PEdestal Neural Network (PENN) in ETS v5

Previously called paraPED

Andreas Gillgren
Chalmers University of Technology
16 September 2020

andreas.gillgren@chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY

Outline

- Database
- Training Neural Networks
- Predictions in ETS
- Error estimation
- Electrons/Ions
- Test case
- Future plans

Database

We have a pedestal database from JET (*EUROfusion JET pedestal database, created by Lorenzo Frassinetti*)

- H-mode plasmas
- ~ 2000 data entries (shot 73342 - shot 92489)

MAIN PEDESTAL PROFILE PARAMETERS

NB: the IMAS column refers to the equivalent quantity in the IMAS Data Dictionary. This quantity is in the SUMMARY IDS (used for cataloguing data in an SQL database), otherwise specified.

BRIEF DEFINITION	details	UNIT	IMAS name
Separatrix T_e	AUG: T_e^{sep}=100eV JET: T_e^{sep}=100eV MAST: T_e^{sep}=50eV TCV: T_e^{sep} : 50eV	eV	local/separatrix/t_e
Separatrix n_e (mtanh)	Separatrix density. Estimated assuming $T(r_{sep})=T_{sep}$ using mtanh fit	m^{-3}	pedestal_fits/mtanh/n_e/separatrix
Separatrix p_e (mtanh)	Separatrix pressure. Estimated assuming $T(r_{sep})=T_{sep}$ using mtanh fit	Pa	pedestal_fits/mtanh/pressure_electron/separatrix
T_e^{ped} (mtanh)	T_e pedestal height using mtanh fit	eV	pedestal_fits/mtanh/t_e/pedestal_height
n_e^{ped} (mtanh)	n_e pedestal height using mtanh fit	m^{-3}	pedestal_fits/mtanh/n_e/pedestal_height
p_e^{ped} (mtanh)	Pe pedestal height from the mtanh fit	Pa	pedestal_fits/mtanh/pressure_electron/pedestal_height
w_{Te} (mtanh)	Te pedestal width in ψ_N space using mtanh	ψ_N	pedestal_fits/mtanh/t_e/pedestal_width
w_{ne} (mtanh)	ne pedestal width in ψ_N space using mtanh	ψ_N	pedestal_fits/mtanh/n_e/pedestal_width

L. Frassinetti Page **1** of **13**

GLOBAL PARAMETERS and ISOTOPES

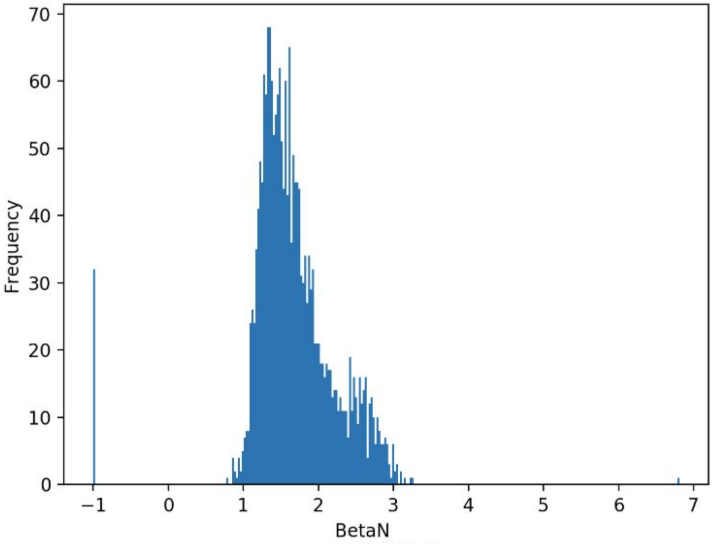
IDL DB PARAMETER	DEFINITION	UNIT	IMAS name
I_p	Plasma current	A	global_quantities/ip
B_t	Toroidal field	T	global_quantities/b0
q_{95}	q95	Adimensional	global_quantities/q_95
R	Major radius	M	global_quantities/r0
a	Minor radius	M	boundary/minor_radius
P_{NBI}	NBI power	W	heating_current_drive/power_nbi_delivered (nominal NBI power delivered from the beams)
P_{ICRH}	ICRH power	W	heating_current_drive/power_ic_delivered
P_{ECRH}	ECRH power	W	heating_current_drive/power_ec_delivered
P_{Ω}	Ohmic power	W	global_quantities/power_ohm
P_{tot}	Total power – NBI shine through – dW/dt	W	global_quantities/power_steady
P_{rad}	Radiated power	W	global_quantities/power_radiated
Vol^{tot}	Total volume	m^3	global_quantities/volume
W_{MHD}	MHD energy	J	global_quantities/energy_mhd volume integral of the pressure, with pressure determined by EFIT

