Status of AToM project and plans/ opportunities for BOUT++ utilization

C. Holland, UCSD for AToM team

J. Candy, D. Bernholdt, D. Green, M. Dorf, D. Schissel, D. Batchelor, S. Diem, V. Izzo, O. Meneghini, D. Orlov, E. D'Azevedo, J.M. Park, S. Smith, P. Snyder, M. Umansky

(and probably more)

BOUT++ 2015 Workshop

December 16-18, 2015 @
Livermore LLNL

AToM: Advanced Tokamak Modeling FES/ASCR SciDAC project: 9/2014-8/2017

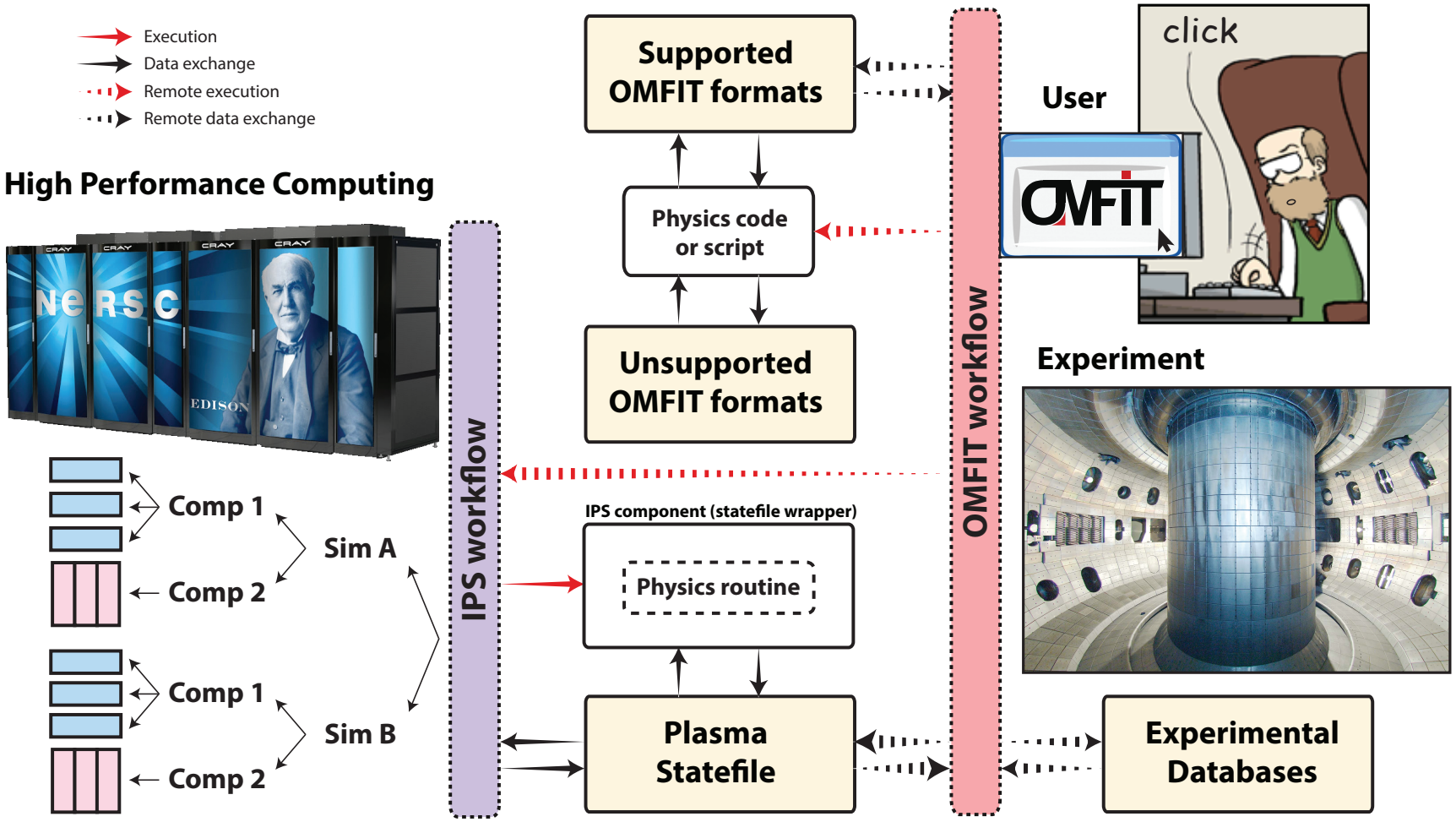
- The goal of AToM is to **enhance and extend** predictive modeling capabilities that currently exist within the US magnetic fusion program.
- The approach is to **support** rather than **subvert** current workflows, build new essential infrastructure, and guide integration.
- The central philosophy is pragmatic: take a **bottom-up** approach that leverages **existing research activities** and **collected wisdom** embodied in legacy tools.
- Move **organically** toward a whole device modeling (WDM) capability that **broad community buy-in**.

AToM has seven research thrusts

1. Maintain OMFIT+IPS **frameworks**, provide wrappers and streamlining
2. Create simulation **workflows** for core, pedestal, and scrape-off-layer
3. Develop workflows for experimental **validation**
4. Accelerate **COGENT** integration into AToM with **FASTMath**
5. Carry out **SUPER** performance engineering of **xGYRO/ NEO**
6. Establish a **data management** scheme, provenance and services
7. Provide user **support** and community outreach

AToM couples OMFIT and IPS frameworks

OMFIT= user interface IPS = HPC scheduling



AToM supports wide range of lightweight and HPC-enabled components

OMFIT	IPS	COGENT	CURRAY
DAKOTA	EFIT	EPED	ESC
FASTRAN	GATO	GENRAY	TSC
GYRO	CGYRO	NEO	CORSICA
NIMROD	NUBEAM	NTCC library	PRGEN
GLF23	TGLF	TGYRO	TORAY
TORIC	M3D-C1	ONETWO	TRANSP
LE3	NEO3D	SURFMIN	C2
BOUT++	AORSA	TORBEAM	SOLPS

HPC code, HPC compound code

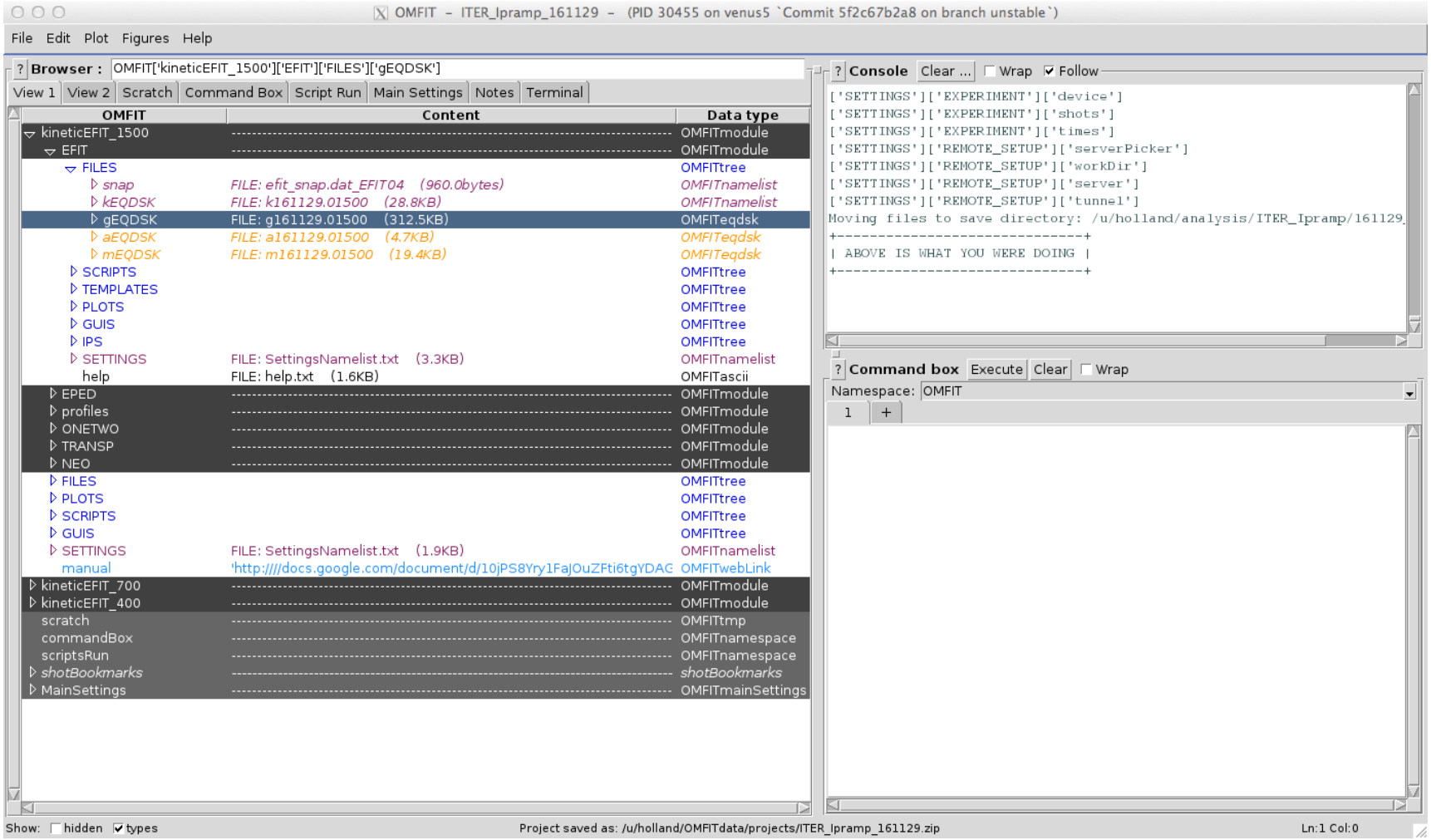
OMFIT centerpiece: tree data structure

OMFIT-tree is a hierarchical, self-descriptive data structure that enables data exchange between different codes

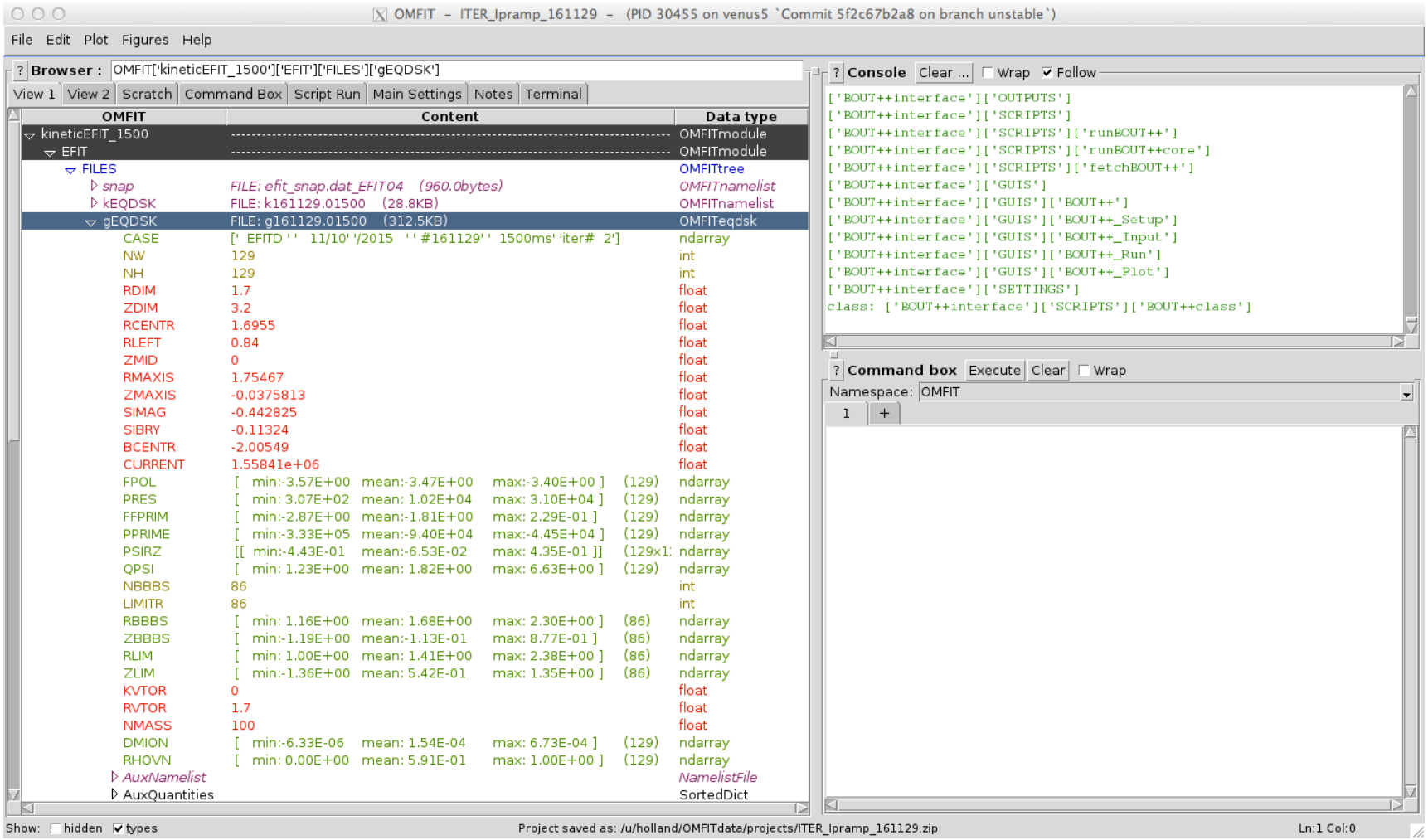
- Collects data **independent of origin/type**
- Component content stored in a **subtree**
- **No *a priori* decision** of what is stored and how
- Codes communicate by referring to **tree data**
- **Free-form** equivalent to elusive fusion "*statefile*"

