has the following settings:

- Radial resolution `NZONES = 20`

- Time grids for input data `TGRID1, TGRID2 = 0.020` sec
- Time grids for output data `SEDIT, STEDIT = 0.020` sec
- NUBEAM mode - Low fidelity

Analysis

Analysis mode is intended to be for routine level of power balance analysis, sufficient for particle and energy balance, but marginal for detailed momentum transport (due to JxB torque) or fast-ion (smooth distribution function) physics. Tracking an individual ~10 ms NBI pulse will require further decrease of the input time base for TGRID1 down to 2 ms or less.

- Radial resolution `NZONES = 50`
- Time grids for input data `TGRID1, TGRID2 = 0.010` sec
- Time grids for output data `SEDIT, STEDIT = 0.010` sec
- NUBEAM mode - Medium fidelity

kineticEFIT

kineticEFIT mode is for resistive current evolution. A premium is placed on the radial resolution to capture the edge bootstrap current, but fidelity is sacrificed on the neutral beam model for time to solution because only the fast pressure is required. The time grids are the same as analysis mode.

- Radial resolution `NZONES = 100`
- Time grids for input data `TGRID1, TGRID2 = 0.010` sec
- Time grids for output data `SEDIT, STEDIT = 0.010` sec
- NUBEAM mode - Low fidelity

NUBEAM modes

NUBEAM is the most time consuming part of TRANSP for power balance. Three levels of pre-defined fidelity are available.

Low Fidelity

Intended for control room analysis for time to solution, and kineticEFITs
Monte-carlo particles `NPTCLS=1000`
Deposition tracks `NDEP0 = 500`
Beam timestep `DTBEAM = 0.005`
Numerical acceleration `GOOCON = 5.0`
No usage of MPI parallel processing because the overhead outweighs the benefits

Medium Fidelity

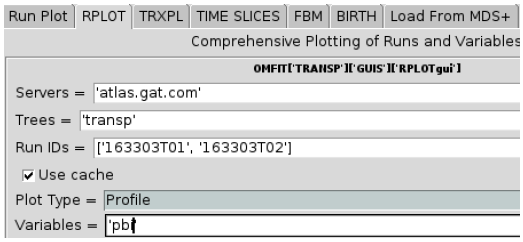
Intended for standard particle and power balance calculations and reasonable torque density
Monte-carlo particles `NPTCLS=32000`
Deposition tracks `NDEP0 = 5000`
Beam timestep `DTBEAM = 0.001`
Numerical acceleration `GOOCON = 10.0`
MPI - 8 cores for 4,000 particles / core

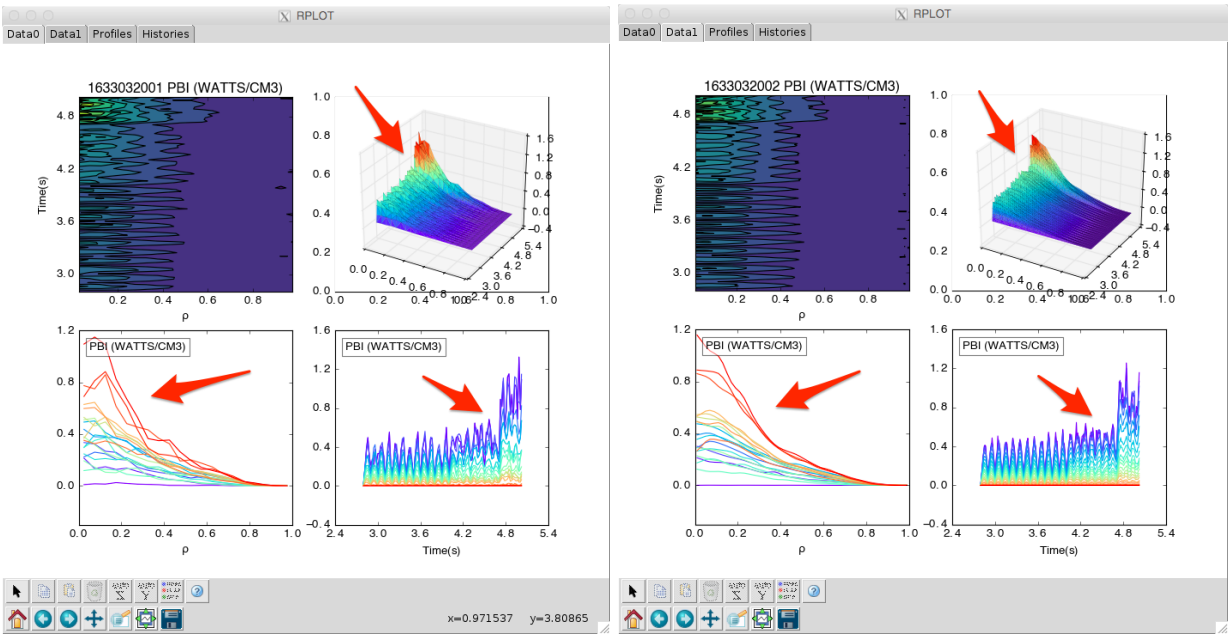
High Fidelity

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

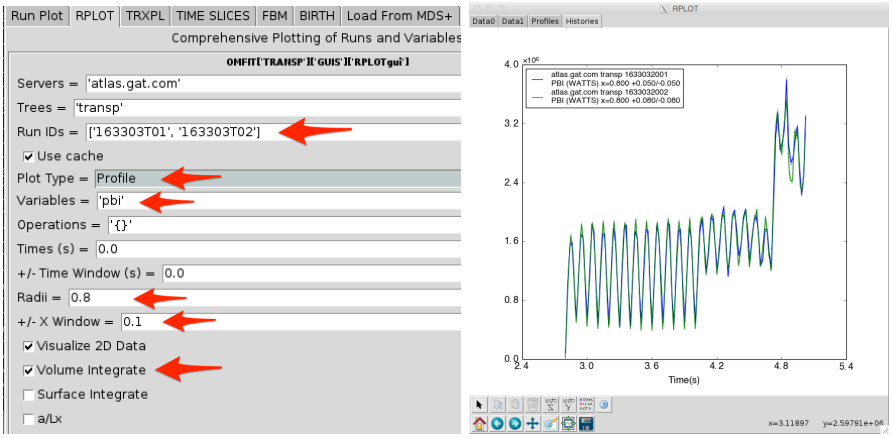
Comparing Modes

Tutorial TRANSP run T01 was repeated in Analysis mode (T02) and kineticEFIT mode (T03). Compare the beam heating:



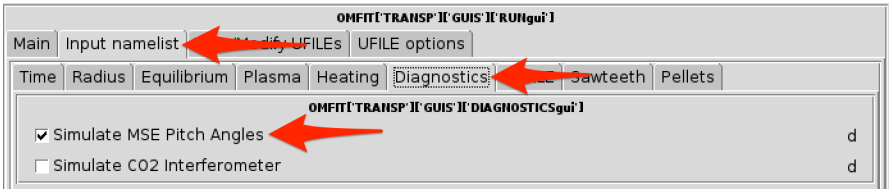


BEAST mode (left) and Analysis mode (right) beam power to ions

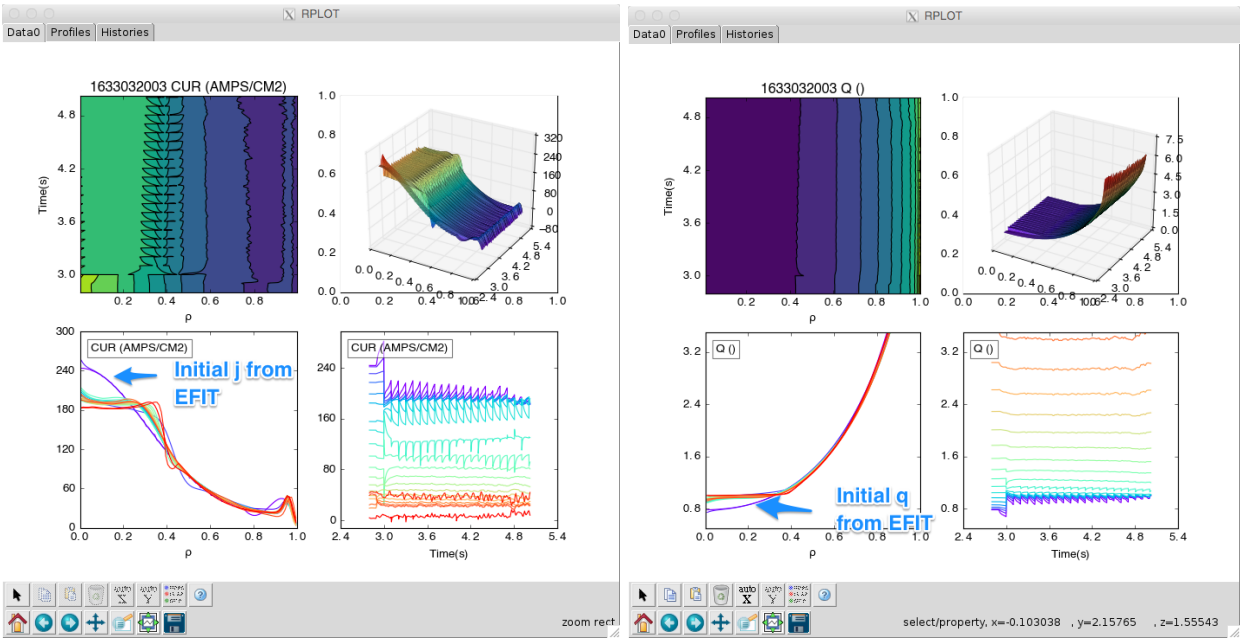


Volume integrated ion power flow to rho=0.8 is still the same

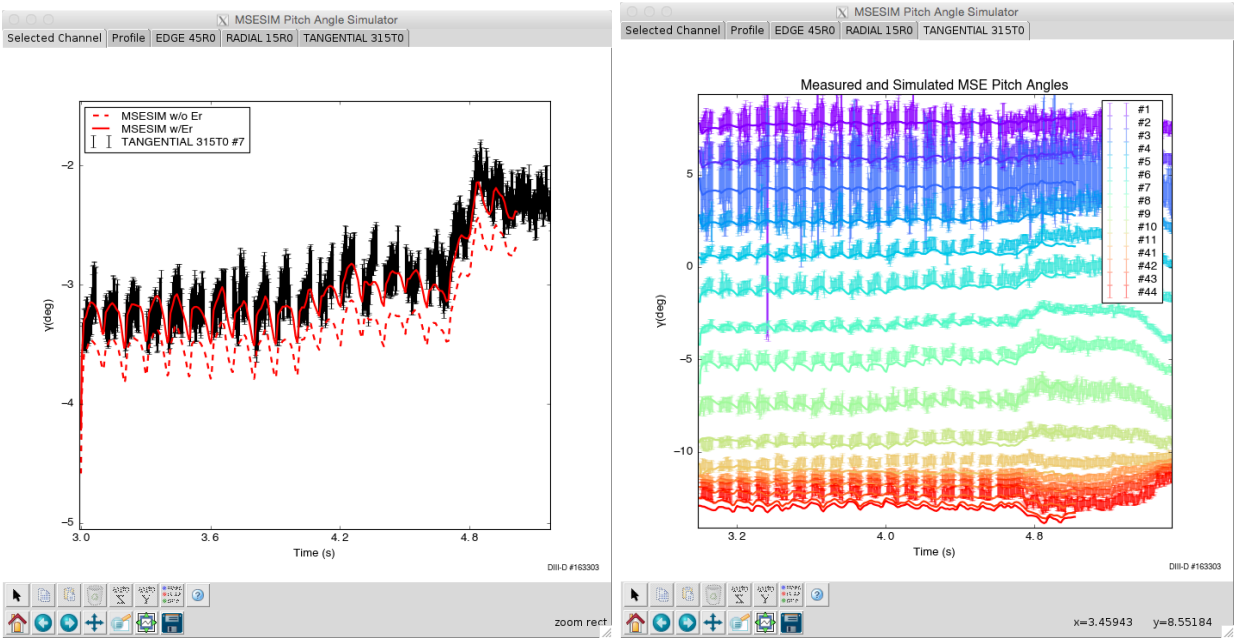
The kineticEFIT mode also included the MSESIM pitch angle simulator with a sawtooth every 100 ms, which is approximately accurate for this shot.