Database

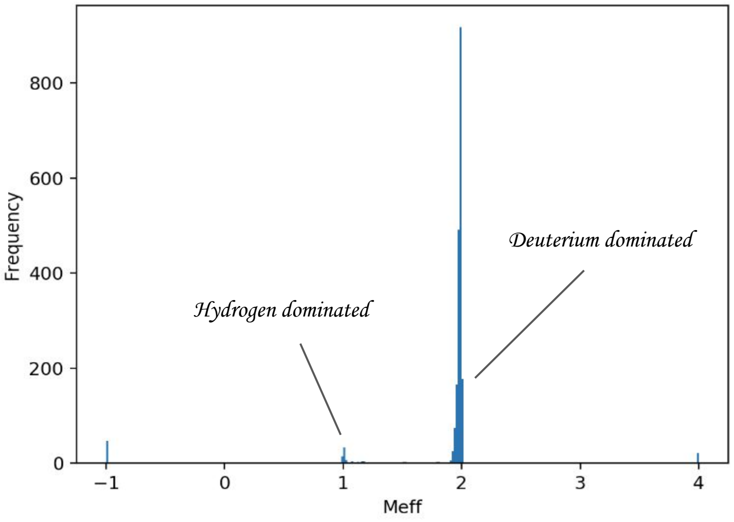
Filter data before training neural network

- Look at parameter distribution
- Exclude data with placeholders (-1 for missing data)
- Also important for choosing parameters

BetaN



Effective Mass



Database

- Choice of parameters

Input parameters

Beta_N (MHD)

I_p (plasma current)

B_0 (toroid field)

R_0 (major radius)

a (minor radius)

Elongation

Upper triangularity

Lower triangularity

P_tot (total power input)

q95

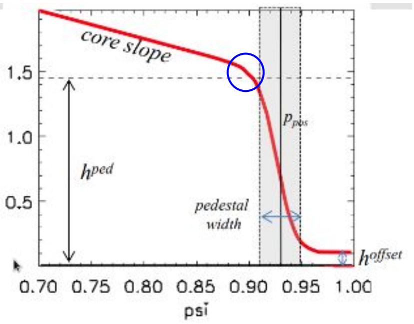
Plasma volume

Neural Networks

Outputs (predictions)

Pedestal temperature (height)

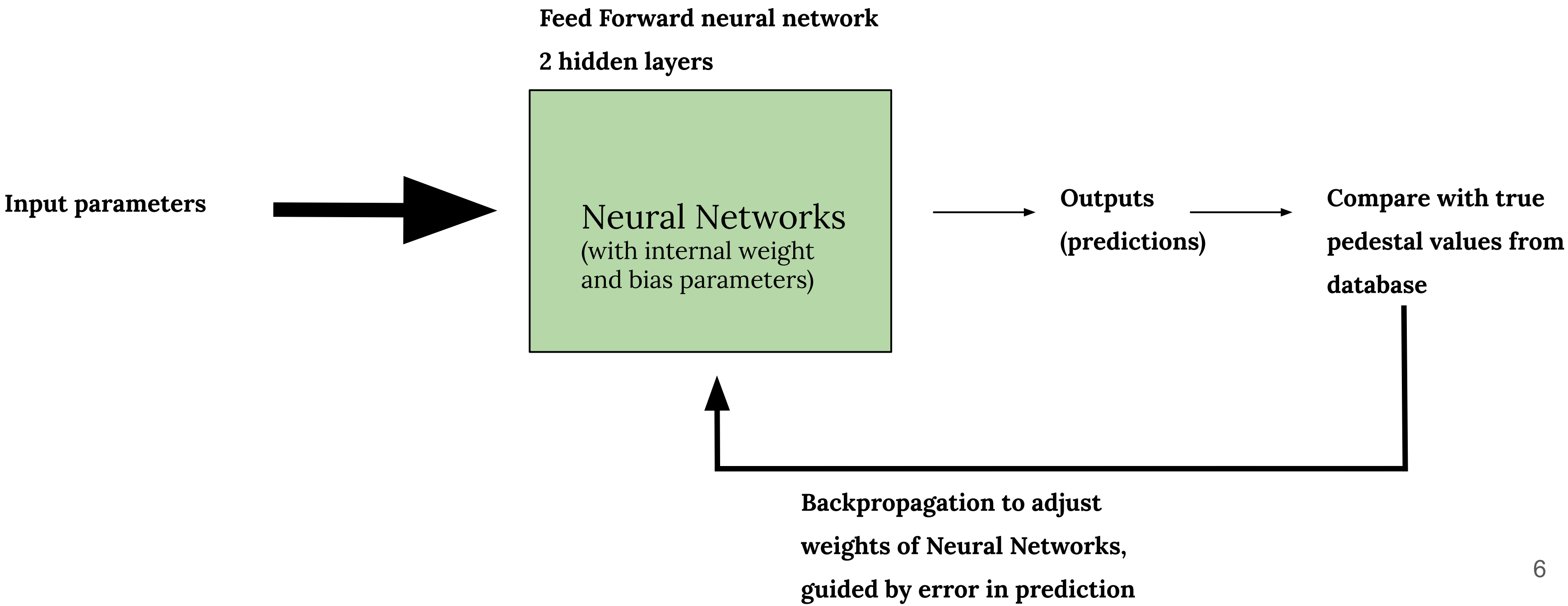
Pedestal density (height)