Like *MDS+* or file system on your own laptop, data is stored in the **most natural form** to accomplish a given task

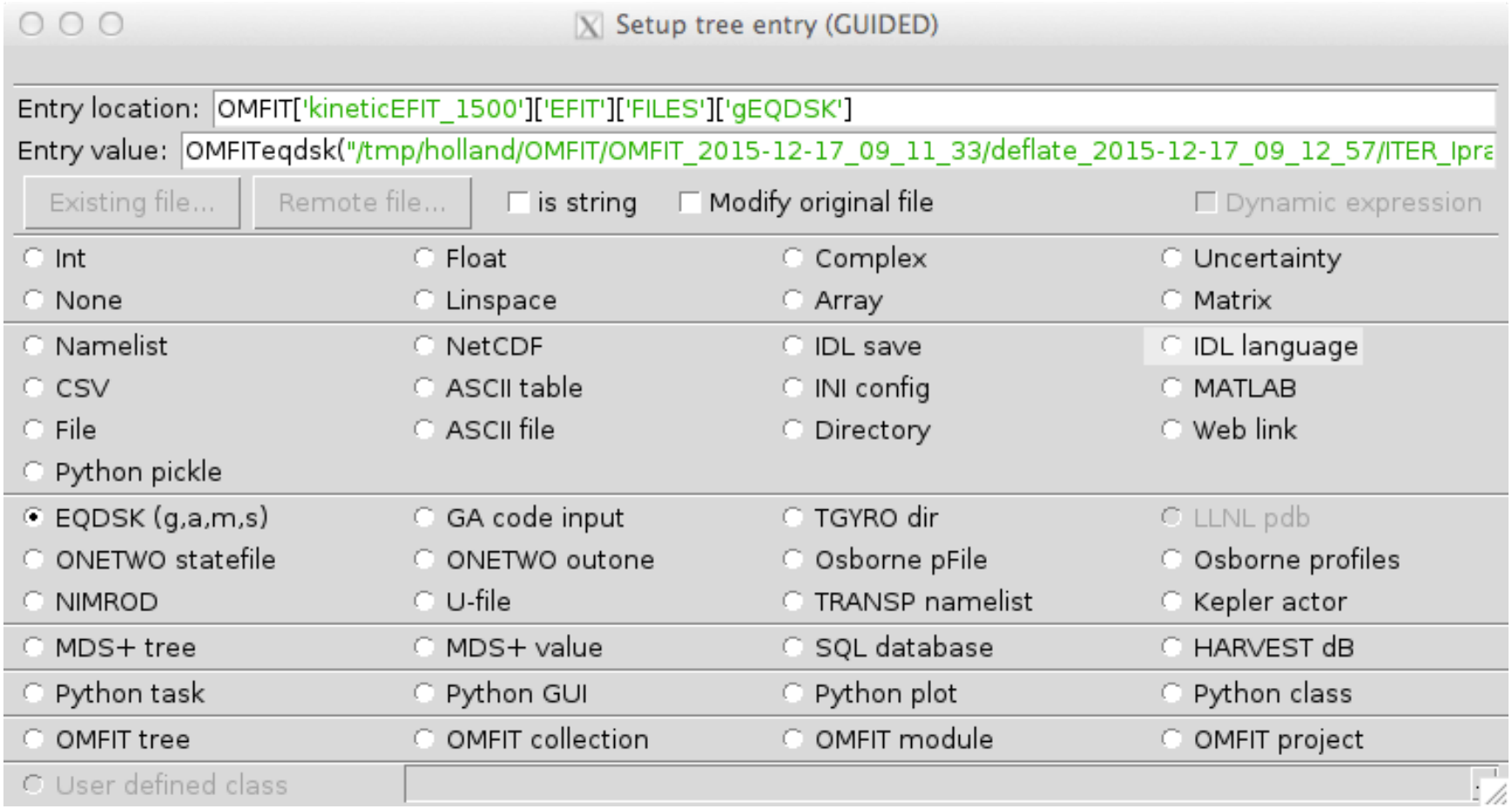
OMFIT provides GUI to explore tree, execute Python commands



OMFIT provides GUI to explore tree, execute Python commands

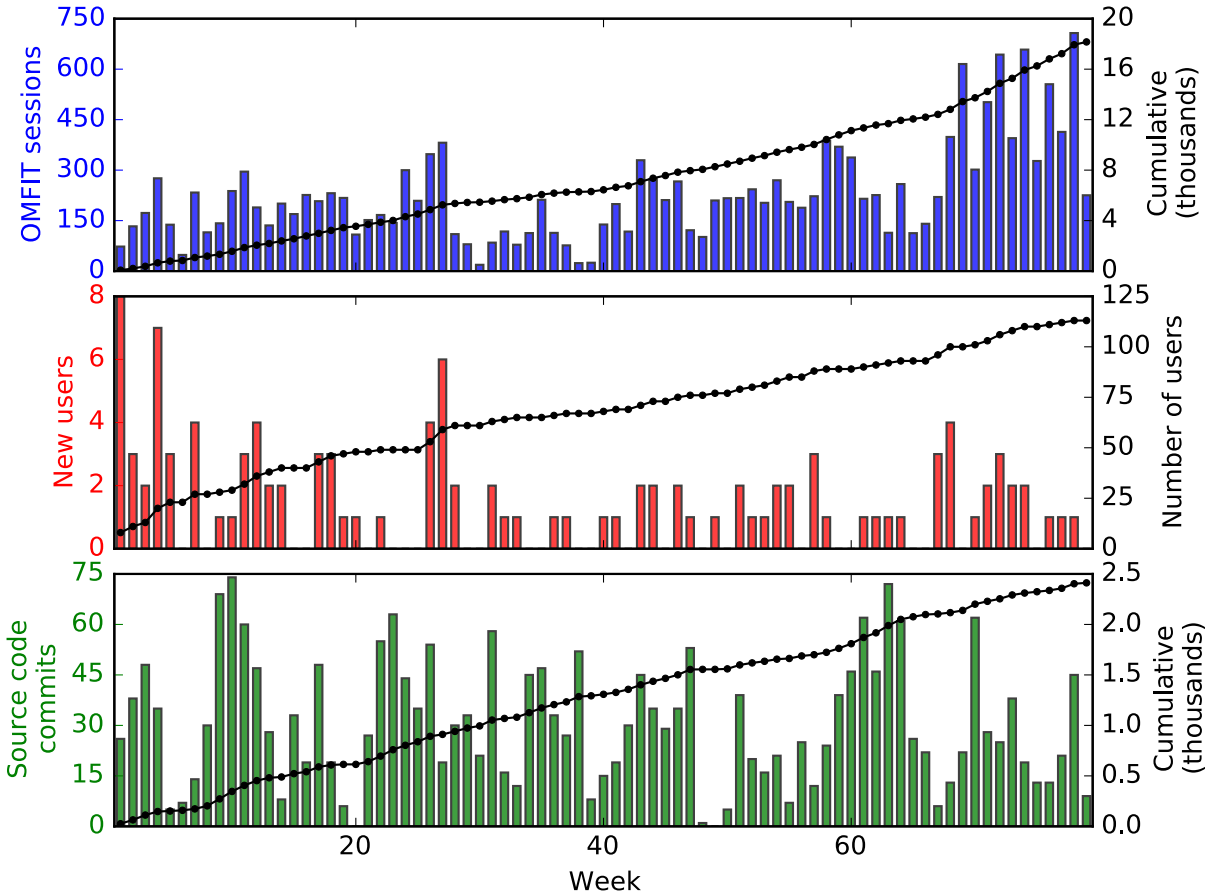


OMFIT supports many common structure and unstructured data formats



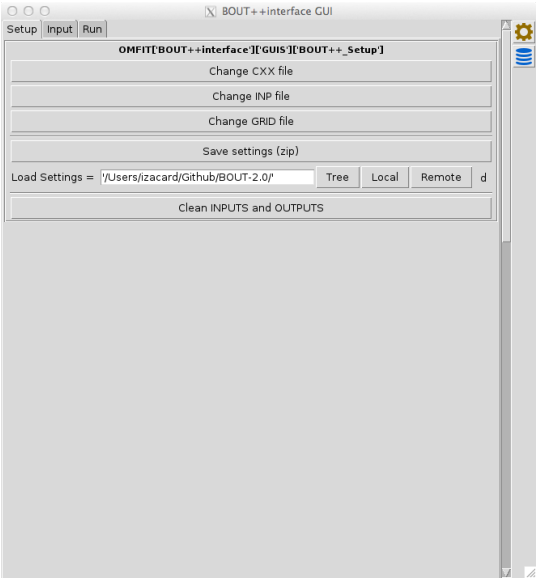
OMFIT has **extensive** documentation and community buy-in

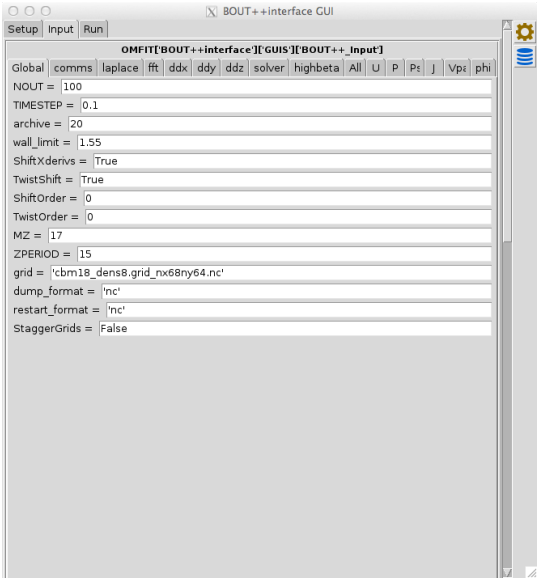
- **http://gafusion.github.io/OMFIT-source/development.html**
- Usage of OMFIT just on local GA system VENUS

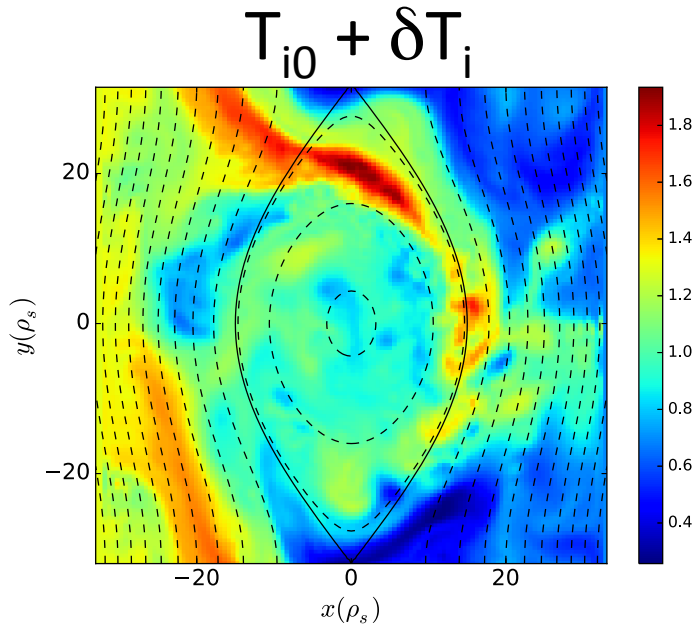


OMFIT has a BOUT++ module and GUI, but focused on basic physics study of slab ITG + magnetic island (not elm-xx) module