The prediction of the current density, q and MSE pitch angles with and without the effect of Er are plotted below.



Plasma current density (left) and q (right). Sawtooth crashes are evident.



MSE #7 pitch angle with and w/o Er (left) and all channels (right).

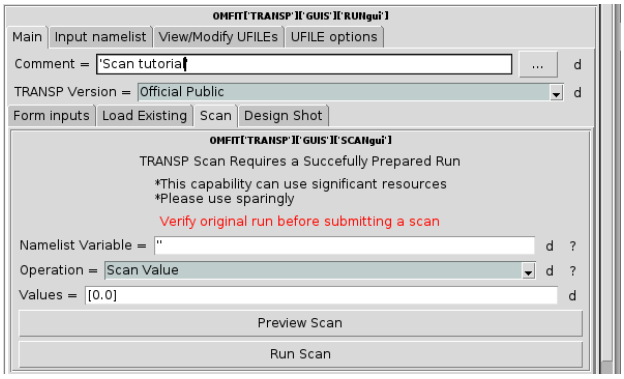
TRANSP Variable Scan

It is common to perform a scan of an input namelist variable, such as fast-ion diffusion or plasma Zeff. We have now provided a scan capability in OMFIT so that the user does not need to repeat the process of preparing each run by hand. There are a few critical things that the user needs to be aware of.

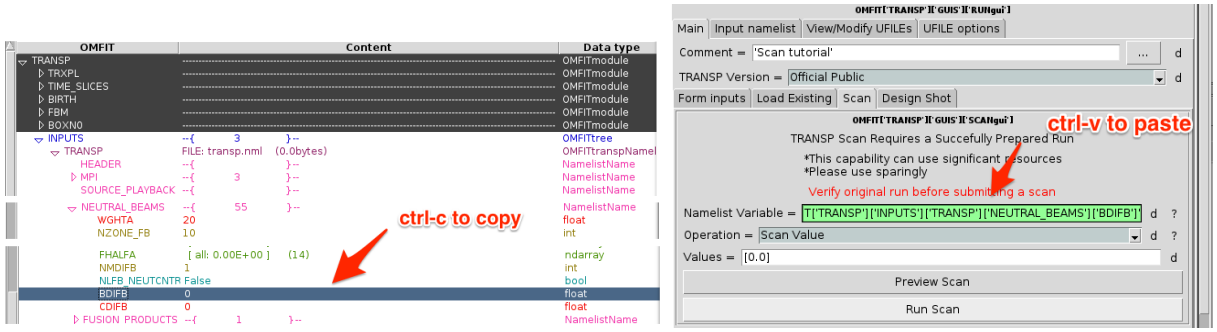
- This capability can use significant computational resources at PPPL and therefore should be used sparingly, especially if using multiple CPUs
- You must have a ready-to-run TRANSP preparation, with successful completion of the namelist, equilibrium, profile and heating UFILEs and settings. We suggest completing one successful TRANSP run that does not bomb first.
- This capability *will not* set TRANSP switches for you; it *will only* scan a variable. Therefore, scanning the beam anomalous diffusion values (`BDIFB`, `CDIFB`) will have no effect if the switch (`NMDIFB`) is not enabled!
- Some variables will be scanned together. Presently if you select to scan either `BDIFB` or `CDIFB` then both `BDIFB` and `CDIFB` will be scanned together. We will add more as needed.

The workflow is therefore

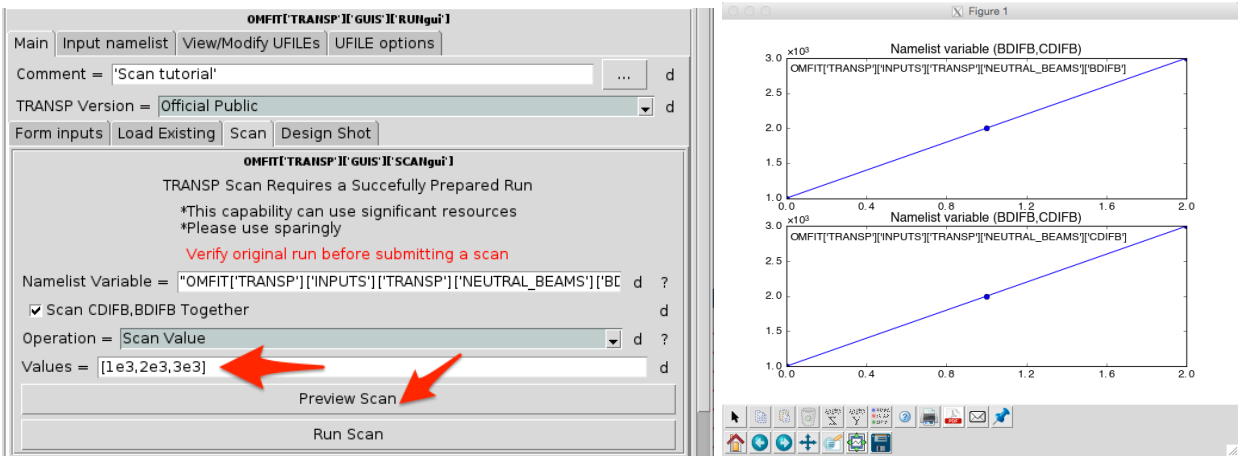
1. Prepare and run TRANSP for your particular case. Verify that the run has completed and that the fidelity is correct for your chosen application.
2. Open the TRANSP GUI and select the next runID
3. Browse tabs Input Namelist → Heating → NBI and select anomalous diffusion
4. Move to the “Scan” tab
5. Enter a string containing the location of the TRANSP namelist variable you want to scan. This is most easily accomplished in three steps: 1) Enter a single double-quotes (“) into the text box in the Scan GUI. 2) Browse in the tree to the variable you want and ctrl-c to copy. 3) View to the Scan GUI and ctrl-v to paste, and enter another double-quotes (“) and type [return].
6. Enter the numerical values for your scan i.e. [1e3, 2e3, 3e3], or a python expression such as np.linspace(1e3,3e3,3)



The Scan GUI



Selecting a variable to scan



Entering and previewing a scan

In the terminal, the following useful information will be shown:

```
Scan variable location:
OMFIT['TRANSP']['INPUTS']['TRANSP']['NEUTRAL_BEAMS']['BDIFB']
Present Value: 0.0
Scan values:
 [ 1000.  2000.  3000.]
Will attempt runIDs:
 ['1821', '1822', '1823']
Checking scan...
... done checking
```

And shows the variable location, present value, and values to be scanned. Also, the runIDs that will be attempted are shown.

When you run the scan, the comment will be appended with the following information:

```
Scan tutorial
 OMFIT TRANSP SCAN
 Scan variable NEUTRAL_BEAMS BDIFB
 Scan variable NEUTRAL_BEAMS CDIFB
 Scan value 1000.0
```

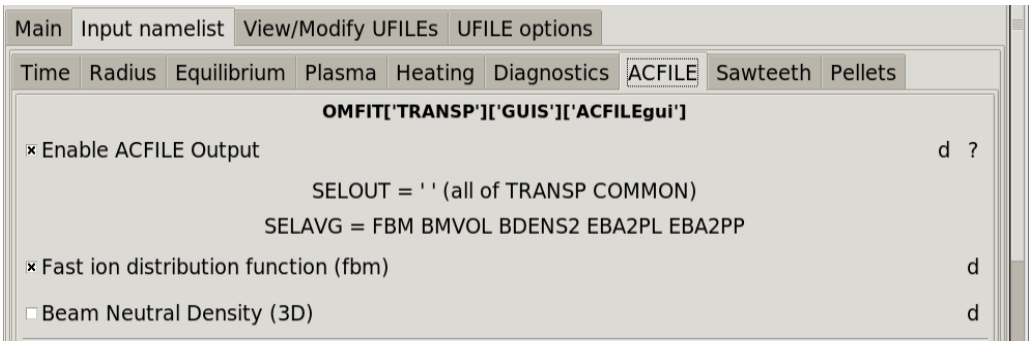
Plotting common scan variable results

There are two types of common scans for interpreting TRANSP results from an experimental discharge. The first is fast-ion anomalous diffusion, which primarily affects the output neutron rate and fast-ion stored energy. The second is the plasma Zeff, which affects the neutron rate, stored energy and plasma resistivity that modifies the predicted surface loop voltage when evolving the current. OMFIT provides post-processing of both of these types of scans in RPLOT.

ACFILES for Fast-ion analysis

Before submitting a TRANSP run with a request for an ACFILE it is best to submit a run without an ACFILE to ensure there are no problems with the prepared namelist or files.

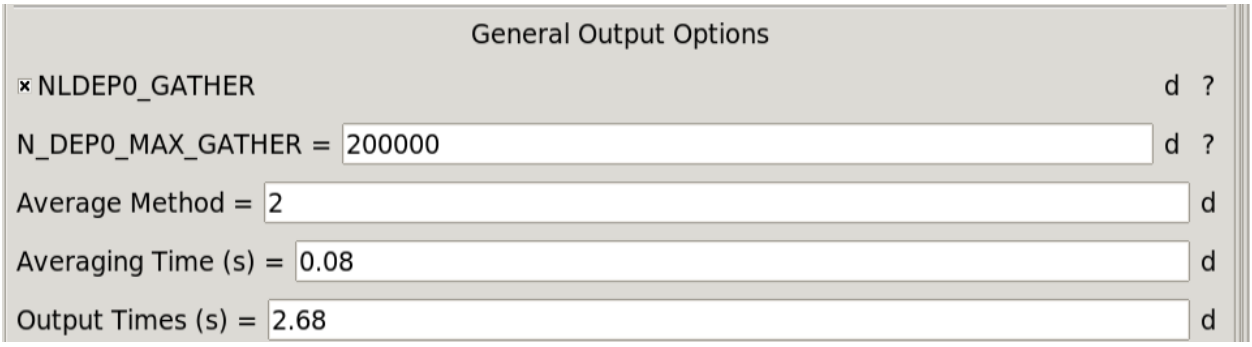
Starting with this run already loaded into the TRANSP gui, navigate to the Input namelist tab then ACFILE. Then select the following checkboxes: Enable ACFILE output and Fast-ion distribution function.



Then setup the requested setup for your ACFILEs. Consider if you want higher fidelity beams and what time you want, if you only want one time point change your initial and final times to reduce the amount of time the run will take.

It is recommended to set NLDEP0_GATHER to True (checkbox). Then select N_DEPO_MAX_GATHER (total number of gather track) according to AVGTIM/DTBEAM*NDEP. Too small a number and beams could be missed and too big of a number could cause memory problems. Average Method=2 (at the end of the output time) is usually used. Select averaging times and output times. The distribution can be dumped up to 200 times from one TRANSP run.

In the example here, the fast-ion distribution and birth profiles will be averaged from 2.6-2.68 s.



Then submit your run as you normally would.

Flight Simulator (Design Shot)

TRANSP has been used for experimental power balance for decades, and predictions of plasma evolution for nearly as long. Historically, predictive TRANSP has been referred to as PTRANSP, however there has only ever been one TRANSP code. PTRANSP has typically referred to running TRANSP with most of the quantities derived from user assumptions of models rather than experimental inputs. Some examples are ITER predictions, and replacing measured fluid variables with a transport model (GLF23, TGLF, etc…) or equilibrium with fixed boundary GS solution (TEQ) or free boundary solution (Isolver).
If you have completed the above tutorial example for automatic analysis, you will understand that TRANSP needs a namelist, equilibria, profiles, and the time history of actuators like NBI.

When designing a shot in OMFIT, the guiding philosophy for the workflow is that we can take *nearly everything* from an existing experiment, or *practically nothing*, and the *entire spectrum in between*. For example, take the equilibrium from one shot, and the heating power from another. Or take everything from a shot, but turn off the ohmic transformer and let the current relax for seconds beyond the discharge termination to assess the steady-state q profile.