**We train 2 separate neural networks, both with the same inputs, but with different outputs*

Training Neural Networks

We feed the ~ 2000 data entries through the Neural Networks



Predictions in ETS

- Once the Neural Networks are trained, we transfer the optimized weights to ETS
- This allows for quick predictions since the training needs to be done only once before the actual implementation
- In RC version, PENN will be available in next tag

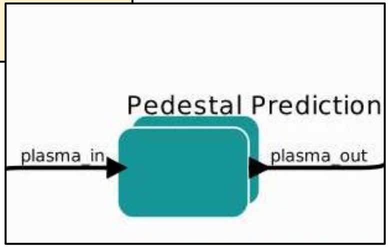
Local CPU/GPU

- Filter data
- Training

Neural Network weights →

<u>ETS v5</u>

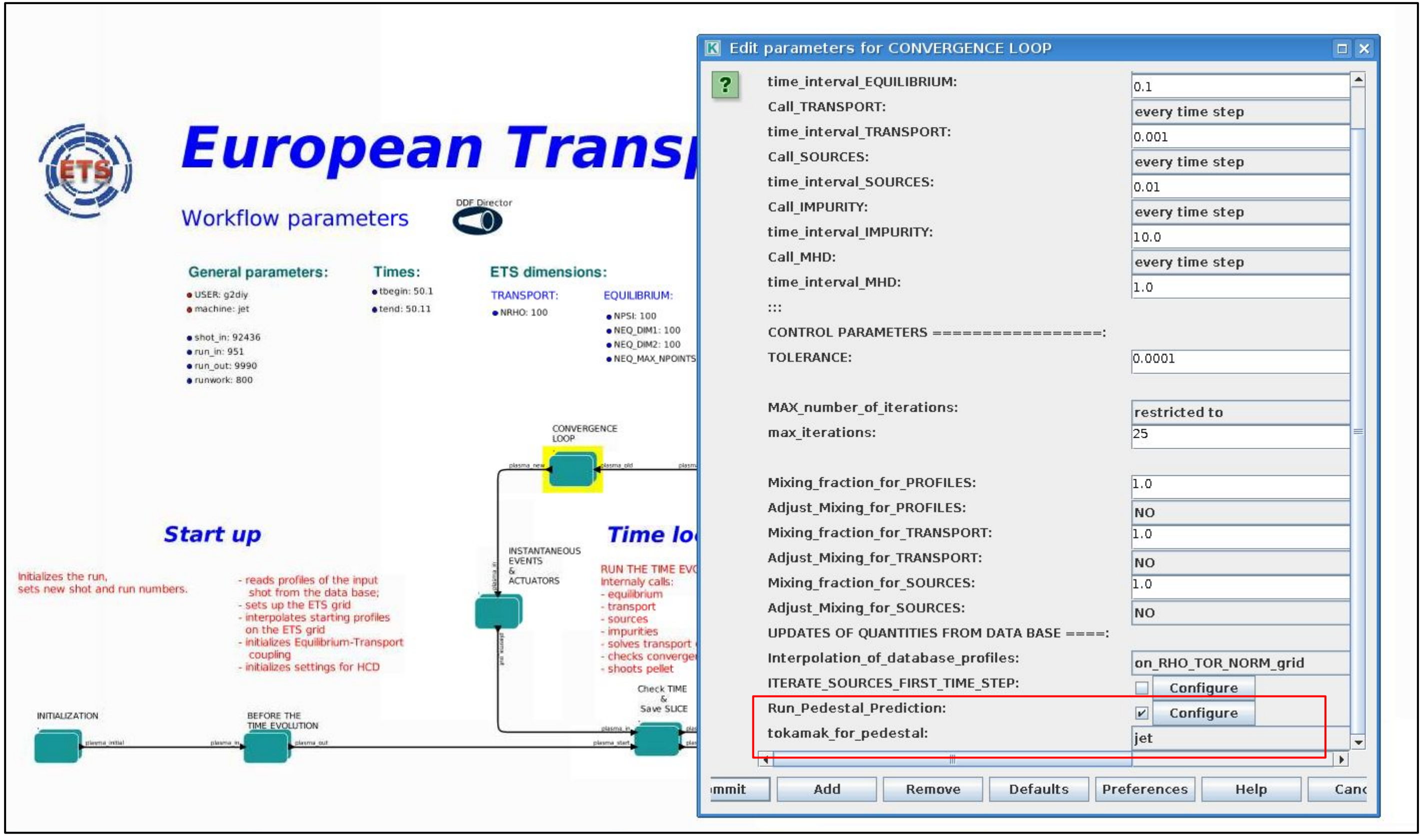
- Python script in kplots
- pedestal actor in Convergence loop/update plasma
- Instead of inputs from pedestal data base, we import the parameters from CPOs: coreprof, equilibrium, etc...