- Module does support running on local machine, SDSC Triton shared computing cluster, and NERSC



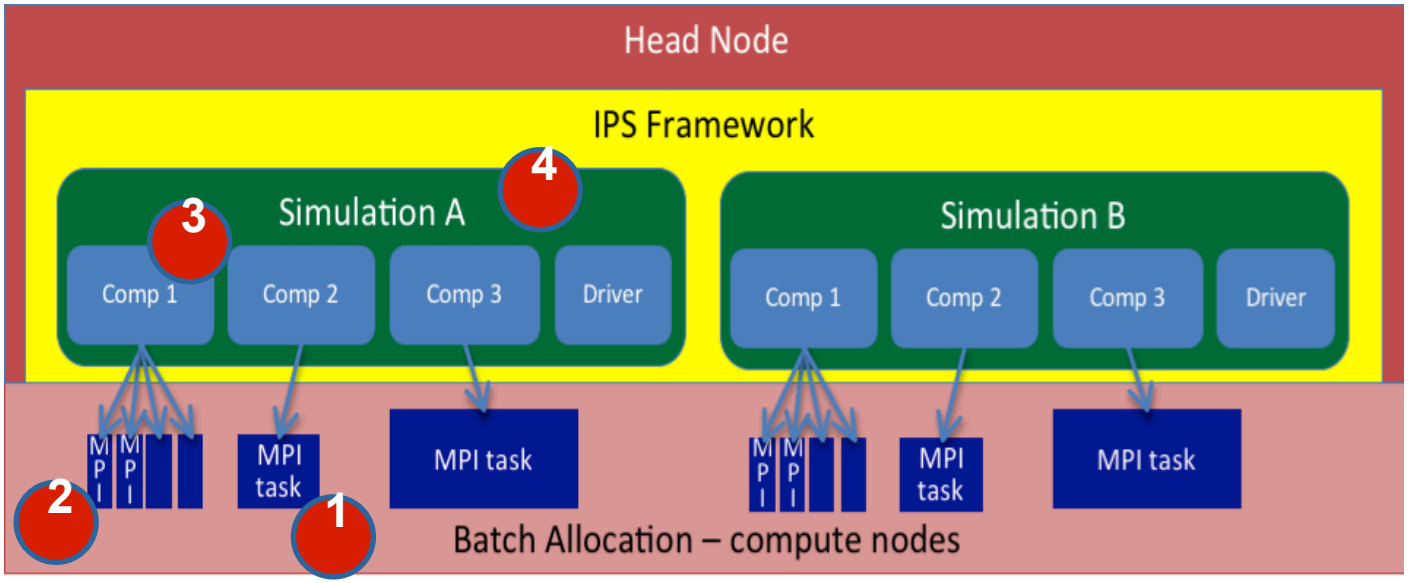




IPS : Multi-Level Parallelism

1. Individual “tasks” (physics executables) can be parallel.
2. Components can launch multiple tasks.
3. Multiple components can run concurrently.
4. Multiple independent simulations can run concurrently.

Maximal resource utilization via hierarchical concurrency



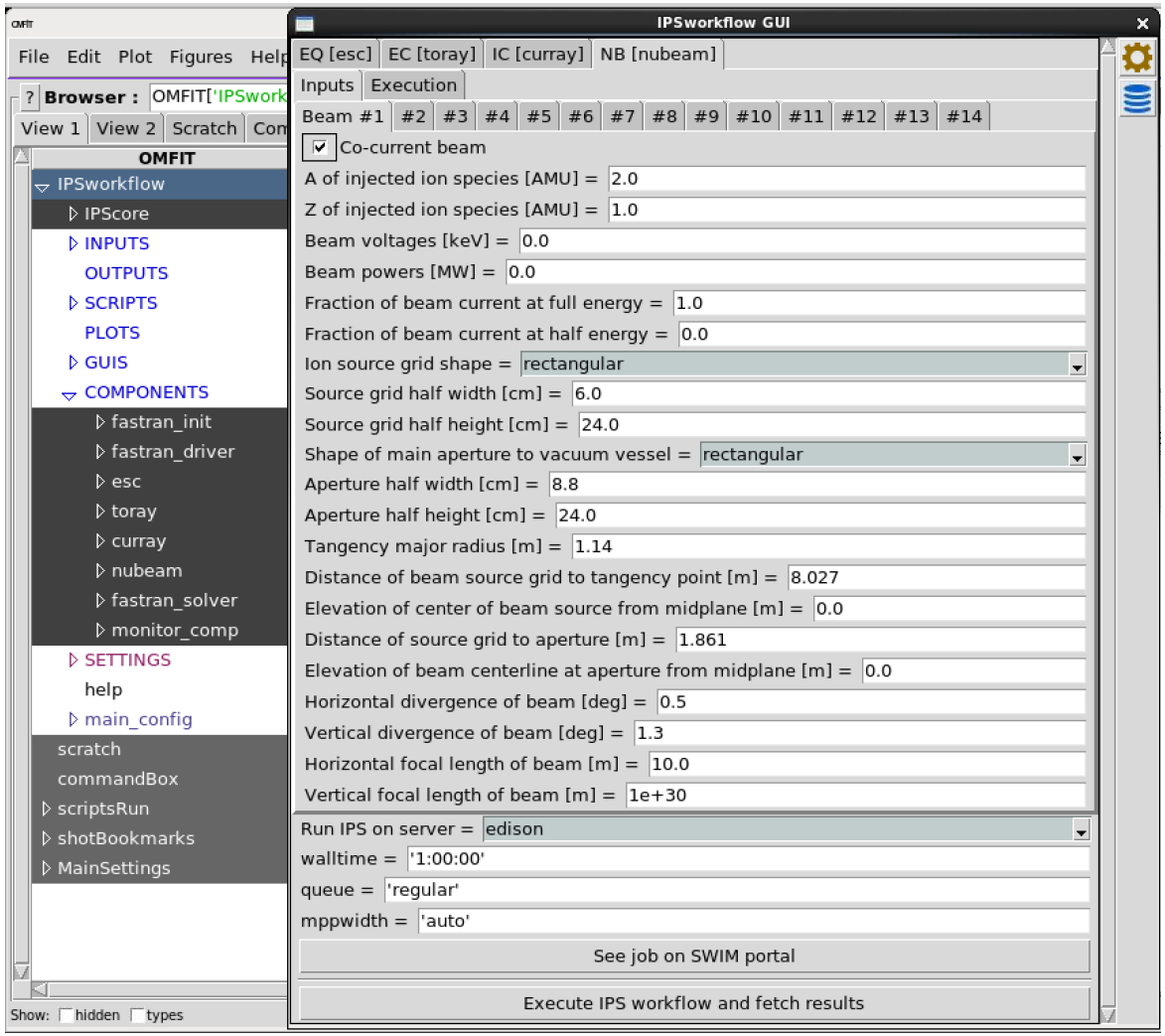
UC San Diego

OMFIT has GUI Interface to IPS

- Two IPS modules:

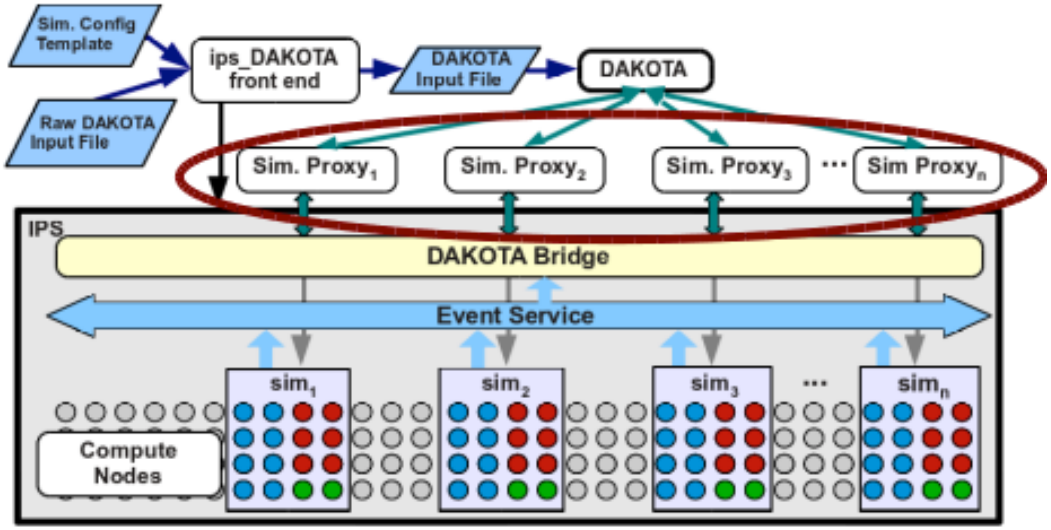
1. **IPScore:** manage IPS configuration and execution

2. **IPSworkflow:** extract workflow from existing IPS simulation



DAKOTA toolkit implemented in IPS

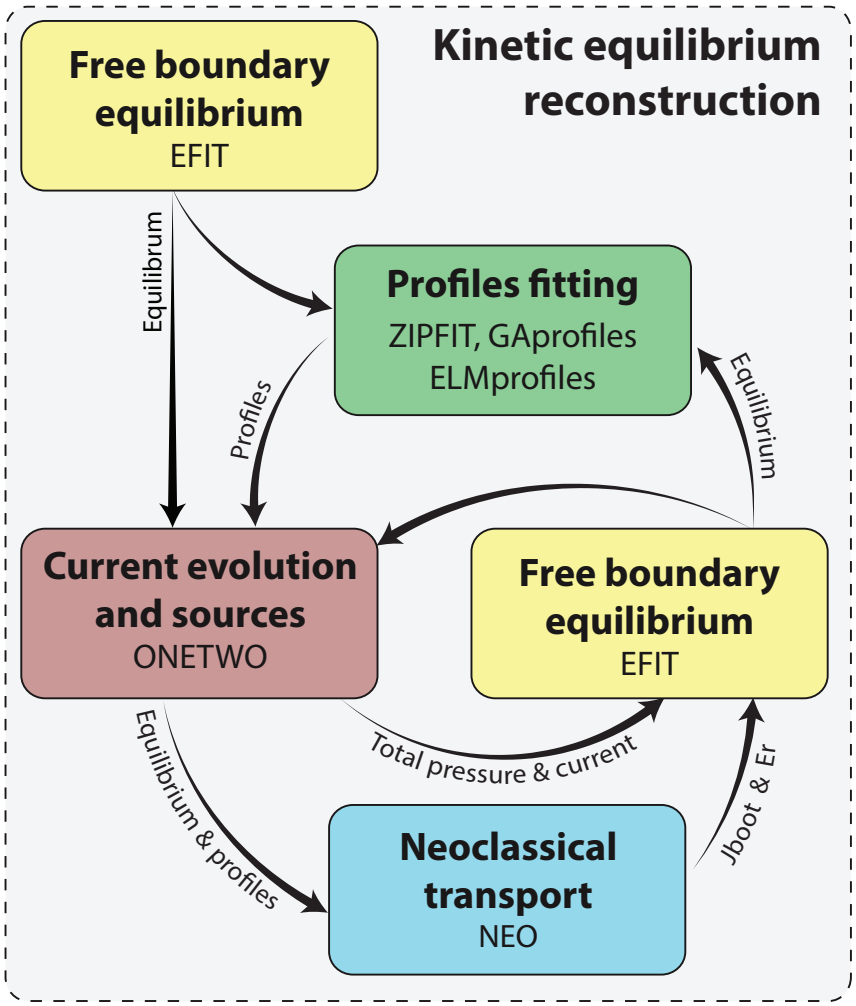
- DAKOTA toolkit from SNL
  - Toolkit for design optimization, parameter estimation, UQ, sensitivity analysis, ..