The TRANSP “Design Shot” workflow is like a flight simulator for the tokamak control room. We rely on the machine operator to deliver the field, current and plasma shape to the session leader. Therefore, we first begin by assuming that we control these parameters. We also design some model plasma profiles (ne, Te, Ti, rotation, Zeff) as a starting point to initialize TRANSP, which are reasonable for the chosen scenario starting point, such as L-mode or H-mode.

Inductive Scenario Modeling

For inductive scenarios, the experimentalist controls the total plasma current.
In this example, we will use a DIII-D reference shot and input the time evolution of Ip, Bt, Vsur and crude approximations of the expected plasma profile evolution. This is then a starting point

for more sophisticated predictions using transport models and free boundary equilibrium evolution for taking the target shot away from what was executed.

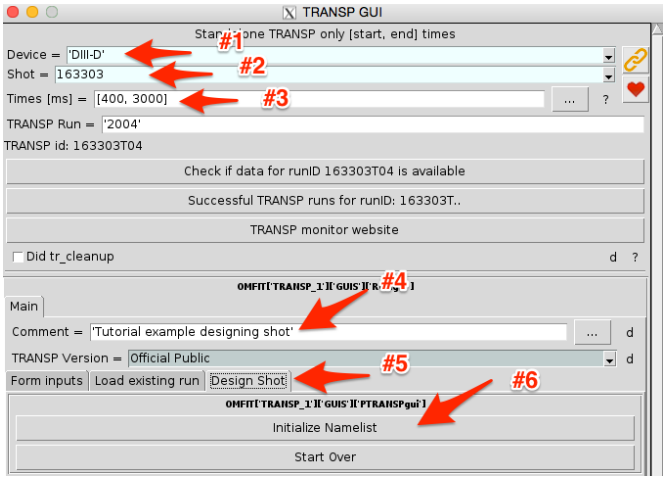
Shot/Times

We can use a shot number that is not an actual shot, but for this example it is convenient to use a plasma shot that we are nearly replicating.
Open the TRANSP GUI and enter the following:

Device = `'DIII-D'`
Shot = `163303`
Times = `[400,3000]`
TRANSP Run = `'T01'`

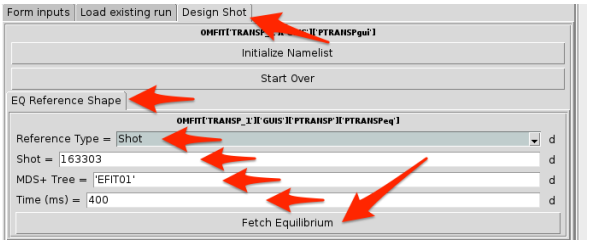
Click the button 'Next Available RunID' to get the next free run ID.

Next move on to enter a comment, such as `'Tutorial example designing shot'` and initialize the namelist. Update the start and end time under Input Namelist -> Time accordingly.



Equilibrium

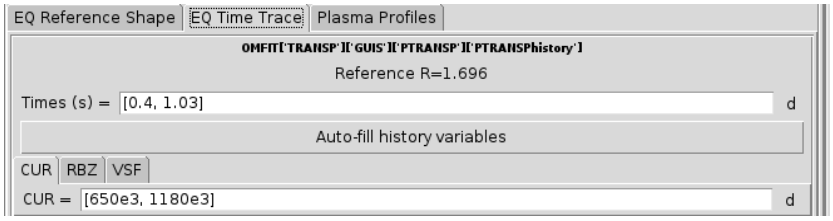
For this case we use reference equilibrium EFIT01 at 400 ms, which is after the shape has formed and the plasma is diverted. Here we get the equilibrium for this case.



When you fetch the equilibrium, we also set the EQ solver to TEQ inside of TRANSP.

The critical times and values for IP for this shot are as follows:

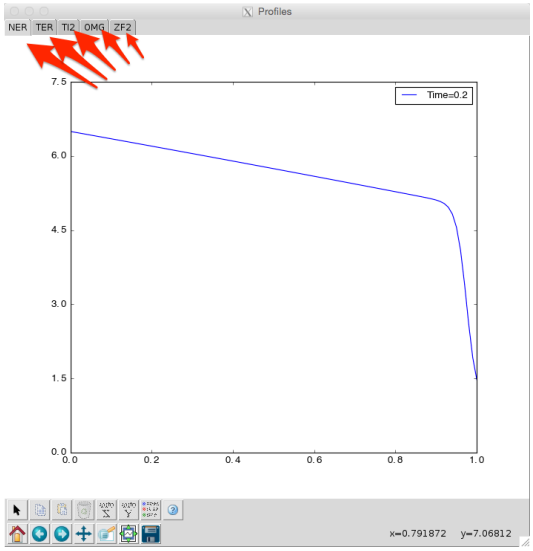
Time=0.4, IP=0.65e6 Amps - the early ramp-up phase from our reference equilibrium
Time=1.03, IP =1.18e6 Amps - the current flat top time and value
On the EQ Time Trace enter Times(s) = `[0.4, 1.03]` [return]
On the CUR Tab enter `[650e3, 1180e3]` [return]
On the RBZ Tab enter `[-3.4, -3.4]` [return]
On the VSF Tab enter `[1.7, 1.7]` [return]



Then click ‘Create UFILEs and Preview” and you can see what you’ve created, which is simply the Ip ramp-up.

Profiles

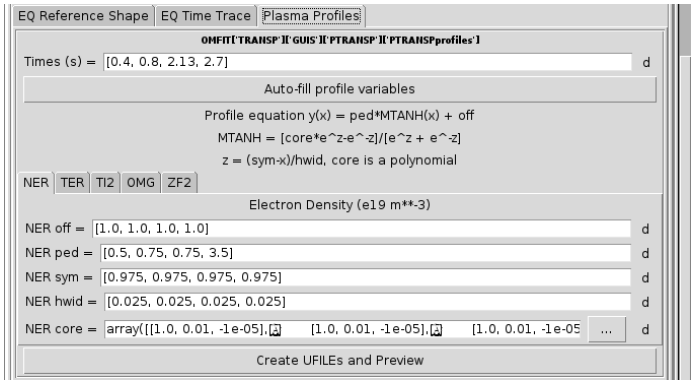
We are going to input full plasma profiles using a modified hyperbolic tangent as our design function, shown in the GUI. There are a few key times that we must be aware of for this case:
Time = 0.4 - start of the input data for the TRANSP run
Time = 0.8 - density rise until 0.8 seconds
Time = 2.13 - Just before the L-H transition
Time = 2.70 - End of density rise after the L-H
We will design profiles at each of these times.
First, click “Create UFILEs and Preview” to see where we are starting.



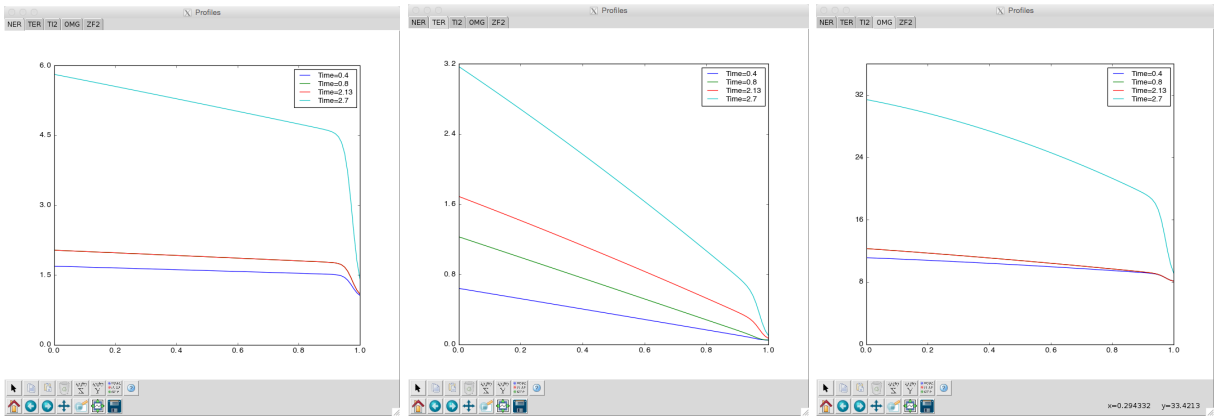
Change these model profiles to typical L-mode parameters
Change NER ped = `[0.5]`
Change TER ped = `[0.01]`
Change TI2 ped = `[0.01]`

Change OMG ped = `[0.01]`
Change ZF2 off = `[1.2]`

On the Plasma Profiles tab enter Times(s) = `[0.4, 0.8, 2.13, 2.7]` [return]
Click “Auto-fill profile variables”, which will complete each profile based on the initial L-mode conditions, to be further modified.
Now continue by changing the following:



Change NER ped = `[0.5, 0.5, 0.5, 0.5]` → `[0.5, 0.75, 0.75, 3.5]`
Change TER and TI2 ped = `[0.01, 0.02, 0.2, 0.5]`
Change TER and TI2 core second column = `[1.5, 1.5, 0.2, 0.15]`
Change OMG ped = `[1.0, 1.0, 1.0, 1.0]` → `[1.0, 1.0, 1.0, 10.0]`
Change OMG core second column = `[0.07, 0.1, 0.1, 0.05]`
Change ZF2 off = `[1.2, 1.2, 1.2, 2.0]`
Click “Create UFILEs and Preview”



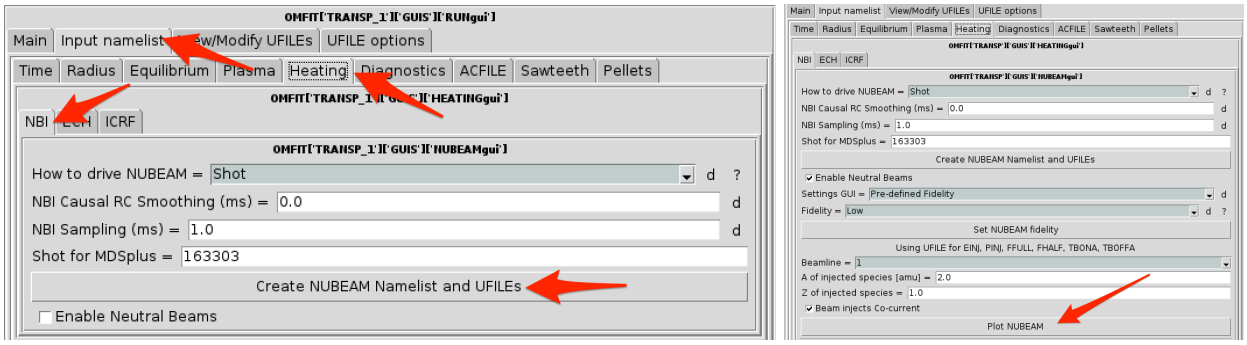
Preview of Electron Density, Electron and Ion Temperatures, and Rotation

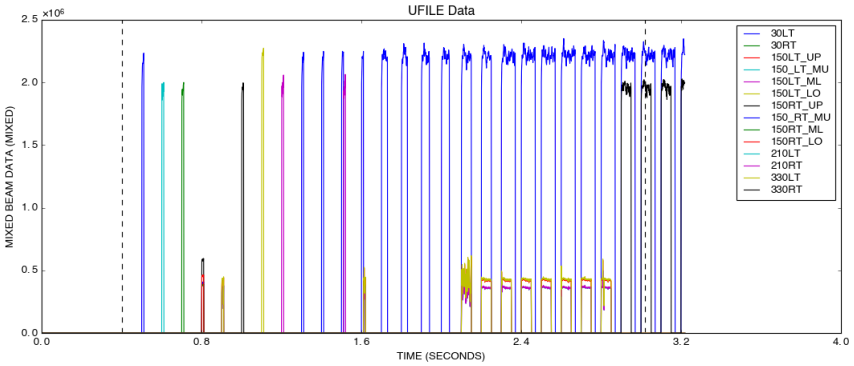
At this point we have prepared the EQ geometry, equilibrium history and plasma profiles. Now we have to design the heating and submit the run.