**On our local CPU/GPU, we use machine learning packages such as Tensorflow and Keras, but we do not need the packages in ETS since the training is already complete (the prediction part post training is straightforward to code compared to the training)*

Predictions in ETS

Activate PENN in convergence loop actor



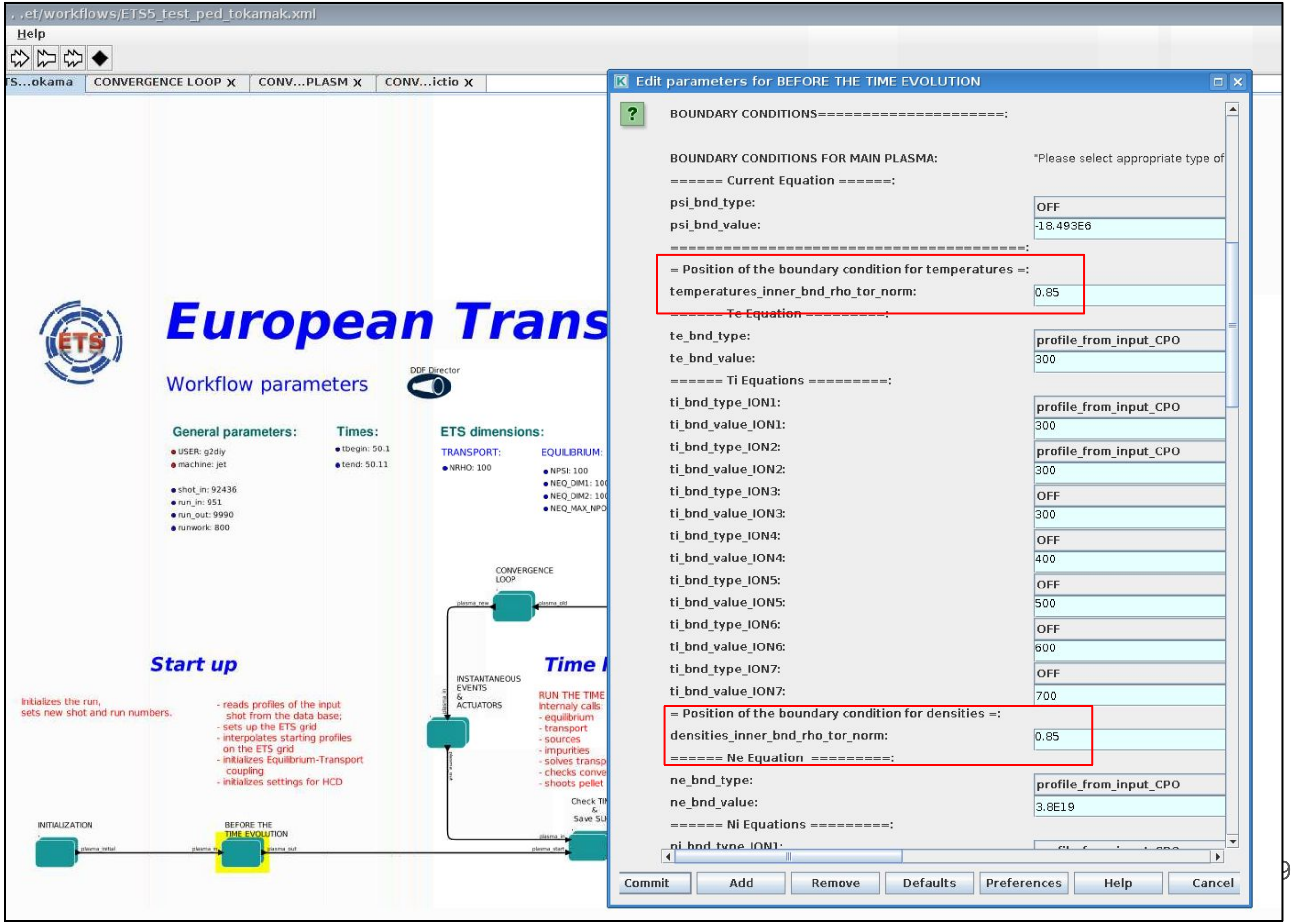
Predictions in ETS

Also need to adjust position of boundary condition in “before the time evolution”