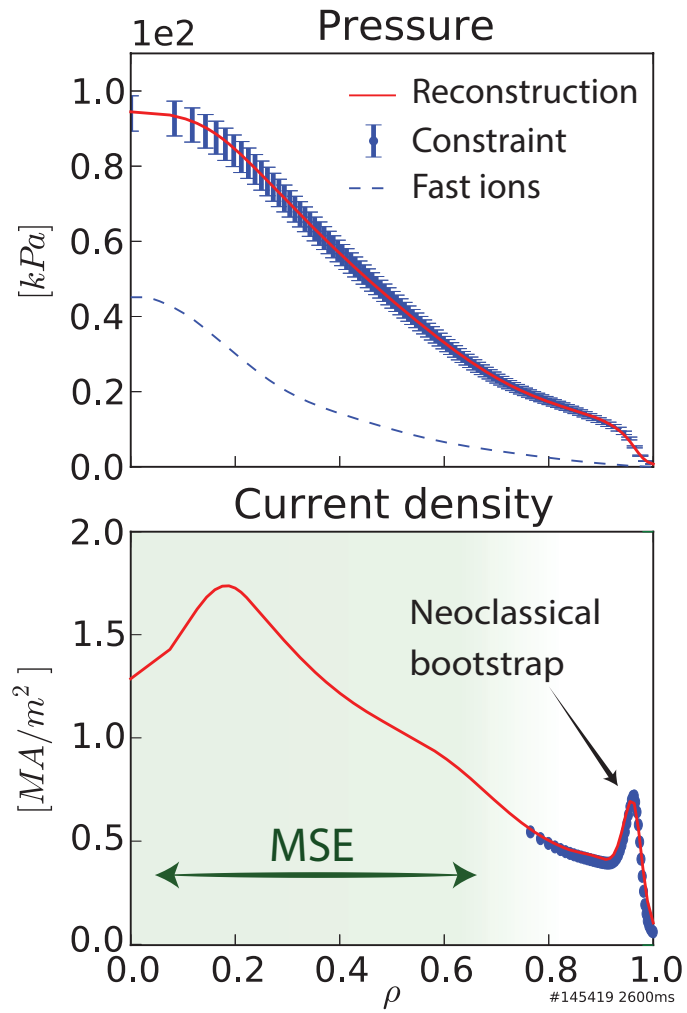


- IPS-DAKOTA integration
  - Single IPS framework instance
  - Manage many, dynamically created DAKOTA (coupled) simulations.
- ATOM use cases, so far these are simple parameter scans ...
  - Core-pedestal coupling (IPS-EPED).
  - TGLF ITER calibration (IPS-GYRO).

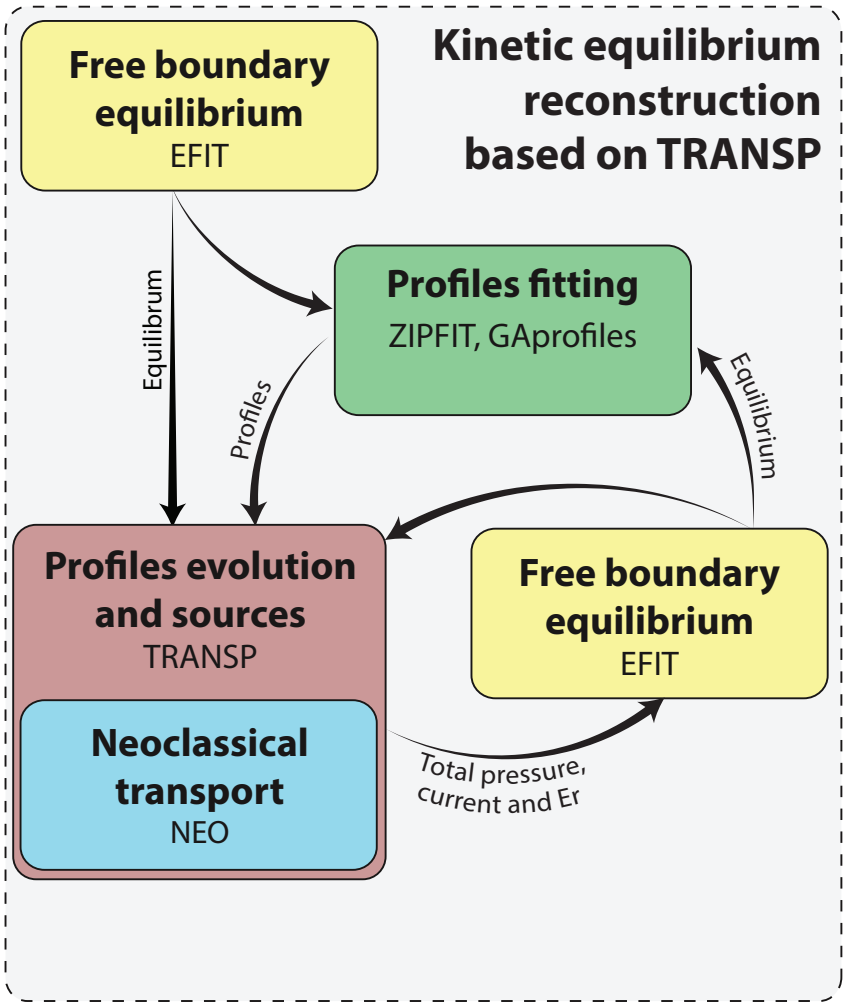
UC San Diego

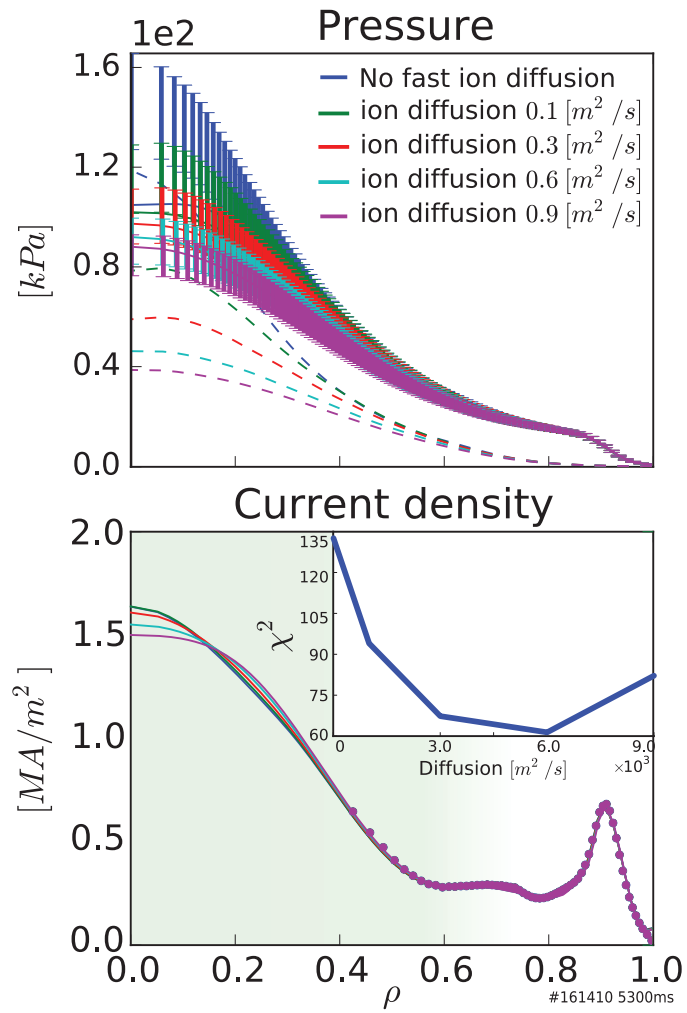
Most common OMFIT workflow: kinetic equilibrium reconstruction





Most common OMFIT workflow: kinetic equilibrium reconstruction

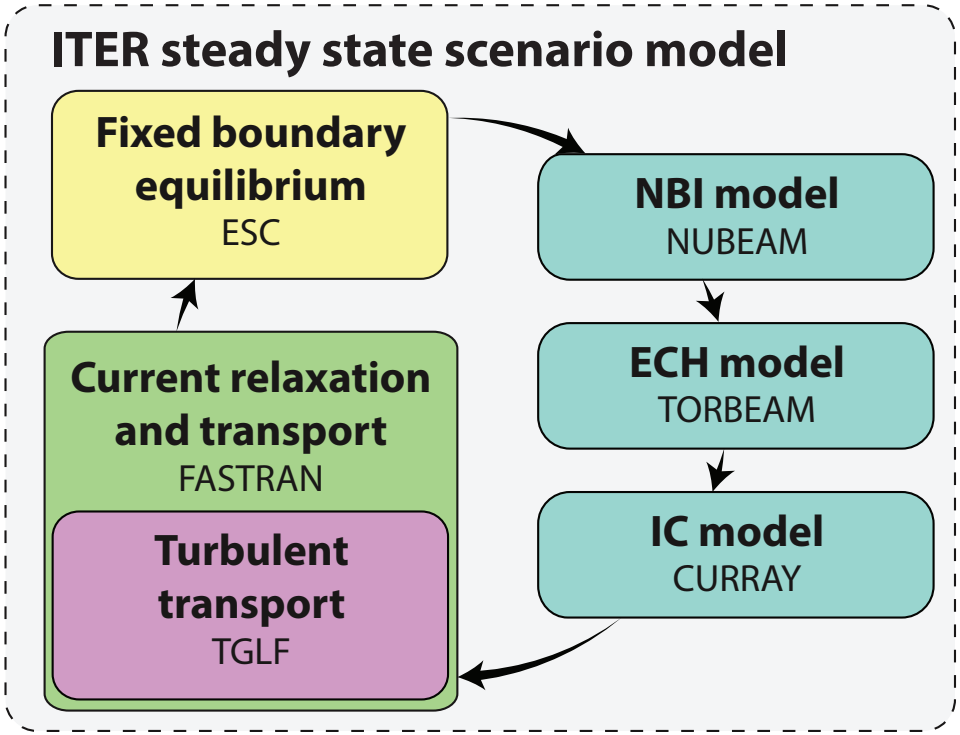


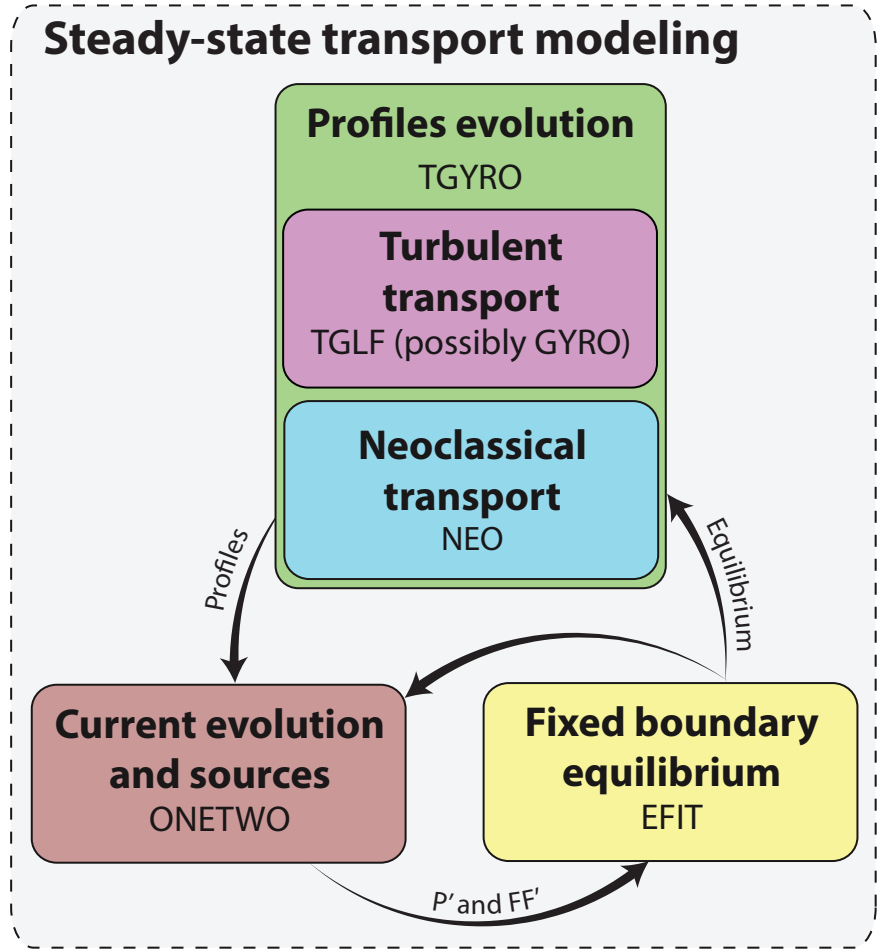


Significant ongoing working to develop improved experimental profile fitting routines for OMFIT