Heating

For this tutorial, we will simply load the neutral beams from the shot, but this is the point from which the user can get creative by designing the heating from NBI, ECH and ICRF. In this

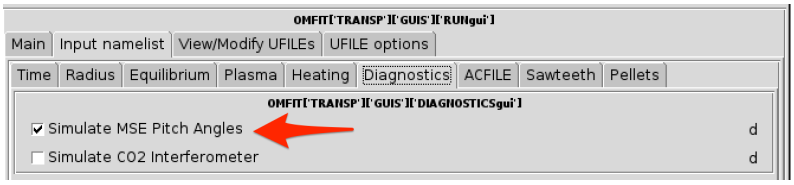
workflow, we drive NUBEAM from the NBI history of the shot, but we can replace this heating program with another shot, or design the heating *ab initio* from namelist only, just like TIMCON in the control room.





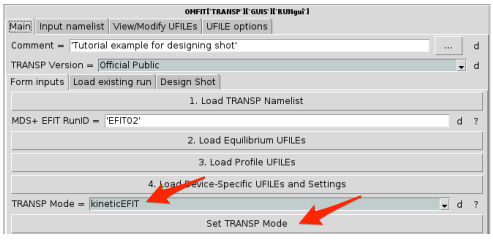
Diagnostics

We want to simulate the MSE pitch angles for this tutorial



TRANSP mode

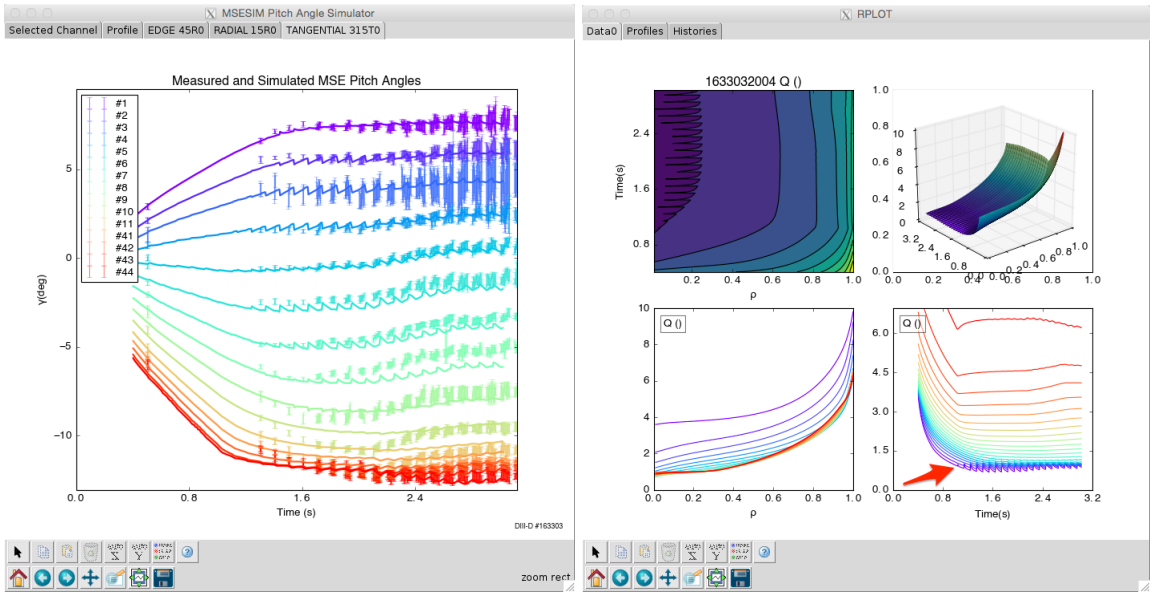
For this tutorial, we will make use of the “kineticEFIT” mode of TRANSP, which sets resistive current diffusion and low fidelity NUBEAM, as well as sawtooth mixing.



Then start and submit the run.

Summary

We have designed a shot, and for this tutorial our design decisions were based on our intuition for how tokamak plasmas evolve, guided from a previous experimental condition. In order to answer a specific scientific question, predict plasma performance or optimize and experimental plan, we may rely more and more heavily on models rather than designer inputs. The balance between what is taken as controllable and what is computed based on models is up to the particular goal. When we view the output of this TRANSP run, we can assess a number of things: How did the pitch angles evolve? What is the Shafranov shift? How much ohmic power is going into the electrons? When was the first sawtooth (it was 1.25 sec in the actual shot)? All of these questions can now be simply answered with the OMFIT TRANSP “Design Shot” workflow.



(left) Predicted pitch angle evolution and measurements. (right) q evolution and first sawtooth

Steady-state modeling

In steady-state modeling, we aim to run the plasma fully non-inductively driven by auxiliary and bootstrap currents. In essence, this changes the equilibrium evolution from matching the total plasma current you designed, to matching the loop voltage you designed; making sure that the loop voltage input is zero. In essence, that is the real difference. Of course you probably want to increase the heating powers for NBCD and use ECCD to drive significant current.

Choosing shot and time ranges

As an example for this workflow, we will investigate a plasma near the conditions of DIII-D discharge 153072, which is a high beta steady-state target. Also, because we are only loosely

basing our investigation on this shot, we will *not* use this shot number for our prediction. We will use the morning power supply test shot #153059 from this experimental day so that the archived TRANSP runs to not appear as analysis runs for plasma shot 153072.

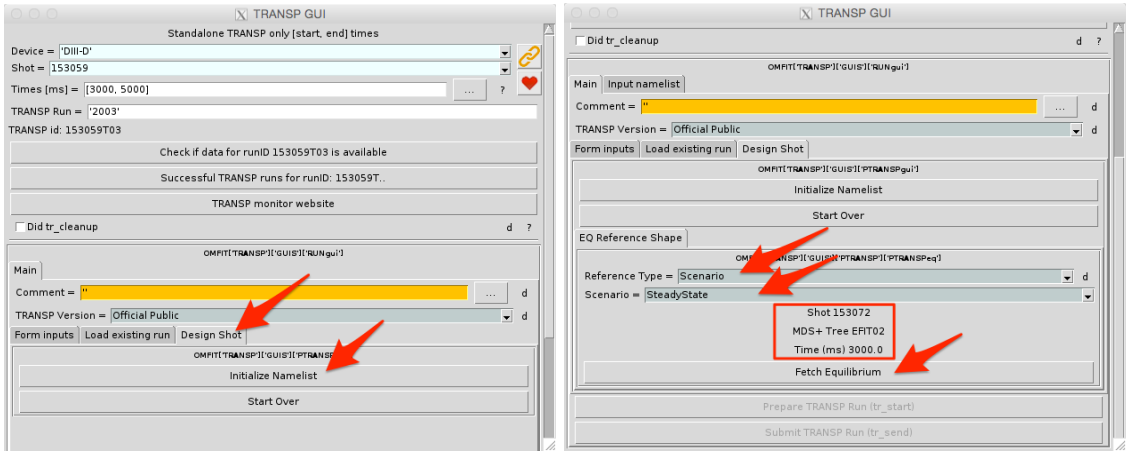
Open the TRANSP GUI and enter the following:

Device = `'DIII-D'`
Shot = `153059`
Times = `[3000,5000]`
TRANSP Run = `'T01'`

Click the button 'Next Available RunID' to get the next free run ID.

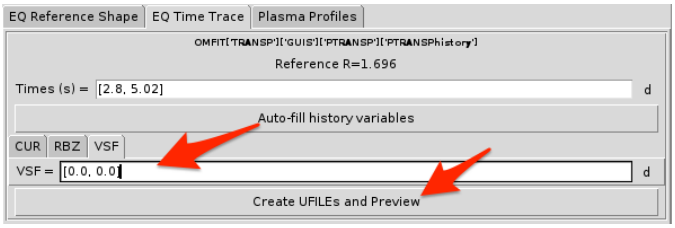
Design EQ, History and Profiles

Move to the tab "Design Shot" and click "Initialize Namelist". Next change from Reference Type = "Shot" to "Scenario", select "SteadyState", and click "Fetch Equilibrium"



Fetching the equilibrium means that we load into the module the equilibrium shape, plasma current, toroidal field and surface voltage taken from the shot. We can modify these things easily, but for the purpose of this tutorial example we will only modify the surface voltage. The reason is that we are trying to answer the following question: "If I turn off the ohmic transformer, What is the steady-state total plasma current, and q-profile, for the heating and current drive that I design", which is selected in the next step. In order to answer this question, we need to set the surface voltage to zero in our predictive simulation.

Move to the Tab "EQ Time Trace", set the surface voltage VSF to `[0.0, 0.0]` [return], and then click "Create UFILEs and Preview". We have now essentially programmed the control system to turn off the ohmic transformer.

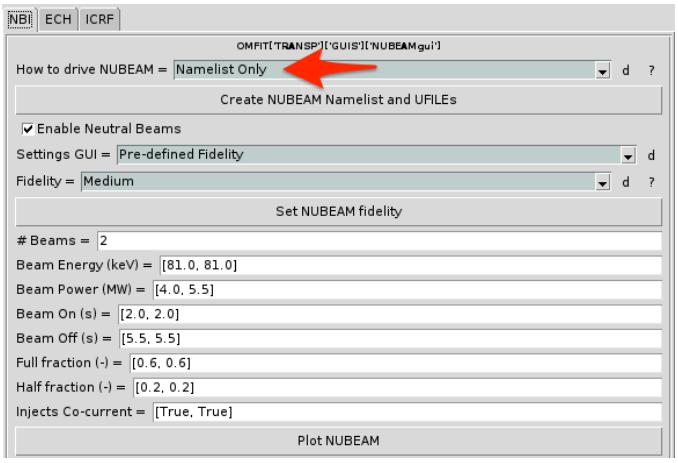


Next we need to design the plasma profiles. We will use reasonable assumptions here that resemble this target shot. Move to the tab “Plasma Profiles” and change the following:
NER ped = `[2.0]`
NER hwid = `[0.05]`
NER core = `array([[1.0, 0.01, 0.002]])`

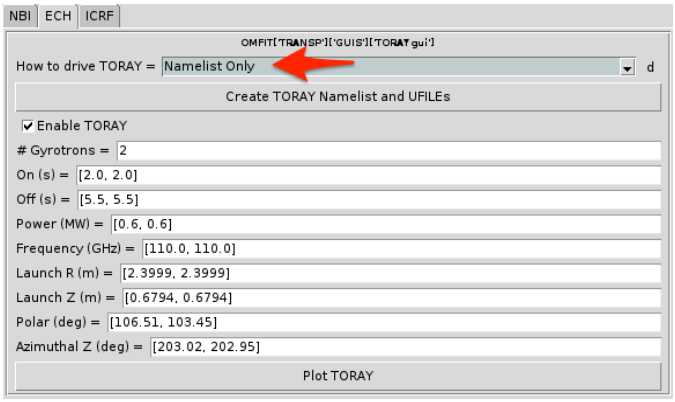
TER sym = `[0.95]`
TER hwid = `[0.05]`
TER core = `array([[1.0, 0.2, -0.004]])`
Then make TI2 like TER. The rotation and Zeff can be left as is.
Click “Create UFILEs and Preview” to view the profiles. These profiles will then be taken as completely time independent in the TRANSP run.

Design Heating and Current Drive

Next we design the heating and current drive. We will use two neutral beam lines that represent the co-tang and co-perp injected power, with values near the sum of 30LT+330LT+150LT and 30RT+330RT+150RT.
Move to the tab Input Namelist → Heating → NBI, select “Namelist Only” and click “Create NUBEAM Namelist and UFILEs”
Enter the values shown below and then you can plot the boring time traces.