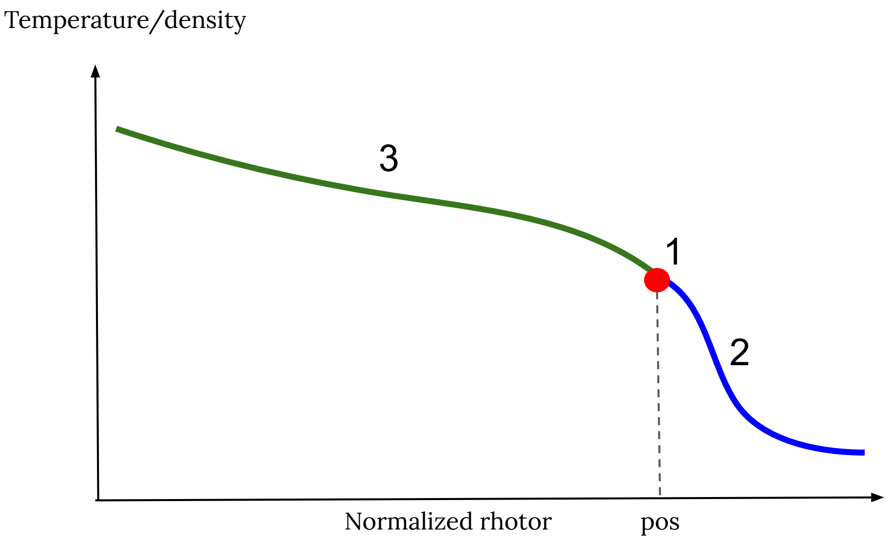
For now, the position has to be the same for temperature and density. We are working on having different positions as an option

For predictive runs, change “profile_from_input CPO” to something else, for instance, value

Now, we are ready to run it!



Predictions in ETS



1. The predictions are made for the pedestal top and new boundary conditions are set

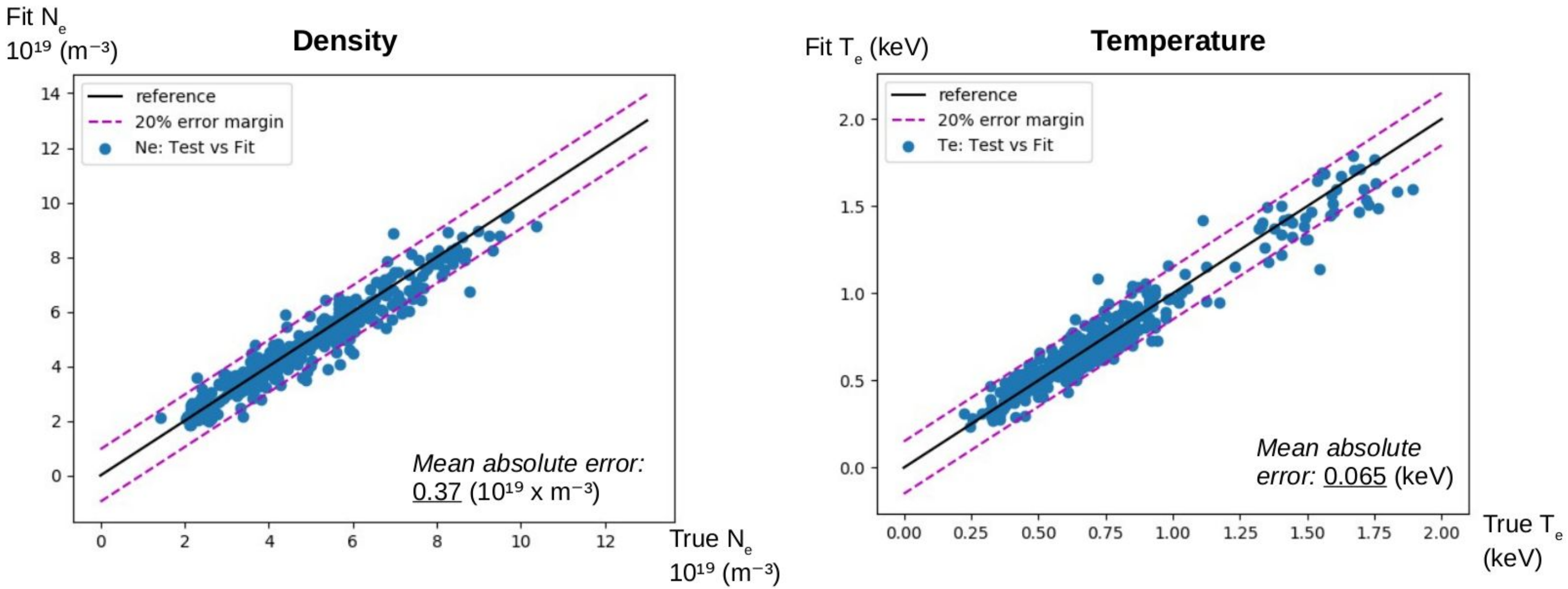
2. The region between the pedestal top and last closed flux surface is adjusted using tanh function to ensure continuity (this is our best current method, but other options might be available later)