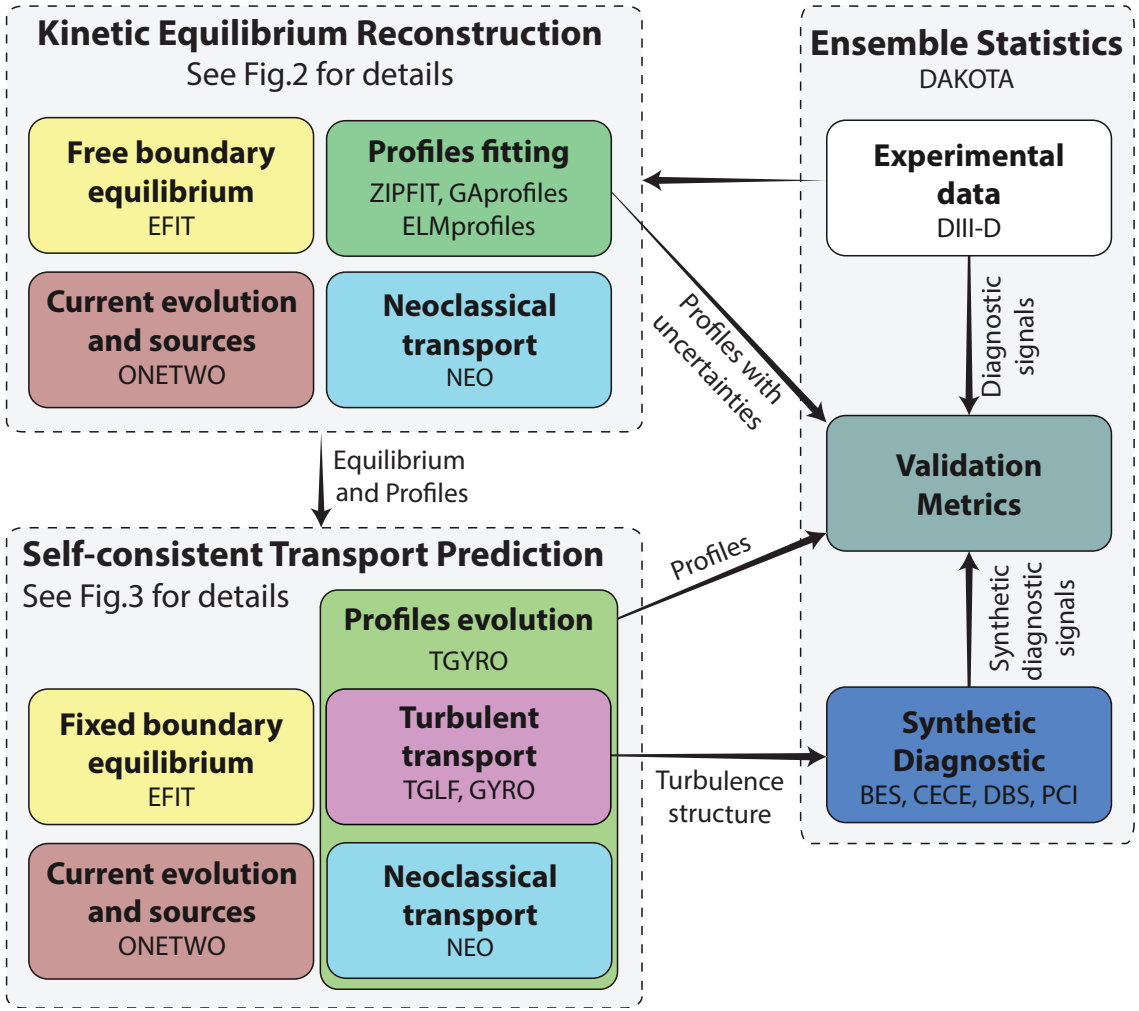
- Many institutions have no standardized fitting tools: homebrewed tools from user to user, group to group
- Historically tools have focused on core fitting at expense of edge or vice-versa
  - Which diagnostics can be used, constraints apply, data filtering, functional forms used, etc.
- New effort aims to combine best aspects of existing tools into new machine-independent, python-based toolset
  - Incorporates various polynomial, spline, picewise linear scale length, and Gaussian process fit forms
  - Mapping and interpolation of data onto uniform timebases
  - Fit either as series of 1D timeslices or direct 2D (R,time)
  - Built upon generic freely available Python modules including uncertainty tracking

AToM workflows for steady-state core transport predictions build upon kinetic equilibrium workflow



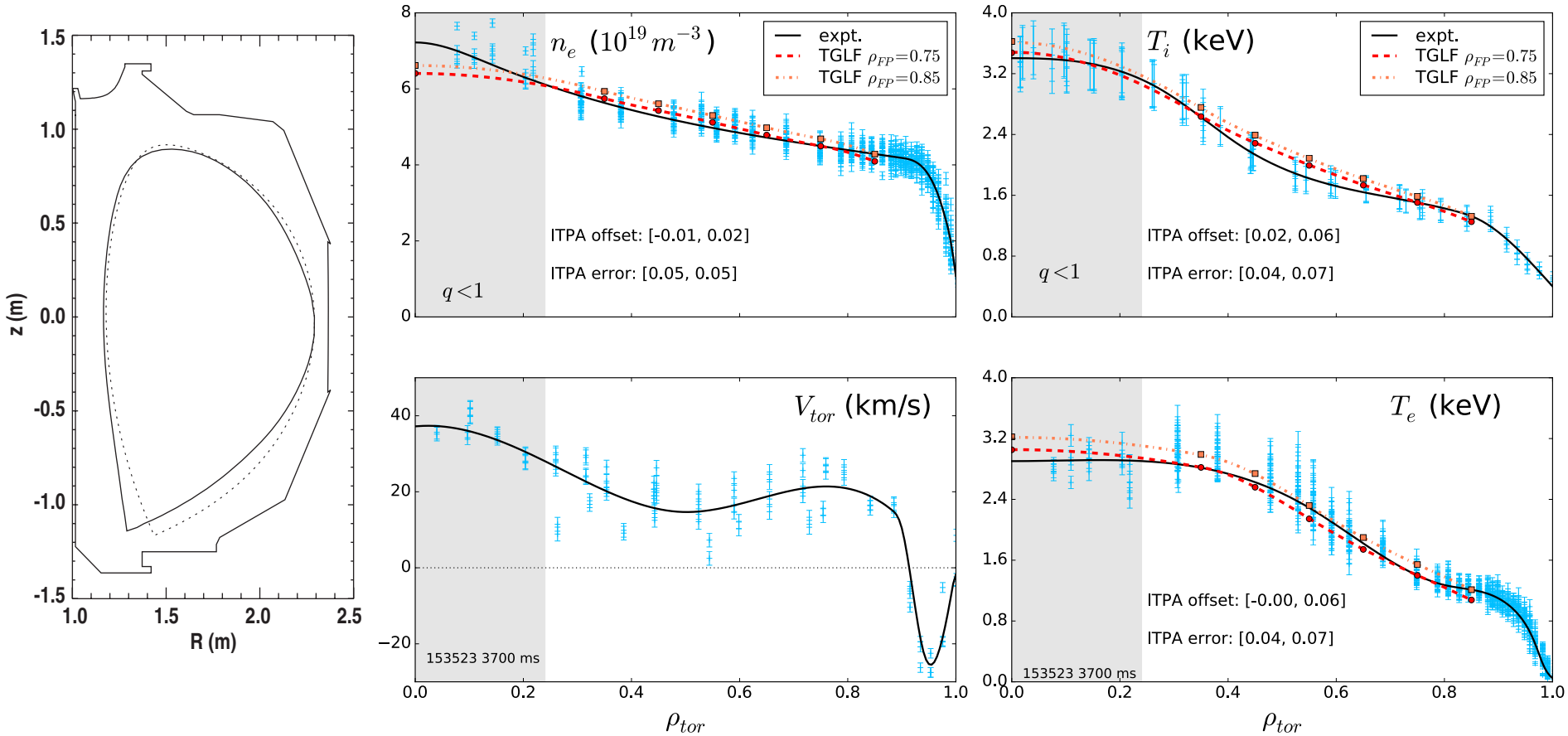


AToM core transport model validation workflow also builds upon kinetic equilibrium workflow



Example: testing predictions of TGLF transport for a DIII-D ITER baseline discharge

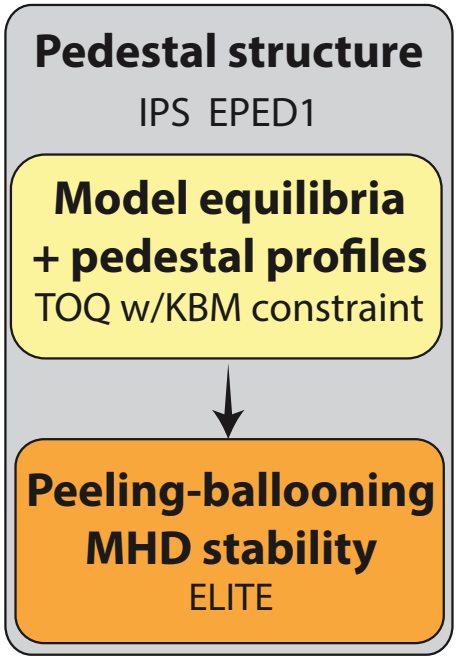
- Discharge has similar shape to ITER, low torque, and dominant electron heating

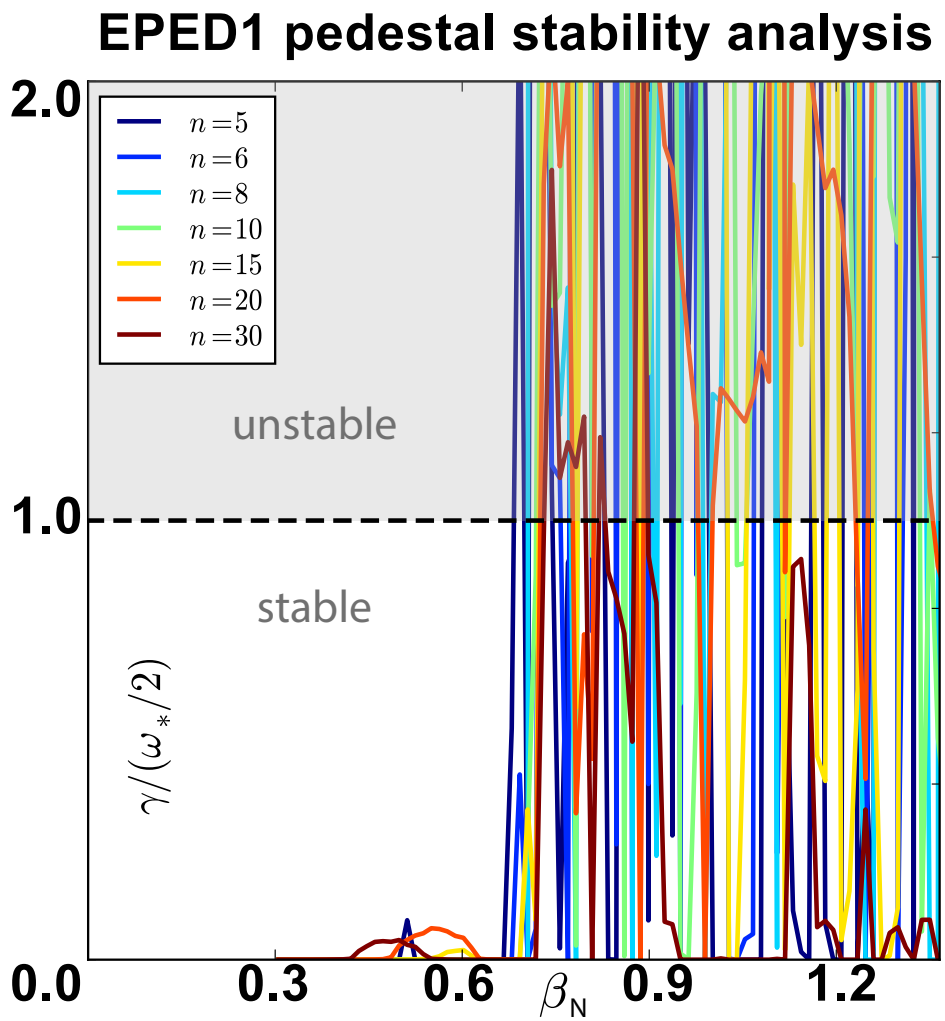


Implement an IPS-EPED1 workflow to enable fast pedestal structure prediction

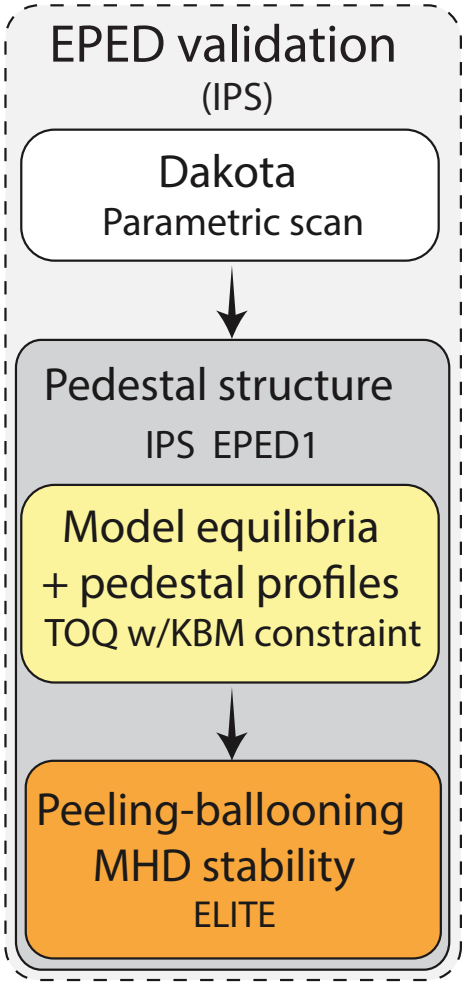
- Parametric variations of β_N to find achievable stable pedestal