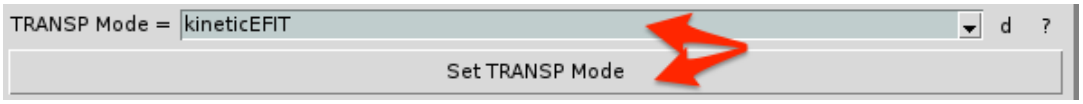
Move to the ECH tab, select “Namelist Only” and click “Create TORAY Namelist and UFILEs” and enter the gyrotron information as shown here:



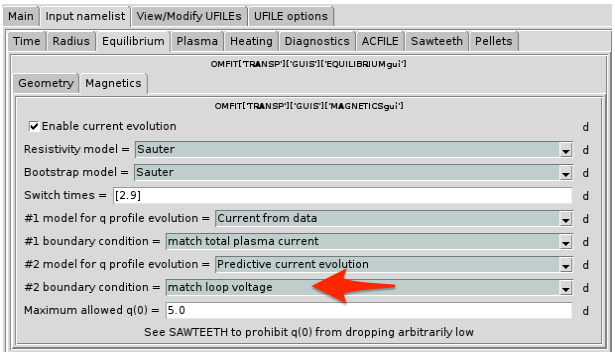
At this point we have set up the equilibrium, time history of scalars (Ip, Bt, Vsur), plasma profiles, and heating and current drive sources. Now we need to set the TRANSP mode, and tweak a few settings specific for our simulation.

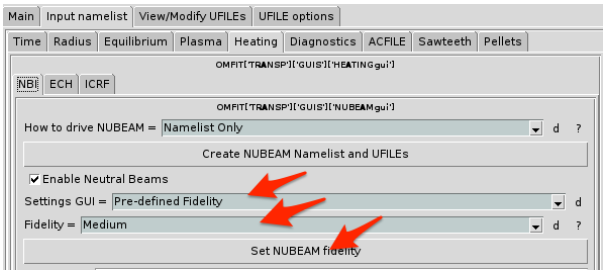
Set TRANSP Mode and Goal-Specific Settings

Return to the Main tab, set the TRANSP mode to “kineticEFIT” and click “Set TRANSP Mode”



For this tutorial, we must customize by changing the boundary condition for the poloidal field diffusion, matching the surface voltage, seen below. We must also increase the NUBEAM fidelity because the neutral beam current drive is important and we want it to be accurate.



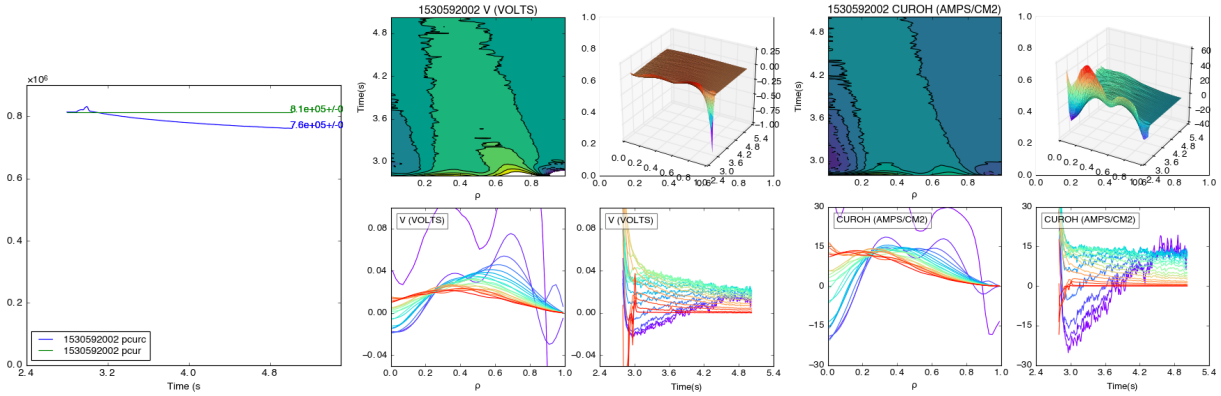


Now we are done preparing this simulation.
Start and then send the TRANSP run.

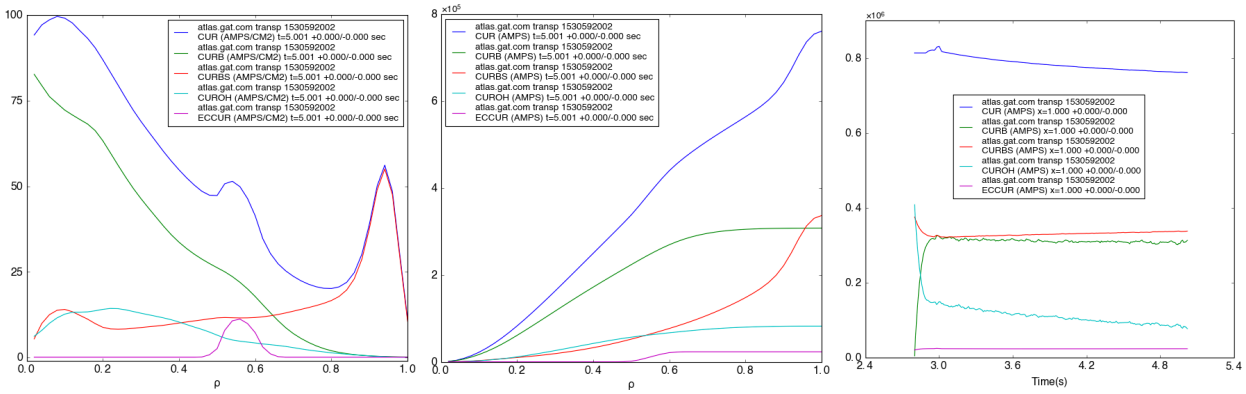
Summary and Plots

In this tutorial example we have designed a shot that will operate fully non-inductively, with the goal of predicting the total current and steady-state q-profile. We have designed the plasma profiles as a reasonable guess for what we expect. As a result, we obtain the time evolution of

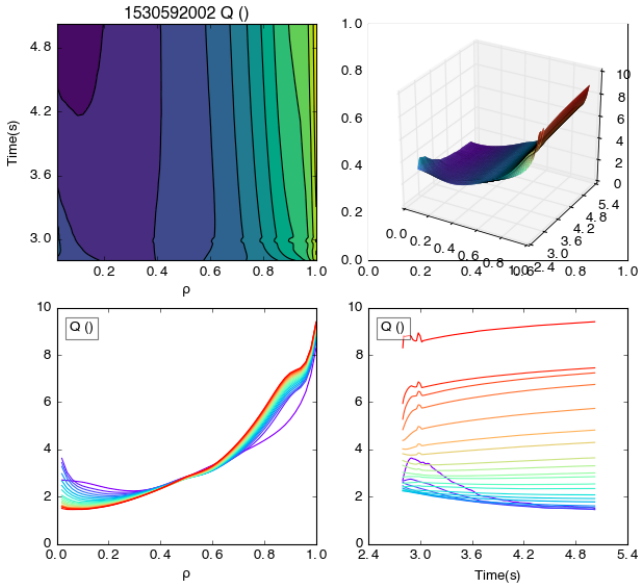
the plasma current density as the toroidal voltage profile and ohmic current density diffuse to zero on a long timescale (longer than our simulation time here). This is a good starting point for performing scans of the NBI power, EC power and aiming, and variation of the plasma profiles that produce the bootstrap current or replacing the profiles with a transport model. Shown below are the time evolution of the quantities of importance:



Input and evolving total plasma current, which is decaying from 0.81 MA to 0.76 MA, decay of the toroidal voltage profile, and decay of the ohmic current density.



Contributions to current density, surface integrated current, and evolution. This case is ~ 50% NB and 50% Bootstrap



q profile, showing lower q(0) and higher q95 due to on-axis current-drive and reduced total plasma current

Use BIRTH and FBM modules to look at fast-ion ACFILES

After your transp run has completed, you need to fetch the ACFILE from PPPL. Navigate to a folder where you want to store all your date. Then fetch the data on the command line with: tr_fetch machinename runID, ex: tr_fetch d3d 172010A02.

This will fetch over the following files:
172010A02_birth.cdf1 172010A02ex.for 172010A02TR.INF
172010A02CC.TMP 172010A02_fi_1.cdf 172010A02TR.MSG
172010A02.CDF 172010A02_nubeam_init.dat 172010A02.yml
172010A02_D3D.REQUEST 172010A02PH.CDF
172010A02.DATA1 172010A02TR.DAT

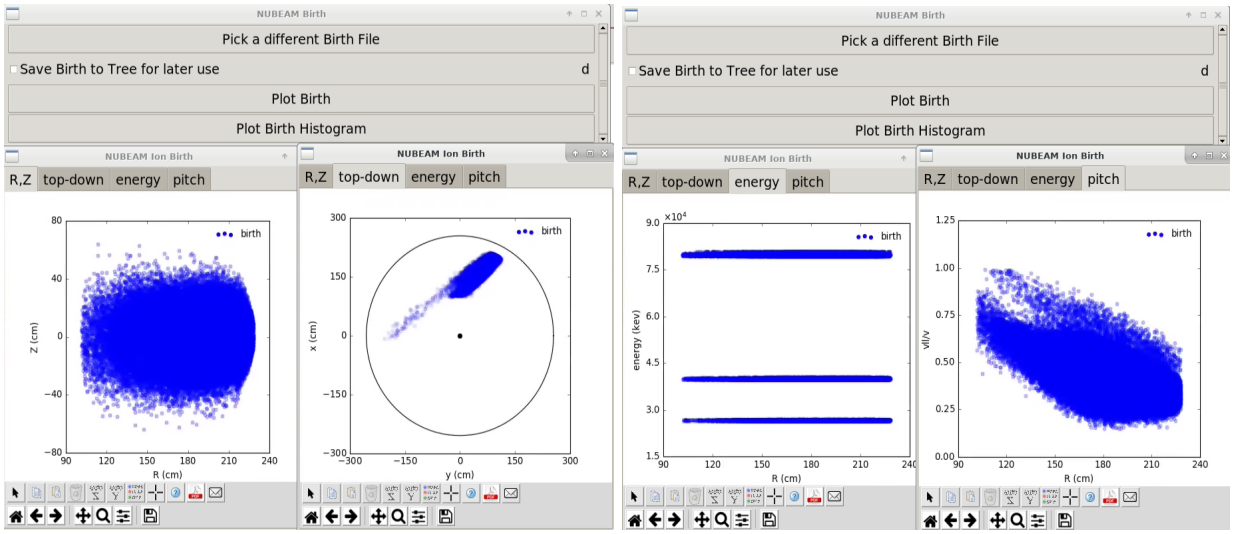
The *_birth.cdf* will be used in the BIRTH module and the *.DATA* will be used in the FBM module.

These two modules can be used within the TRANSP module or standalone.

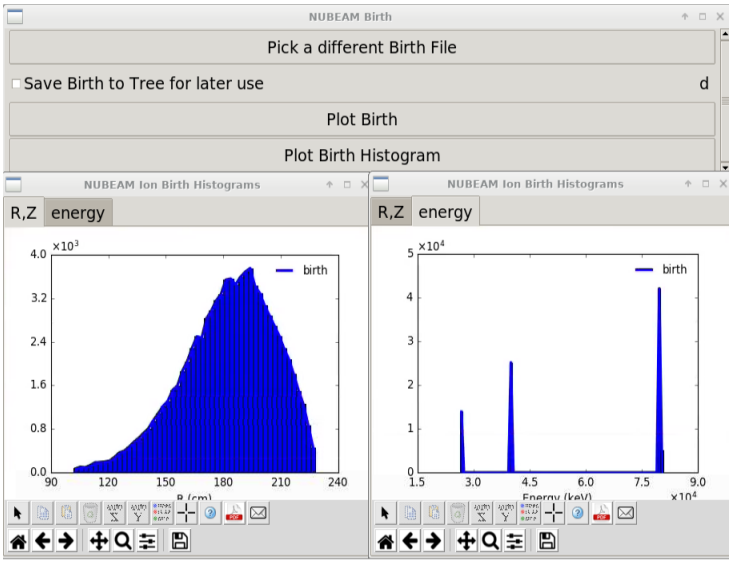
BIRTH module

Double click on the gui, navigate to the appropriate directly and then select your *_birth.cdf* file.