3. The inner part is calculated using transport equations with the new boundary condition

** A new prediction is made at the first iteration in the convergence loop for each new time step (this is adjustable)*

Error estimation

To estimate error/accuracy of the model, we can exclude some data during training phase and test towards it (test set)



Error estimation

Another approach to estimate uncertainty: train several neural networks on same data and compare predictions! (different initial weights will give slightly different neural networks)

Beta_N (MHD)

I_p (plasma current

B_0 (toroid field)

R_0 (major radius)

a (minor radius)

Elongation

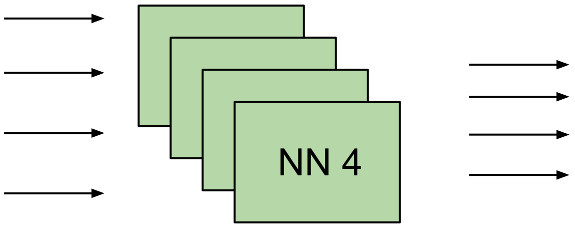
Upper triangularity

Lower triangularity

P_tot (total power input)

q95

Plasma volume



Temperature 1
Temperature 2
Temperature 3
Temperature 4

Final prediction:
mean temperature

Relative uncertainty/error:

$$e_{rel} = \frac{\sqrt{a}}{|V|}$$

where ***a*** is the standard deviation of the predictions and **V** is the mean value

**We currently train 10 Neural Networks for each quantity*

Same method as in “*Self-consistent core-pedestal transport simulations with neural network accelerated models*”, O.Meneghini, *et al.* 2017

Electrons/Ions

Only electrons in database -> We can only predict electron temperature and density

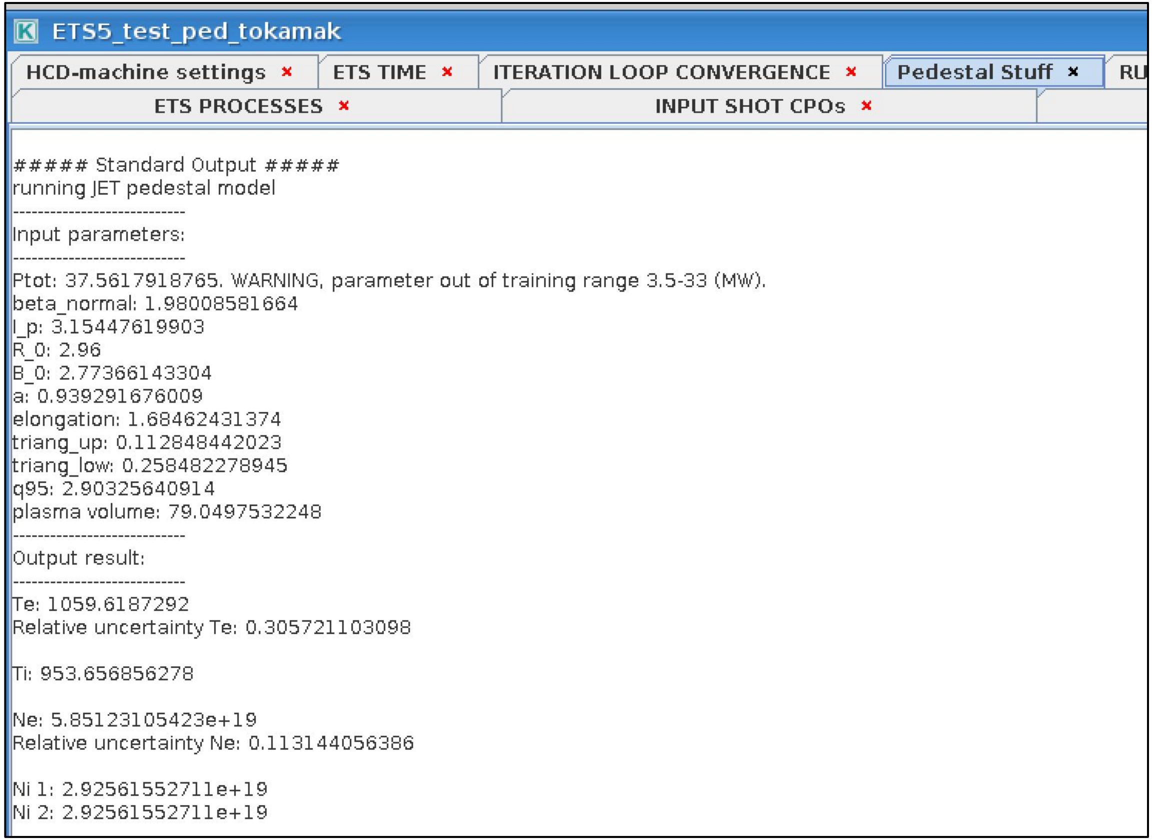
Current solution for Ion temperature: $\frac{T_i}{T_e} = \tau$ (adjustable in pedestal actor, currently set to 0.9)