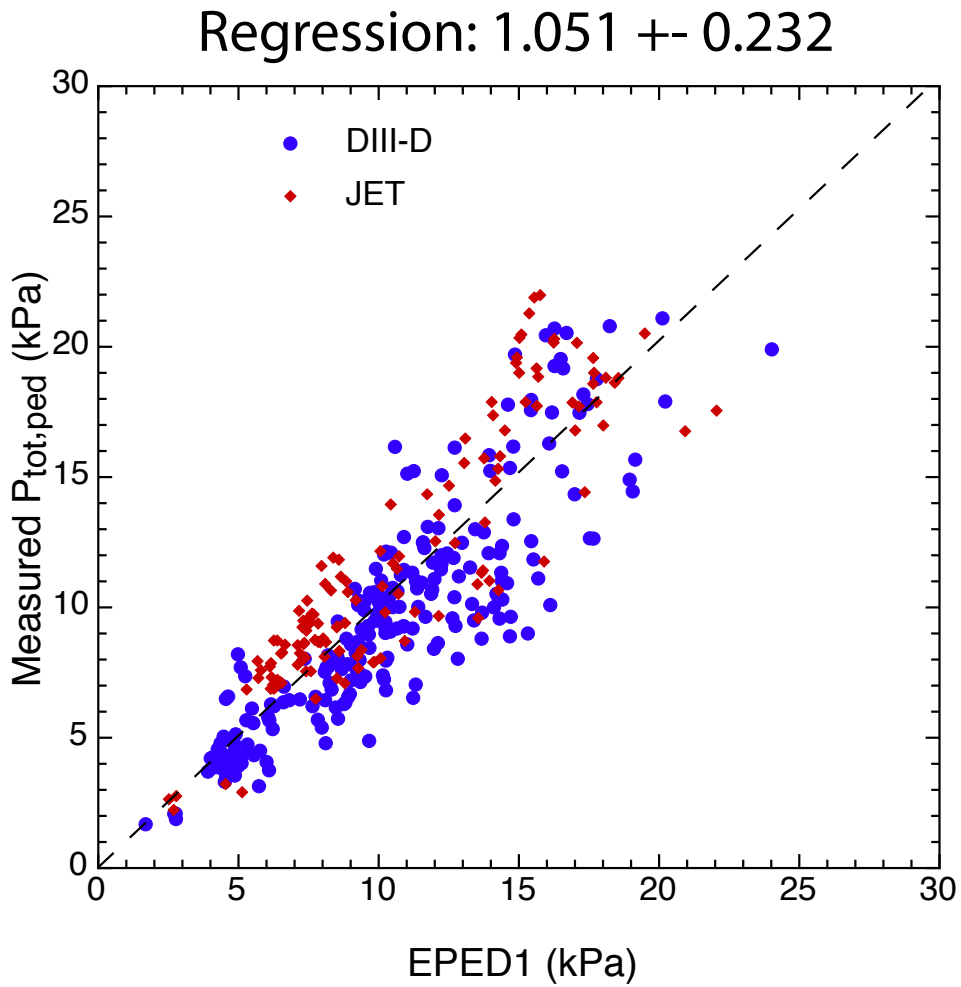
- **93** TOQ equilibria, **651** ELITE runs: ~ **20 min** on **651** cores at NERSC





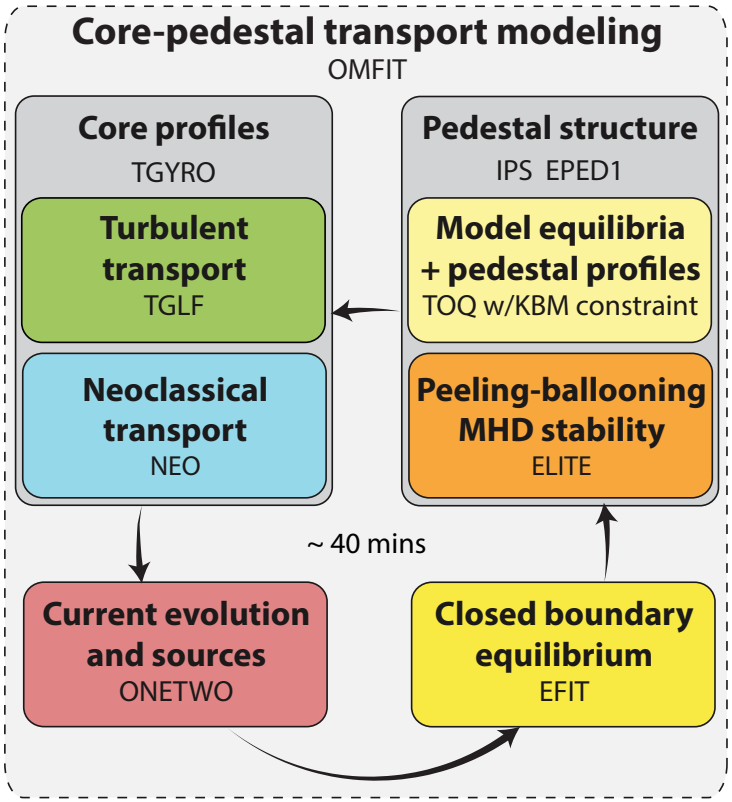
IPS-EPED1 workflow reproduces [Snyder:2009] validation results in **1.5 hours at NERSC on 3600 cores** vs. **1 week on GA workstation**

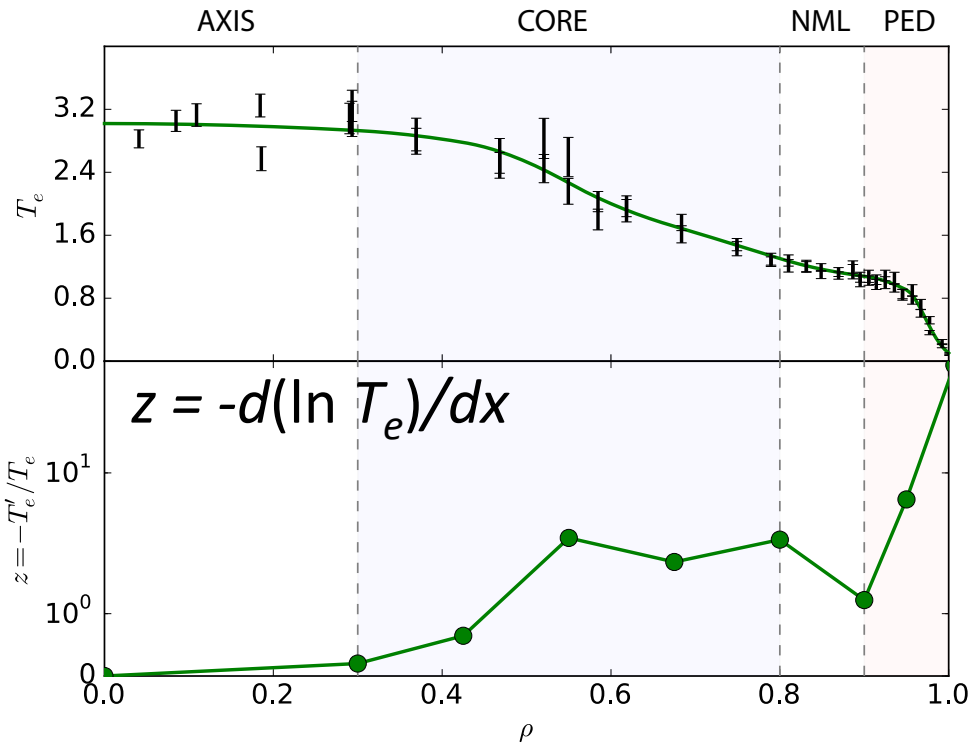




New AToM workflow enables self-consistently coupled equilibrium, core transport, and edge stability calculations

Enabled by separation of transport, MHD, and current diffusion timescales





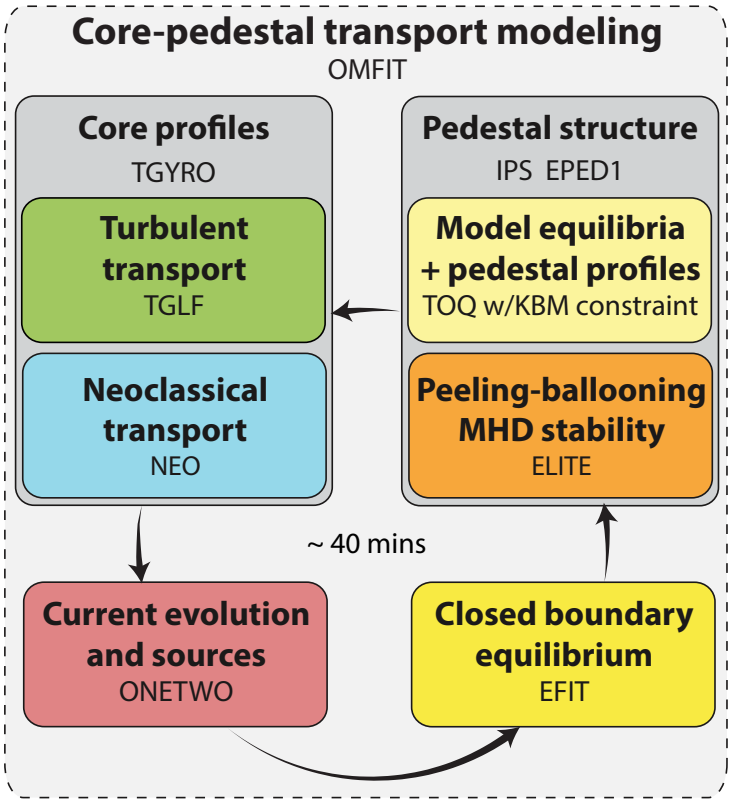
Describe profiles in terms of piecewise linear inverse scale length profiles *z* with small number of radial nodes

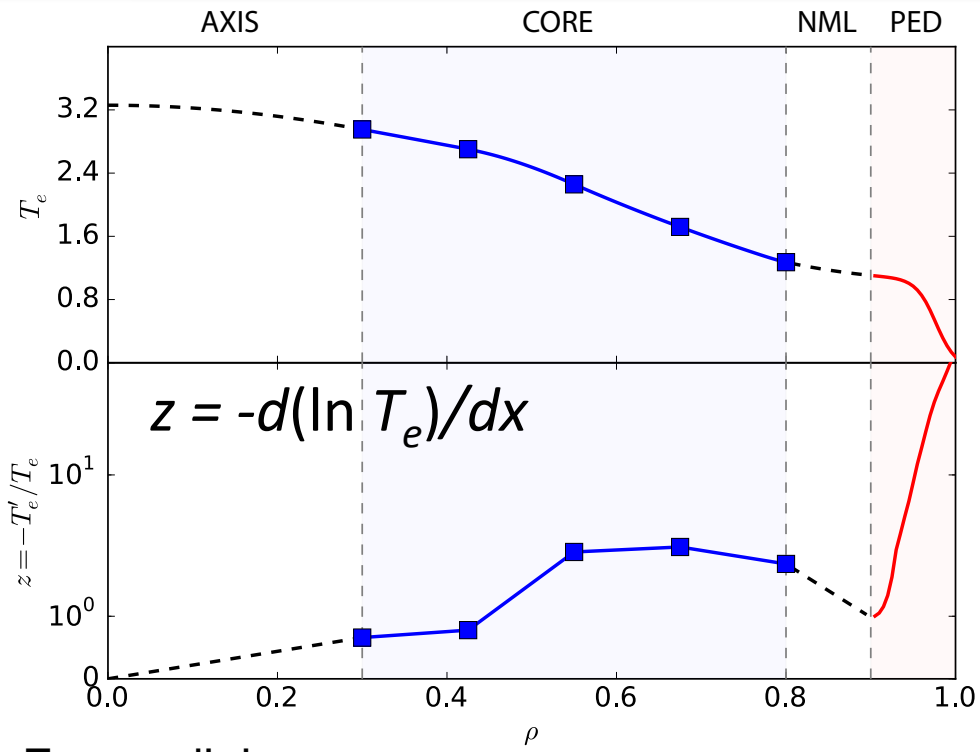
Approach inspired by ability of linear scale-length fitting to capture experimental trends