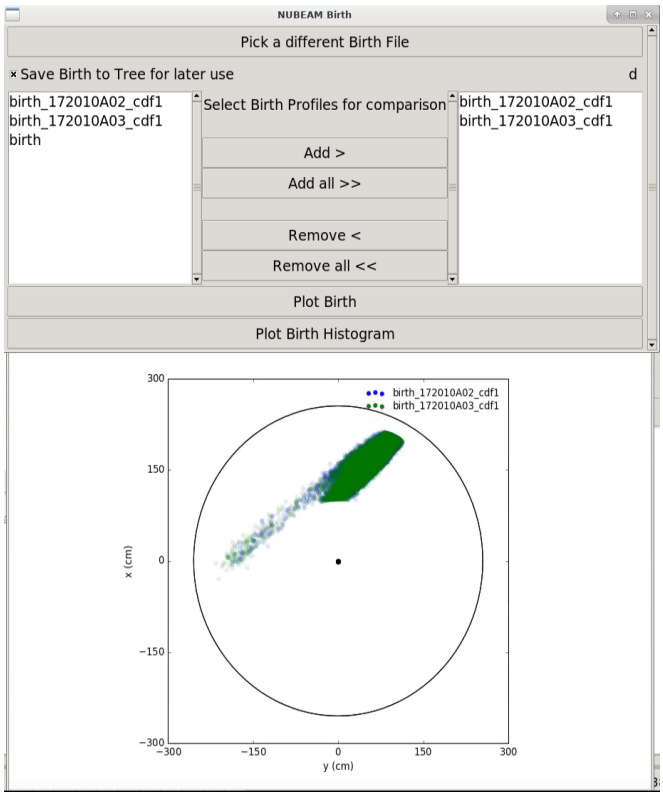
Then click on “Plot Birth”, which will open a figure notebook with an R,Z and top-down view of your birth particles and then the energy and pitch of those particles.



You can also use select “Plot Birth Histogram”, which will give you the vertically binned R and Energy.



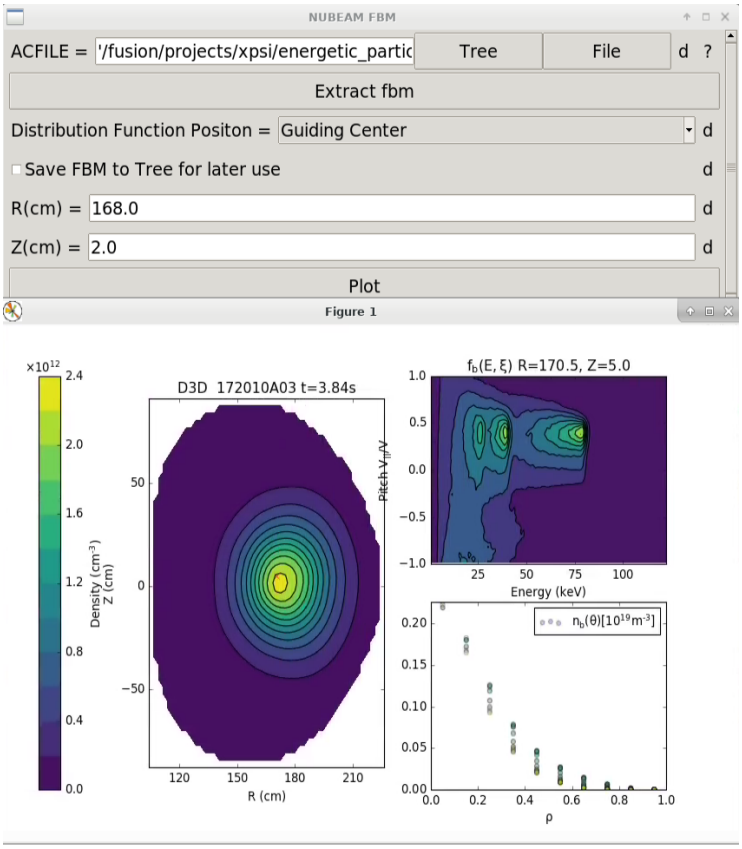
You can then pick a different birth file to plot. It is recommended to save the birth to the tree for later use that way you don’t have to keep reselecting the file. After you have two or more birth files saved in the tree, the gui will change to a list editor and you can then select specific birth files to compare with the plotting, as shown below.



FBM module for fast-ion distribution

Double click on the gui, navigate to the appropriate directly and then select your *.DATA* file and then press extract fbm after selecting the distribution function position. Pressing “Extract fbm” will run the get_fbm code and will take a few minutes.

Then select the R,z position of interest and press “Plot”. This will generate the following plots of fast-ion distribution: (on the left): R, z space, (top right): pitch, energy space and (bottom right): along the midplane in rho.



Similar to the BIRTH module, you can save the FBM to the tree for further use but can only plot one FBM at a time.

Interpretive Transport Analysis: What are the power flows?

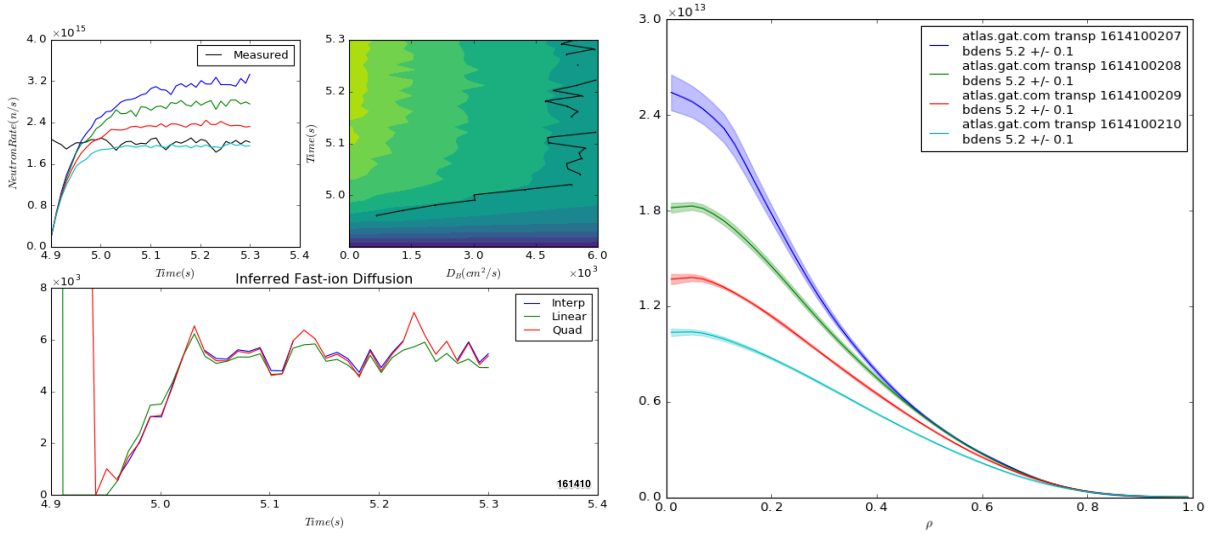
This is the simplest type of transport analysis and there are a few assumptions that come with this question. First the user must provide time histories and plasma profiles. Time histories include the plasma current and field, neutron rate, recycling and heating powers. Plasma profiles include the equilibrium quantities such as q, pressure and diamagnetism, as well as the fluid variables ne, Te, Ti and others. TRANSP places these inputs onto a uniform time grid and steps forward in time computing the heating power injected, time derivatives of the fluid and equilibrium profiles and arriving at the power flows. The power flows are the sources minus the sinks minus the d/dt terms and are known as the “observed” transport power density. In simple form

$$dW/dt + \nabla \cdot Q = S - K$$

Where W is the energy, Q is the flux, S is the source and K is the sink. TRANSP knows W, S and K and computes the divergence of the flux as S-K-dW/dt, which is a profile in time and space.

Need for data consistency

One unique benefit of comprehensive transport analysis is overall data consistency. By this, we mean that the user profile analysis and injected beam power should result in a TRANSP simulation that matches the equilibrium reconstruction stored energy and measured neutron rate. If current diffusion is used the surface loop voltage and internal inductance should also reasonably match indicating a good reconstruction of the plasma average resistivity. In the event of poor matches and evidence of strong MHD and Alfven eigenmodes, one may invoke anomalous fast ion transport via diffusion transport. This is critical to the overall reconstruction of the energy, particle and momentum fluxes provided by the heating beams, as well as the driven current. When the user performs multiple TRANSP simulations scanning the fast-ion diffusion (i.e. 0.2, 0.4, 0.6 m**2/s), we can reconstruct the experimental diffusion via interpolation, shown below. This is available as an RPLOT option, where the first runID is the analysis run, and the subsequent runs increase the input diffusion. The resulting ion energy flux changes by a factor of two for this case, which underscores its importance.



Left: neutron rates and inferred diffusion; Right: Fast-ion density

In order to execute an anomalous fast-ion diffusion scan, there are two options.

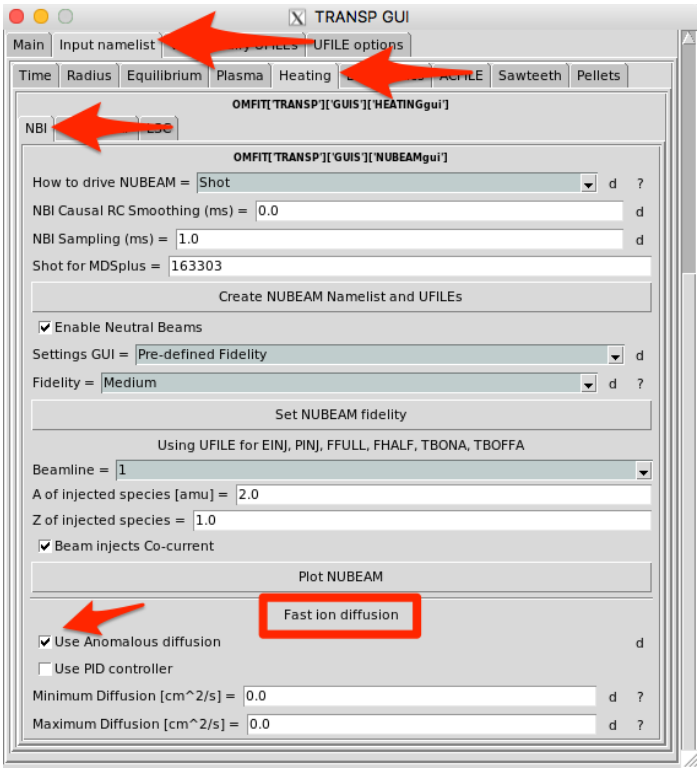
1. Vary the anomalous diffusion “by hand”
2. Use the TRANSP neutron feedback PID controller

Scanning DIFB by hand

After setting up your TRANSP run, move to the tab

Input Namelist → Heating → NBI

and click “Anomalous diffusion”



Here you can set the values for Minimum (`BDIFB`) and Maximum (`CDIFB`) diffusion. They *should be set equal to each other* for a spatially flat diffusion. Otherwise the resulting diffusion will be a profile proportional to the electron particle diffusion coefficient within the min and max bounds.

For example, you set

Minimum Diffusion = 1000.0

Maximum Diffusion = 1000.0

And then run TRANSP.

Then change the numbers, and run again, etc…

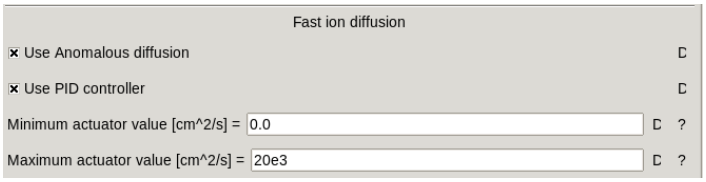
Or you can use TRANSP Variable Scan.

Once this is completed, you can write a UFILE and re-run with a time-dependent DIFB created by linear or quadratic interpolation by using the OMFIT RPLOT tool.