For ion density, we use the initial ratio between the densities of the ion species, then we enforce quasineutrality with this ratio (since we know the electron density from predictions). This quasineutrality is not exact since we do not include, for instance, impurities.

Test case (prerequisites)

When testing PENN, it is an advantage if the input parameters are within the training range (otherwise, the user will get a warning in the interaction tab)

The database consists of mainly deuterium plasma, we cannot expect great accuracy when looking at other cases



Test case

Here, we start with initial temperature profiles from CPO, then evolve it in time using PENN

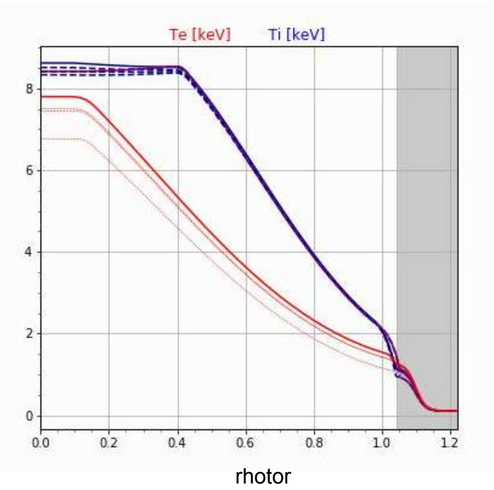
With new boundary conditions, the red profile (electron temperature) shifts upwards

We have some non-smooth behaviour at the pedestal top. A remedy for this would be to make the outer tanh part actively fit the derivative of the inner profile

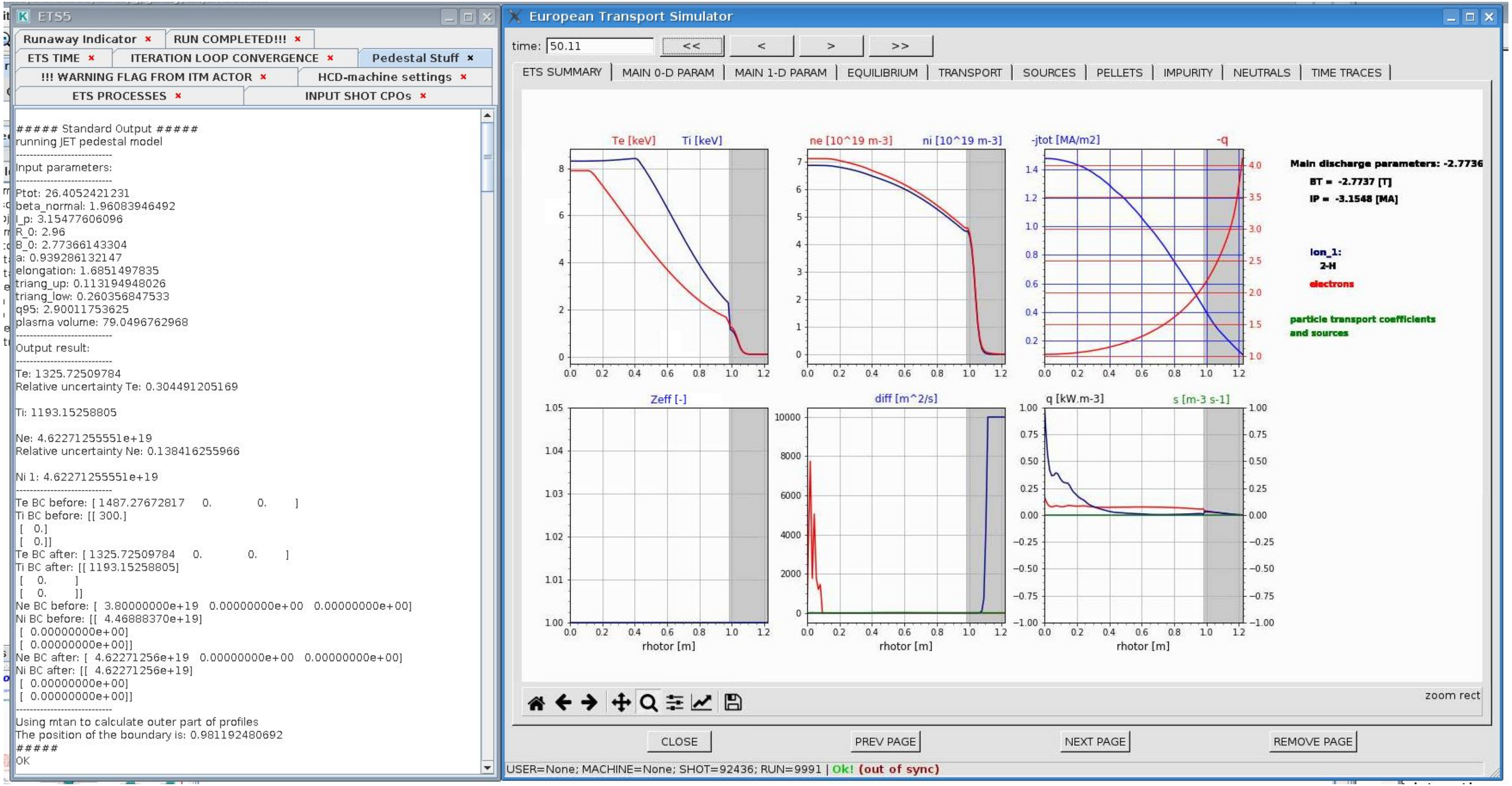
However, this is not straightforward since the inner part is calculated with the transport equations after the outer part is already set