UC San Diego

New AToM workflow enables self-consistently coupled equilibrium, core transport, and edge stability calculations

Enabled by separation of transport, MHD, and current diffusion timescales



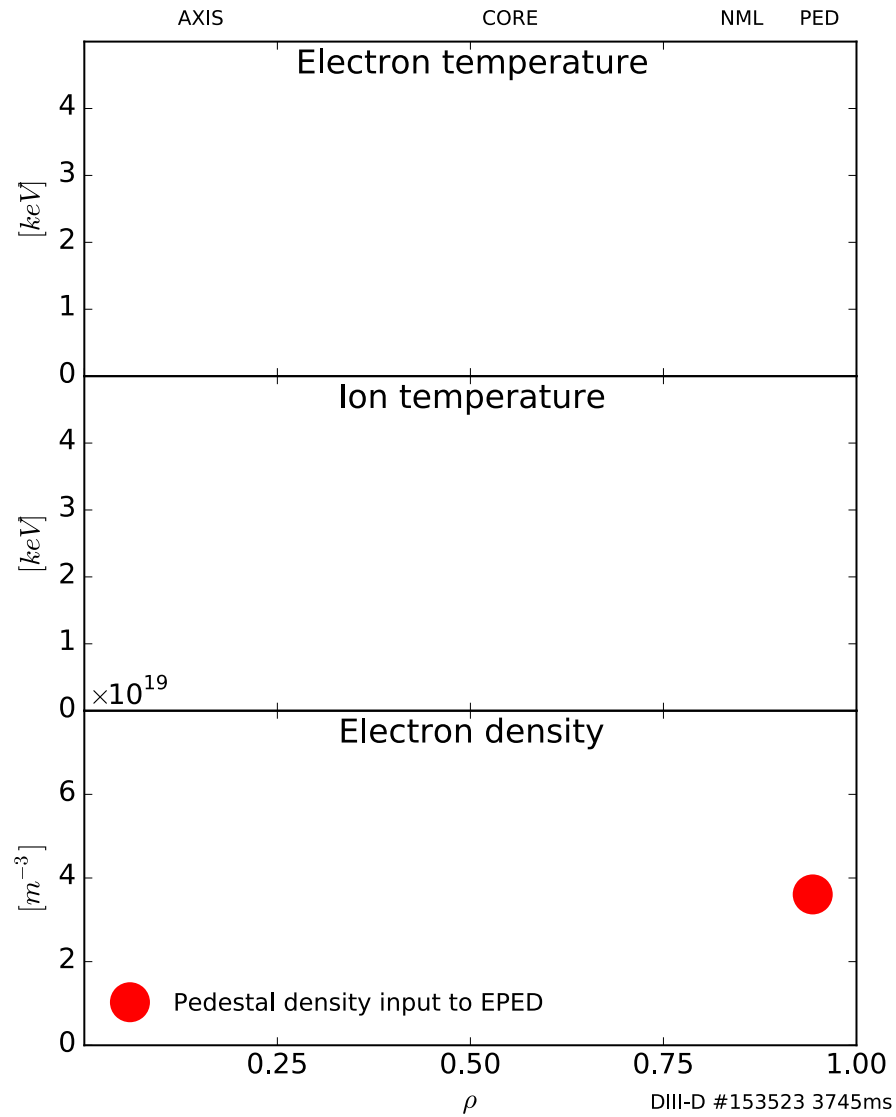


Four radial zones:

- **AXIS**: *z* linearly to zero from **CORE**
- **CORE**: local GK or GF model
- **NML**: linear interp. of *z* from **CORE** to **PED**
- **PED**: *z* from pedestal structure model

Apply workflow to DIII-D ITER baseline case

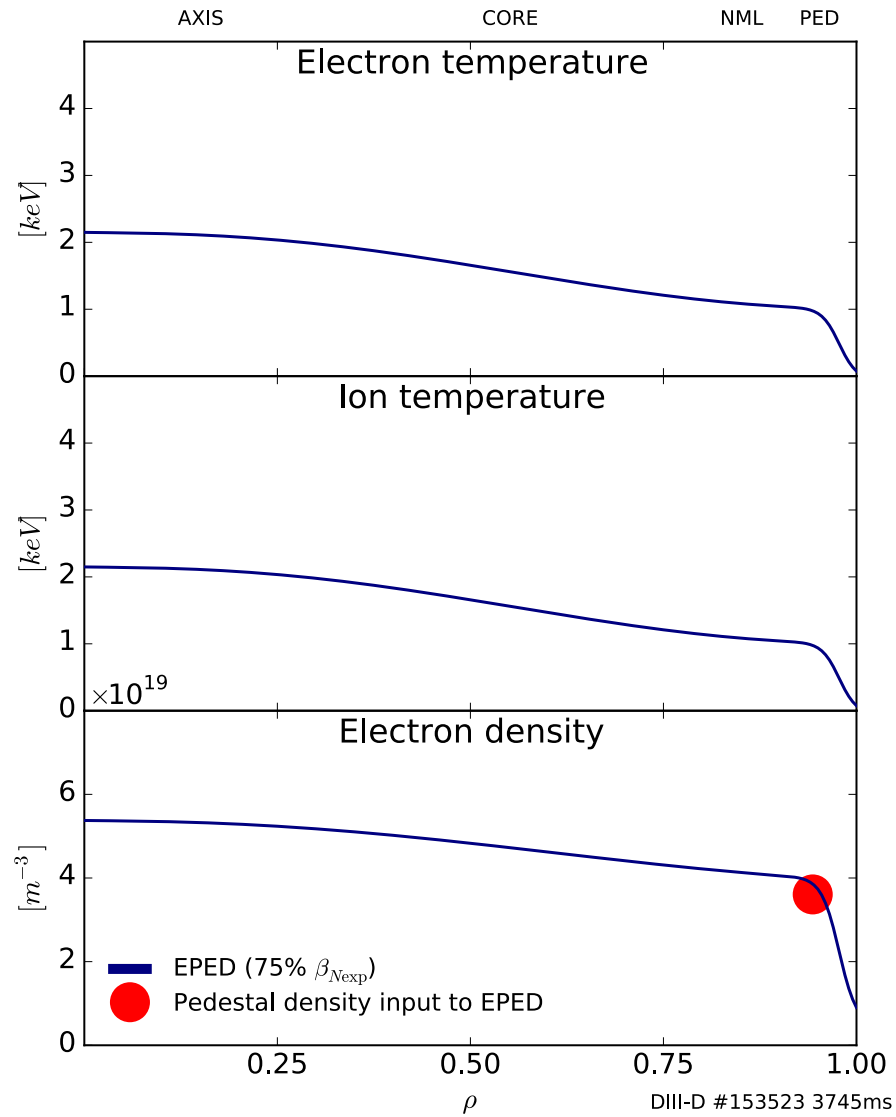
- Inputs are plasma shape, B_T, fixed $J_{tor}(r)$ and $V_{tor}(r)$, Z_{eff}, sources, **$n_{e,ped}$**, and **guess for β_N**



Apply workflow to DIII-D ITER baseline case

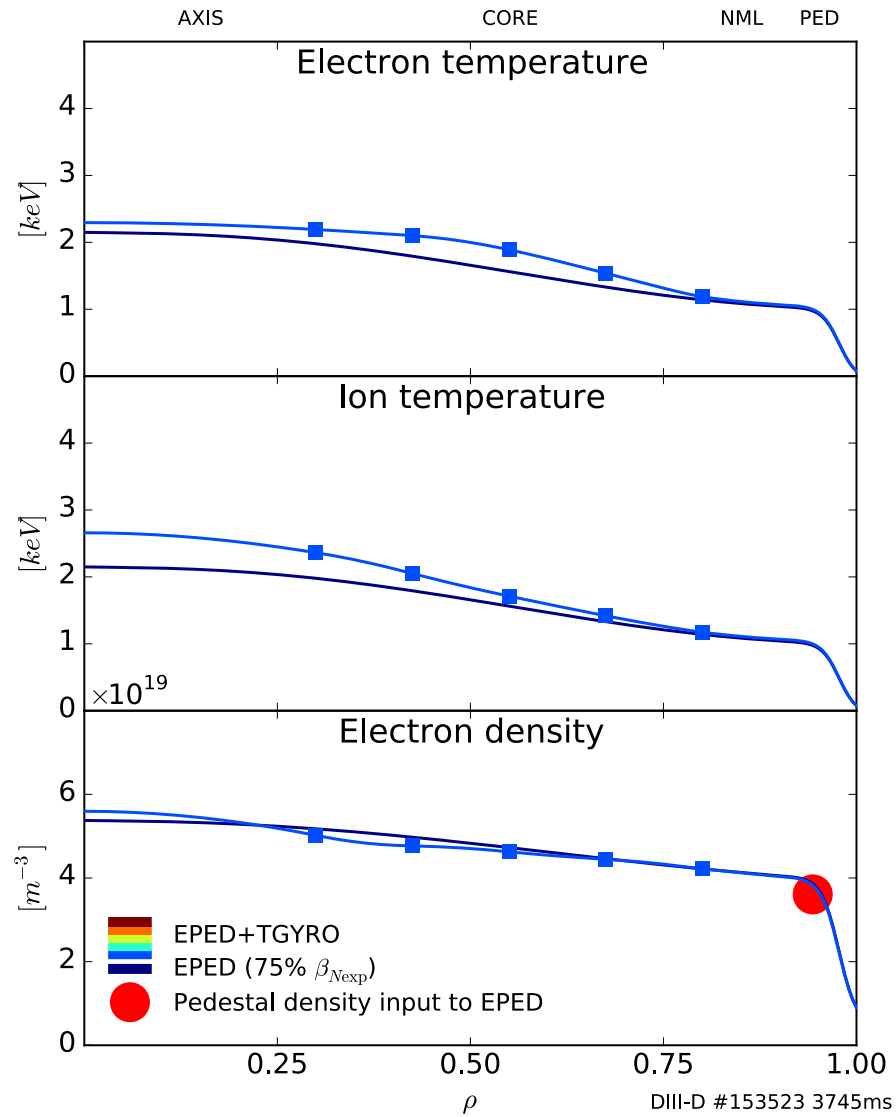
- Inputs are plasma shape, B_T, fixed $J_{tor}(r)$ and $V_{tor}(r)$, Z_{eff}, sources, **$n_{e,ped}$**, and **guess for β_N**

- Start with EPED profile, **$\beta_N = 0.75\ \beta_{N,exp}$**



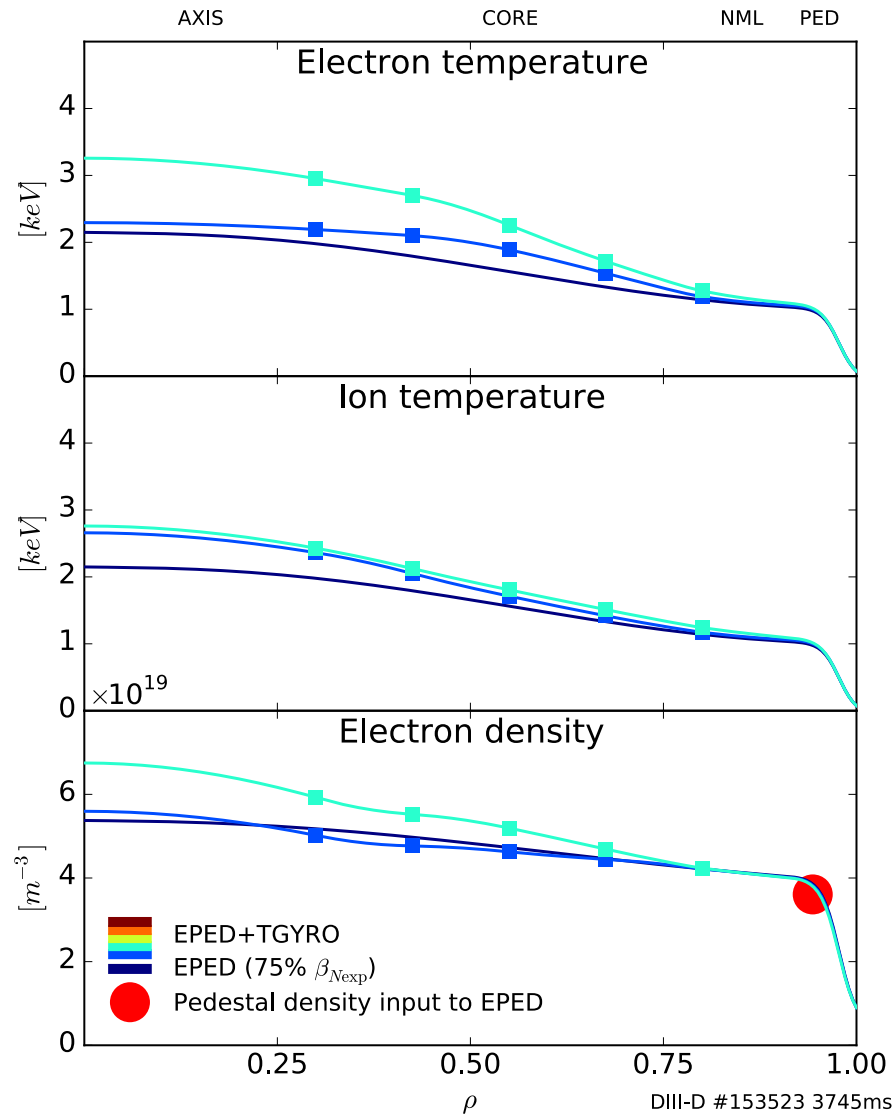
Apply workflow to DIII-D ITER baseline case