Using the PID controller

Original release notes here.

Essentially, we can have a feedback algorithm adjust DIFB in time so that we don't have to scan it and do the interpolation ourselves. The version sets the minimum (`BDIFB`) and maximum (`CDIFB`) baseline diffusion equal to zero and lets the feedback controller adjust a spatially flat, time dependent anomalous diffusion. Here in the GUI, then minimum actuator value (`FB_NEUTCNTRL_UMMIN`) and maximum actuator value (`FB_NEUTCNTRL_UMAX`) are set instead.



A comparison of the two techniques is given by comparing:

153072Z06, Z07

Current Evolution for kinetic EFIT Reconstruction: What is the resistive current profile?

This type of TRANSP analysis is almost as simple as “interpretive”, with one fundamental difference; The current profile from EFIT is replaced by the current profile from resistive current diffusion. In a kinetic EFIT reconstruction, we specify internal profiles of pressure and current to arrive at a more complete reconstruction than can be accomplished with magnetics and MSE alone. This type of TRANSP run is done via kineticEFITtime OMFIT module.

Let us consider how the equilibrium evolves. We are driving currents from an ohmic transformer. We swing the coil one way in vacuum to charge it, then swing it the other way with a gas puff to cause the plasma breakdown. The current flows on the outside, and penetrates into the core by resistive diffusion.

Input Equilibrium

It is often standard practice to input an equilibrium into TRANSP for time-dependent analysis. This means that we are specifying the boundary shape, internal flux-surfaces and q-profile for all time. This constrains the plasma resistivity as follows, keeping in mind that the resistivity η in a 1D transport simulation is defined as $\langle \mathbf{E} \cdot \mathbf{B} \rangle = \eta \langle \mathbf{j} \cdot \mathbf{B} \rangle$.

TRANSP knows q(r,t) from input. Now, $q \approx rB_\varphi/RB_\theta$ and and B_φ comes from the B-coil that is known to TRANSP, as well as r and R. Therefore, B_θ is defined by q. Now we know **B**(r,t). From Faraday we know $\nabla \times \mathbf{E} = -\partial \mathbf{B}/\partial t$. Therefore, we can integrate $\partial \mathbf{B}(r,t)/\partial t$ to get **E**(r,t) and the loop voltage profile within a constant. The constant comes from the input surface loop voltage $\partial \psi_p(a,t)/\partial t$. We then have **B**(r,t) and **E**(r,t). Now we can take $\eta = \langle \mathbf{E} \cdot \mathbf{B} \rangle / \langle \mathbf{j} \cdot \mathbf{B} \rangle$. In reality, there are other driven currents (beam, wave, bootstrap) so these get taken off when evaluating the ohmic heating power:

$$\eta = \langle \mathbf{E} \cdot \mathbf{B} \rangle \, / \, (\langle \mathbf{j} \cdot \mathbf{B} \rangle - \langle \mathbf{j}_{driven} \cdot \mathbf{B} \rangle)$$

In snapshot ONETWO, E|| is the derived quantity and η = ηNC.

The pitfalls are generally from the fact that q(r,t) must evolve smoothly. It never does from EFIT. All the wiggles contribute to a large variation for each **E** profile and therefore the resistivity will bounce around from timeslice to timeslice. It will be negative some places too.

Resistive Diffusion: Experimental Flux Surfaces

When running in poloidal field diffusion mode (sometimes called MAGDIF with NLMDIF=True), now the plasma resistivity is defined from a model (Sauter, NCLASS) and the q-profile evolution is derived rather than input. The boundary condition for this mode of operation comes from either matching the experimental total plasma current (allowing Vsurf to float) or matching the experimental loop voltage (allowing Ip to float). For experimental analysis we match Ip, which is better known that Vsurf. But for projecting fully non-inductive operation we match Vsurf and set the input Vsurf to zero which will output the total non-inductive plasma current.

Grad-Shafranov Equilibrium: Self-consistent flux surfaces

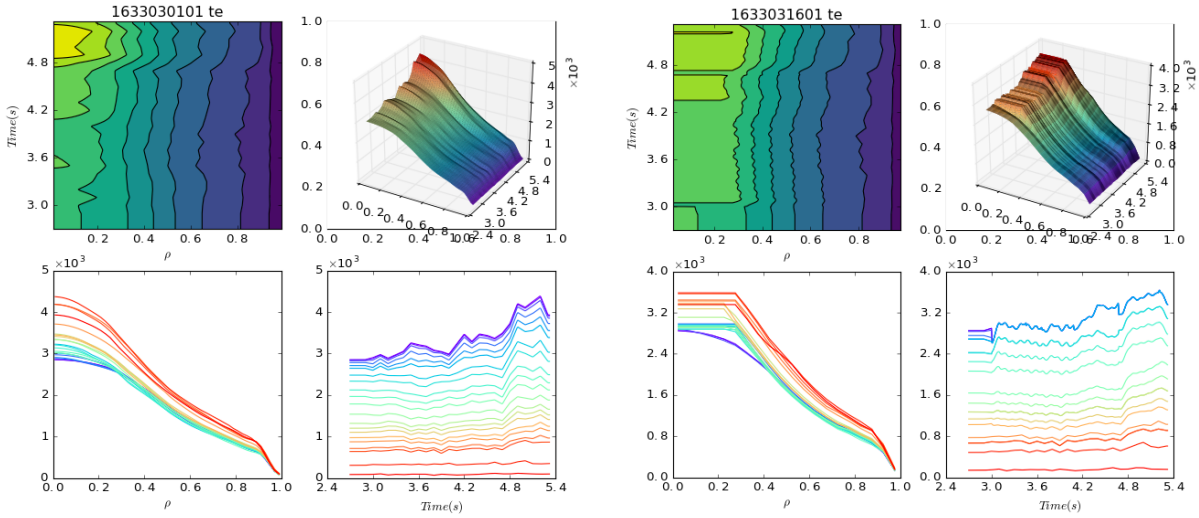
When moving beyond evolving current density (B_θ, q), we can use a solver to solve for ψ given the internal pressure p(ψ) and currents. This is a kinetic equilibrium solution because the kinetic pressure (thermal + fast) and driven current profiles are constrained. This uses either TRANSP version of TEQ or the free-boundary solver ISOLVER.

Predictive TRANSP

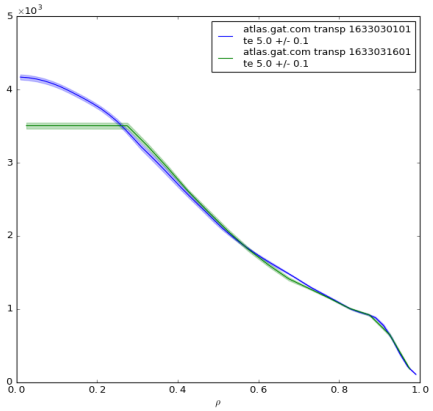
Transport model validation: How well does transport model “X” compare to experiment?

This type of TRANSP run replaces one of the fluid variables (Te, Ti, ne, Rotation) with the prediction from a transport model (GLF23, MMM, TGLF, etc…). The prediction of the profile (i.e. Te) comes from simple Fick’s law type of equation where $q = - n \chi \partial T/\partial r$. TRANSP knows q from source-sink, and the model provides q given $\partial T/\partial r$, so the temperature profile is iterated until they match in a *transport solver*. This provides a new temperature profile, i.e. the “prediction”. In predictive TRANSP, the solver is called PT_SOLVER and has its own switches and options. The PT_SOLVER is a time-dependent transport solver and has similarities to XPTOR, or in the steady-state, TGYRO.

Note that, in general, the solution of stiff transport equations is very challenging, and the user must be aware that interaction with the TRANSP developer team is generally necessary to verify that a given predictive run is well converged.



Example of (left) experimental and (right) predicted Te using GLF23



Overlay of profiles displaying reasonable agreement in the confinement region and impact of sawtooth model (flat Te)

TRANSP Environment Variable Requirements

portal.pppl.gov

If you have access to PPPL portal, TRANSP runs can be submitted directly without fusion grid access. The following commands need to be executed first:

```
module load intel/2019.u3 ntcc
```

iris.gat.com

Open your $HOME/.bashrc file and add the following lines

```
# TRANSP variables for DIII-D
Export TRANSPGRID_SERVER=”transpgrid.pppl.gov”
export MDS_TRANSP_SERVER="atlas.gat.com"
export MDS_TRANSP_TREE="transp"
export TR_EMAIL=”[whatever email you used for globus]”
export PREACTDIR="/dev/null"
export ADASDIR="/dev/null"
```

On GA iris you need access to the TRANSP project area `/fusion/projects/codes/transp` and make your username directory there.
To access the TRANSP utilities use

```
irisa: module load ntcc
```

And then you can use `trdat, tr_start, tr_send`

Submitting TRANSP runs from the command line

portal.pppl.gov

If you have access to PPPL portal, TRANSP runs can be submitted directly without fusion grid access. The following commands need to be executed:

Type `module load ntcc`
Type `tr_start` and answer the questions
Type `tr_send`

iris.gat.com

On GA iris go to your TRANSP user area

```
/fusion/projects/codes/transp/$USER
```

You must be starting from an existing TRANSP run in a sub-directory of the above
Type `module load ntcc`
Go to the run directory and do the following

1. If you are modifying the namelist and making a new run, then copy the namelist to a new file (i.e. `cp 999999Z01TR.DAT 999999Z02TR.DAT`), otherwise if you are submitting a prepared or failed run, then use the existing namelist.
2. If you copied the run into a new directory, modify the namelist so that INPUTDIR points to the new directory.
3. Type `tr_start [runid]` and answer the questions
4. Type `tr_send [runid]` and you may need to enter your password for the fusion grid

Example session follows for submitting a prepared or failed run:

```
module load ntcc
cd /fusion/projects/codes/transp/$USER/[runid]
tr_start [runid]
```

```
  [may ask for number of NUBEAM cores, use NPTCLS/4000]
  [tok.yr] y
  [email] y
  [mdsplus] y
  [send as tar] y
tr_send [runid]
```

TRANSP Access via FusionGrid

Requesting access

First email transp_support@pppl.gov to request a transp_only account and cc the TRANSP responsible officer at DIII-D (Shaun Haskey shaskey@pppl.gov), and include the name of somebody who can help you learn TRANSP. After the account is created, follow the instructions here: https://transp.pppl.gov/cilogon.html

FAQ

How do I get the namelist from a previous TRANSP run?

Use the OMFIT module “Load Existing” tab, and “Load Namelist from MDS+”.

How do I put a TRANSP netCDF file into MDSplus?

Get the .CDF file on GA workstation (via tr_fetch or scp).
It can be written to mdsplus from the linux command line provided the necessary TRANSP environment is already set up, for example by 'source'ing a suitable login with your TRANSP settings and `module load ntcc`. The command is:

mdsplot t s atlas.gat.com t transp n [transp shot] q [CDF file w/o .CDF]
i.e.
mdsplot t s atlas.gat.com t transp n 1633032601 q 163303Z01

For this to work, the TRANSP tree for shot number 1633032601 needs to exist in mdsplus. Unless this was a failed TRANSP run, it most likely doesn't exist. This can be remedied within IDL using mdstcl commands, eg
IDL>mdsconnect, 'atlas.gat.com'
IDL>mdsopen, 'transp', -1

```
IDL>mdstcl, 'create pulse 1592430801'
IDL>mdsput,'.device','"D3D"'
IDL>mdsclose
IDL>mdsdisconnect
```

Where there can be the error that the tokamak device can be named the runID, rather than “D3D”.

How do I look at a crashed TRANSP run?

In the TRANSP Grid Monitor it will probably show this: 

and you can LOOK at it.
Use the OMFIT TRANSP module’s RPLOT tab, and enter
Servers = [‘transpgrid.pppl.gov’]
Trees = [‘trlook_d3d’]
Runids = your runid as normal.
Be sure to un-check “Use cache” if you’re repeating a run.

What are the sign conventions?