tokamak = jet

shot = 92436



Test case


ETS5
Runaway Indicator
RUN COMPLETED!!!
ETS TIME
ITERATION LOOP CONVERGENCE
Pedestal Stuff
!!! WARNING FLAG FROM ITM ACTOR
HCD-machine settings
ETS PROCESSES
INPUT SHOT CPOs
Standard Output
running JET pedestal model
Input parameters:
Ptot: 26.4052421231
beta_normal: 1.96083946492
I_p: 3.15477606096
R_0: 2.96
B_0: 2.77366143304
a: 0.939286132147
elongation: 1.6851497835
triang_up: 0.113194948026
triang_low: 0.260356847533
q95: 2.90011753625
plasma volume: 79.0496762968
Output result:
Te: 1325.72509784
Relative uncertainty Te: 0.304491205169
Ti: 1193.15258805
Ne: 4.62271255551e+19
Relative uncertainty Ne: 0.138416255966
Ni 1: 4.62271255551e+19
Te BC before: [1487.27672817 0. 0.]
Ti BC before: [[300.]
[0.]
[0.]]
Te BC after: [1325.72509784 0. 0.]
Ti BC after: [[1193.15258805]
[0.]
[0.]]
Ne BC before: [3.80000000e+19 0.00000000e+00 0.00000000e+00]
Ni BC before: [[4.46888370e+19]
[0.00000000e+00]
[0.00000000e+00]]
Ne BC after: [4.62271256e+19 0.00000000e+00 0.00000000e+00]
Ni BC after: [[4.62271256e+19]
[0.00000000e+00]
[0.00000000e+00]]
Using mtan to calculate outer part of profiles
The position of the boundary is: 0.981192480692
#####
OK
European Transport Simulator
time: 50.11
<<
<
>
>>
ETS SUMMARY
MAIN 0-D PARAM
MAIN 1-D PARAM
EQUILIBRIUM
TRANSPORT
SOURCES
PELLETS
IMPURITY
NEUTRALS
TIME TRACES
Te [keV]
Ti [keV]
ne [10^19 m-3]
ni [10^19 m-3]
-jtot [MA/m2]
-q
Zeff [-]
diff [m^2/s]
q [kW.m-3]
s [m-3 s-1]
rhotor [m]
Main discharge parameters: -2.7736
BT = -2.7737 [T]
IP = -3.1548 [MA]
Ion_1:
2-H
electrons
particle transport coefficients
and sources
zoom rect
CLOSE
PREV PAGE
NEXT PAGE
REMOVE PAGE
USER=None; MACHINE=None; SHOT=92436; RUN=9991 | Ok! (out of sync)

Future plans

Apart from features regarding PENN, we work on also making it available for ETS v6

- TCI
- Fortran version

The main job is to adjust how inputs and outputs of the model are handled

Pedestal Neural Network (PENN) - User Manual

Andreas Gillgren

Chalmers University of Technology

1 Introduction

Thanks for listening!

Questions?