There is a list of four TRANSP runs to cross-check the sign conventions.
155407A06 (default), A08(reverse Ip), A09(reverse Bt), A10(reverse Ip and reverse Bt)

GEQDSK and EFIT
In the GEQDSK files we have the following changes when going from standard to the other configurations
standard → rev-Ip has g.cpasma change sign
standard → rev-Bt has g.bcentr and g.fpol change sign
standard → rev-Ip and rev-Bt has g.cpasma, g.bcentr and g.fpol change sign
It looks like prirz does not change, and g.ssibry and g.ssimag are always negative and increasing with ψN. This is consistent with what Ron Prater and Lang Lao had told me some years ago.

GADAT and AEQDSK
From GADAT and AEQDSK files we have the following
IP: Plasma current, which changes sign with the direction Ip.
PSIBDY: Poloidal flux at the boundary, used to calculate the surface voltage (-2π dψbdy/dt), which changes sign with the direction of Ip.
RBCENT * BCENTR: Vacuum major radius (=1.69550) times vacuum toroidal field at this radius, where BCENTR changes in time, and with sign with the direction of Bt.

DIAMAG: Diamagnetic loop, which will (for the same plasma) change sign with the direction of Bt.

What TRANSP wants

IP: Plasma current history. This is enforced to be positive in TRANSP and the input UFILE data is changed internally to conform to this, and the output will also reflect this. TRANSP's decision to flip this or not is not kept, but the input sign convention (NLJCCW) will be enforced for calculations of things like Er.

RBT: History of vacuum major radius times vacuum field at this radius. This is enforced to be positive *but* TRANSP keeps the sign and applies it to the diamagnetic loop input data.

What we give TRANSP

In TRANSP we currently provide the following info:

1D UFILEs:

IPL: Plasma current, which we input as it is returned from GADAT.

VSF: Surface voltage which (should) flip signs when we reverse the plasma current

RBT: R*Bcenter which flips signs when we reverse the toroidal field

DFX: Diamagnetic flux

scrunch2:

GRB which is the EFIT "g" function R*BT. However in the GRB UFILE the sign never changes, just the associated scalar fields that indicate direction of Ip, Bt.

PLF which is poloidal flux

TRF which is toroidal flux

Using scrunch2 we see the following:

The sign of the UFILEs never changes

The associated scalars do change signs

TRANSP Output

We expect that the plasma current PCUR will be positive and the same ✓

We expect that the surface voltage should always be positive during q evolution ✓

We expect that the vacuum BZ*R will be the same ✓

We expect that the input and computed diamagnetic flux DFLXM, DFLUX will be ≈ the same and negative (plasma is diamagnetic, Bt reduced compared to vacuum by plasma) because this carries the sign of Bt ✓

We expect that the toroidal rotation will flip sign (due to input) for reverse Ip ✓

We expect that Er in the high rotation ordering (VRPOT) will be identically flipped for reverse Ip but unchanged for reverse Bt ✓

We expect that the neoclassical poloidal flow will generally flip sign with the toroidal field (but change subtly for rev-Ip due to change in beam dilution) ✓ Note: TRANSP enforces Bt>0 so the sign will be "wrong" from DIII-D convention.

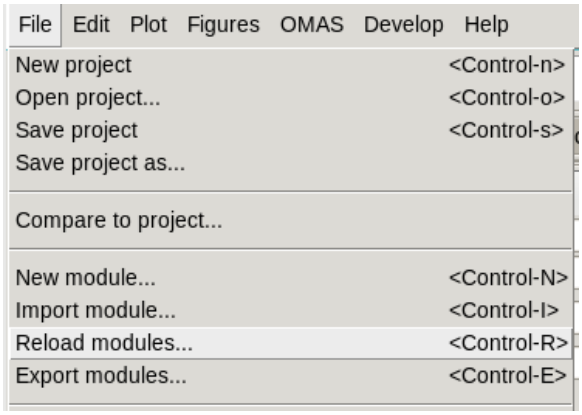
We expect that the beam ion density will be shifted off axis for reverse Ip ✓

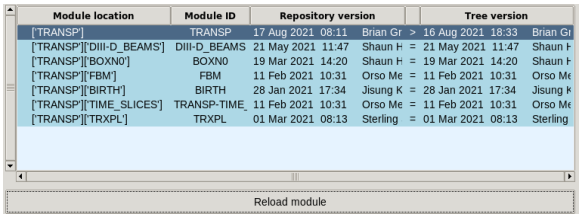
We expect that the beam ion current drive will be flipped for reverse Ip ✓

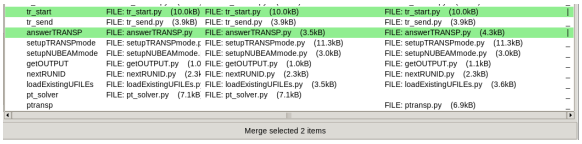
OMFIT says says python2.7 not found. How do I fix?

Around Feb 2021 OMFIT upgraded from python2.7 to python3. This caused an issue with the script that answers the TRANSP command prompt questions to run tr_start. See commit here. This can be fixed by re-loading the module.

1. Go to File → Reload modules
2. Select “TRANSP” and click “Reload module”
3. Find the scripts TRANSP[‘SCRIPTS’][‘answerTRANSP’] and TRANSP[‘SCRIPTS’][‘tr_start’]
4. Use the spacebar to toggle the merge of scripts from the repository into your project.
5. Click the button to “Merge 2 selected items”
6. Then try to start your TRANSP run again







How do I configure TRANSP for a mixed H/D isotope plasma?

When attempting to create a TRANSP run from a DIII-D experiment that has a mix of deuterium and hydrogen beams you may encounter this error:

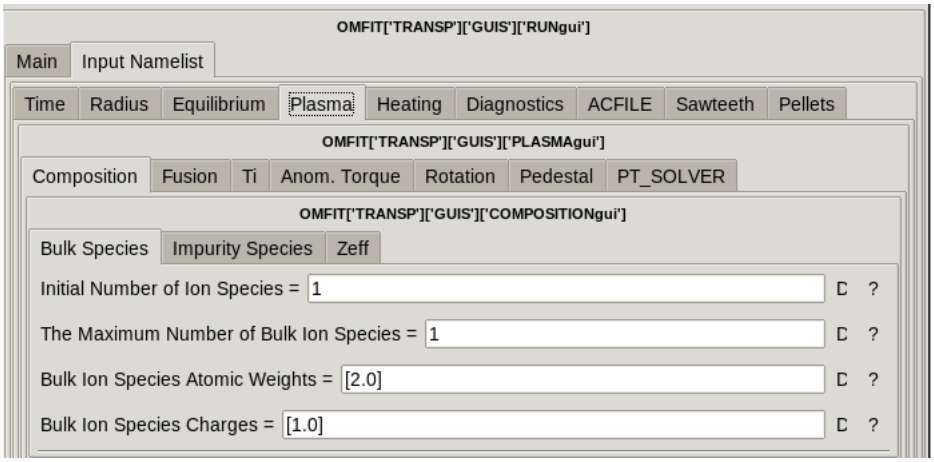
?DATCKB-- HYDROGEN BEAM MODEL NEEDS HYDROGEN THERMAL PLASMA

The reason is that the default for DIII-D is deuterium plasma, and if hydrogen will be injected you must tell TRANSP beforehand how many ions are in a given run.

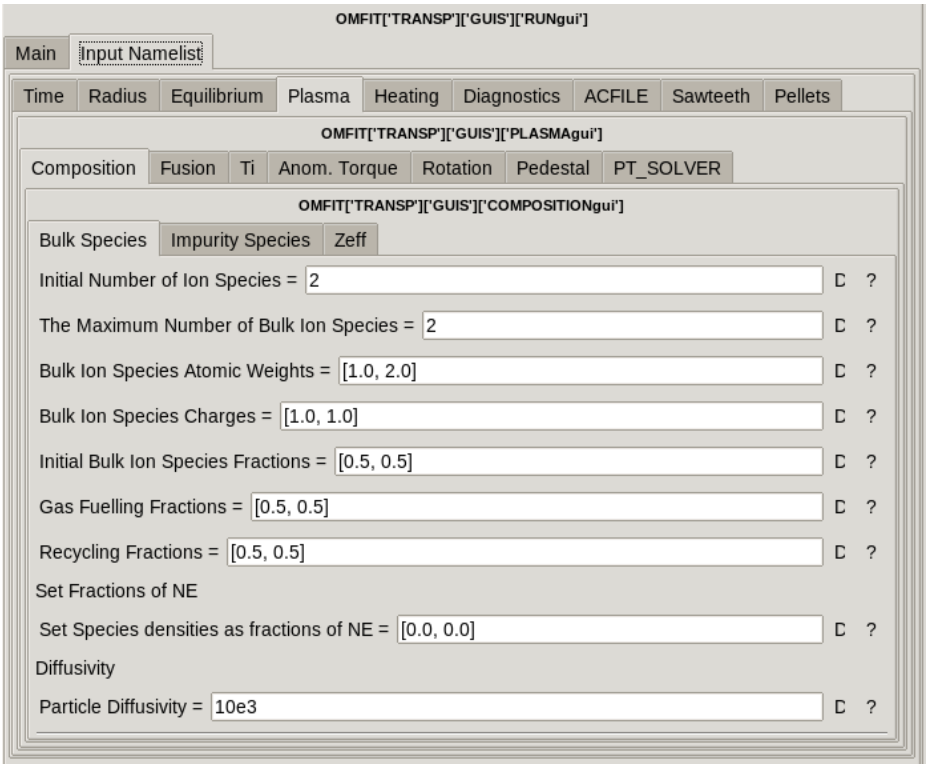
The things to check are

- Are there both hydrogen and deuterium beams being injected with non-zero power?
    - If yes, then continue
    - If no (i.e. one beam is hydrogen gas but does not inject), then change that beam to be deuterium in the NUBEAM namelist
- In order for TRANSP to compute particle balance, you must prepare your run to have the following features, which can be set in the GUI
    - Multiple isotopes of hydrogen defined
    - Initial isotopic ratio at the beginning of the run set
    - Recycling fractions of the hydrogenic species configured

These are all configured in the GUI as displayed below



Default single hydrogenic species (D) plasma



Modified for 50/50 mix of H, D

This will produce a mixed H/D plasma but the composition in time will depend on the amount of particle source from core NBI fueling.
To make i.e. hydrogen dominant, increase the fractions to i.e. [0.99, 0.01] where the first value corresponds to the first species mass (here 1.0=H).
This should be set based on what gasses were used in the experiment.
Beyond this, one must assess the agreement between stored energy and neutrons to determine if the isotopic mix is reasonable. Please work with a TRANSP expert at DIII-D to go beyond this simple setup.

How do I retrieve the TRANSP output for further analysis

Often TRANSP is just a step in an analysis pipeline. Once you've run TRANSP and you wish to continue your work, you will need to pull the output files from transp over from the PPPL cluster. To pull these files back to IRIS go to the directory where the run was created (i.e /fusion/projects/codes/transp/haskeysr/164436Z19_tmp/ then:

```
module load ntcc
grid-proxy-init
~~tr_fetch -nowait D3D.[year] [shotid] # (not working 09/23)~~
tr_fetch [shotid] D3D # i.e tr_fetch 164436Z19 D3D
```

Where [year] is the year of the shot, and [shotid] is the TRANSP shot ID. For example, if I pull files from my TRANSP run into /path/185781

```
[prechelg@iris185781]$ module load ntcc
[prechelg@iris185781]$ grid-proxy-init]
Your identity: /DC=org/DC=cilogon/C=US/O=University of California-Irvine/CN=Garrett Prechel A45253321
Creating proxy ..... Done
Your proxy is valid until: Wed Nov 3 11:09:28 2021
[prechelg@iris185781]$ tr_fetch -nowait D3D.20185781Z02
Looking for files on TRANSP host
Retrieving files from TRANSP host
(files will be transferred over)
Files retrieved
```

Do note that after you retrieve these files from PPPL, they will be deleted from the PPPL cluster after some relatively short timescale so if you need them for analysis you'll want to hang onto the copy you now have on IRIS.