- Inputs are plasma shape, B_T, fixed $J_{tor}(r)$ and $V_{tor}(r)$, Z_{eff}, sources, **$n_{e,ped}$**, and **guess for β_N**
- Start with EPED profile, **$\beta_N = 0.75\ \beta_{N,exp}$**
- Then TGYRO-TGLF to predict CORE+AXIS profiles and new **β_N**, and iterate…



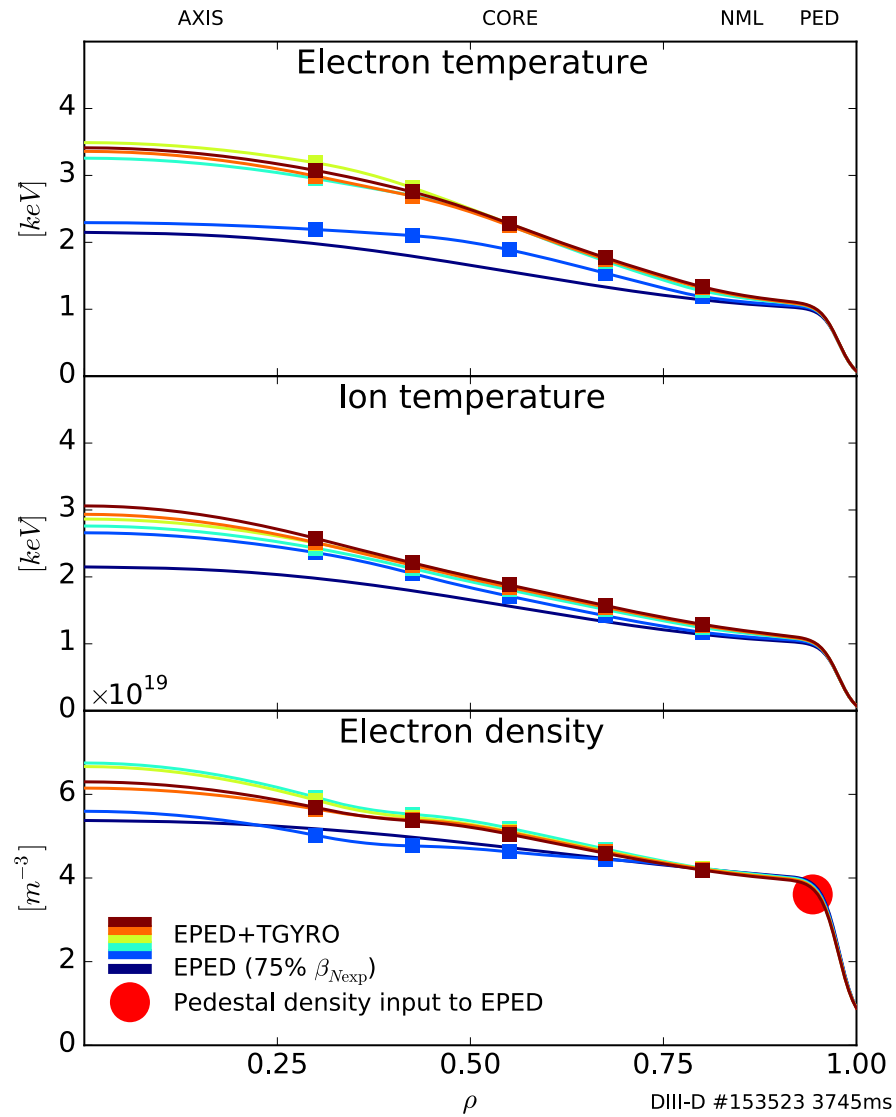
Apply workflow to DIII-D ITER baseline case

- Inputs are plasma shape, B_T, fixed $J_{tor}(r)$ and $V_{tor}(r)$, Z_{eff}, sources, **$n_{e,ped}$**, and **guess for β_N**
- Start with EPED profile, **$\beta_N = 0.75\ \beta_{N,exp}$**
- Then TGYRO-TGLF to predict CORE+AXIS profiles and new **β_N**, and iterate…



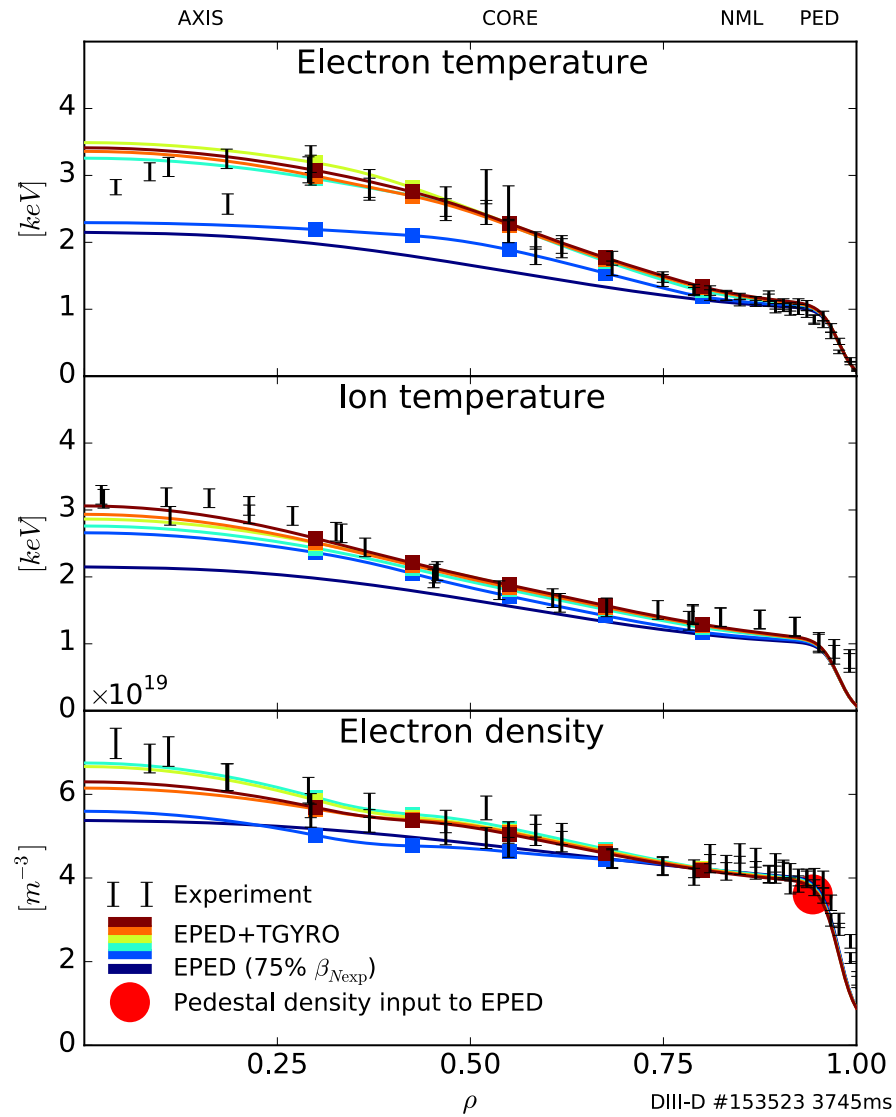
Apply workflow to DIII-D ITER baseline case

- Inputs are plasma shape, B_T, fixed $J_{tor}(r)$ and $V_{tor}(r)$, Z_{eff}, sources, **$n_{e,ped}$**, and **guess for β_N**
- Start with EPED profile, **$\beta_N = 0.75\ \beta_{N,exp}$**
- Then TGYRO-TGLF to predict CORE+AXIS profiles and new **β_N**, and iterate…

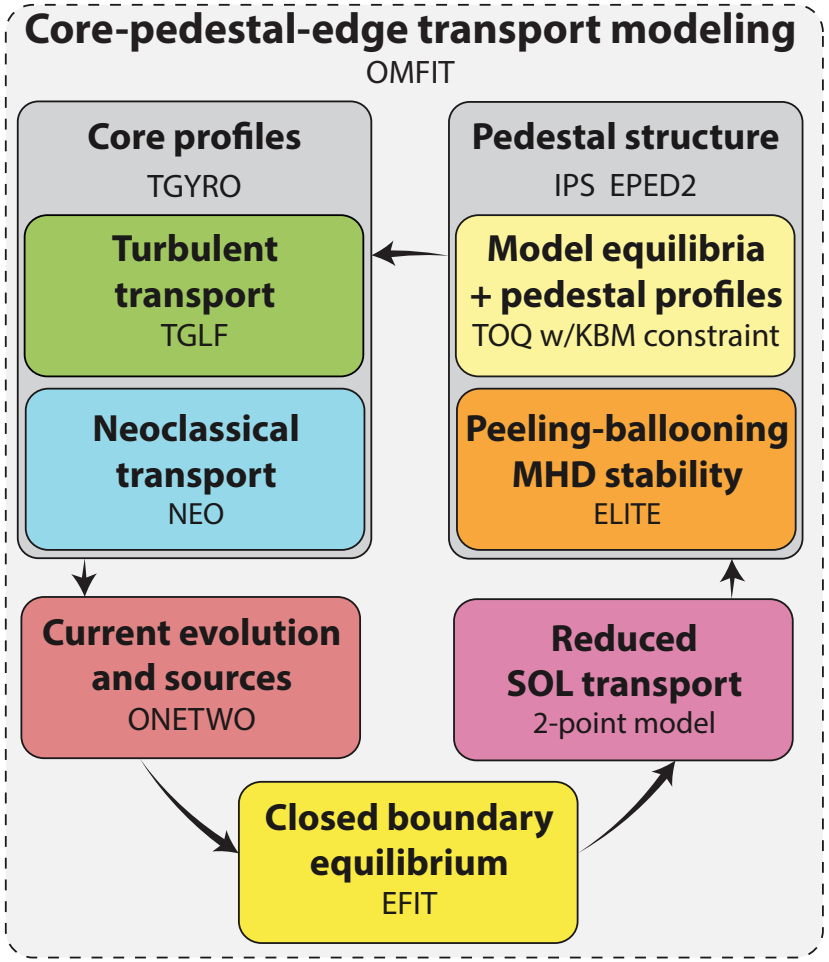


Apply workflow to DIII-D ITER baseline case

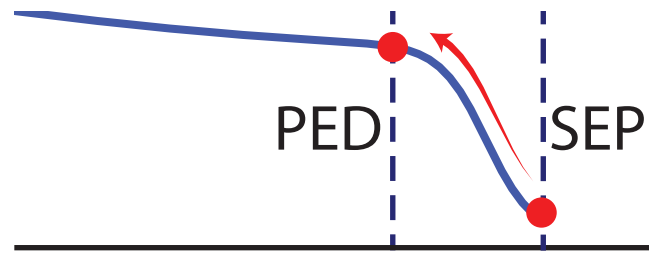
- Inputs are plasma shape, B_T, fixed $J_{tor}(r)$ and $V_{tor}(r)$, Z_{eff}, sources, **$n_{e,ped}$**, and **guess for β_N**
- Start with EPED profile, **$\beta_N = 0.75\ \beta_{N,exp}$**
- Then TGYRO-TGLF to predict CORE+AXIS profiles and new **β_N**, and iterate…
- Compare results against expt. data



Next step: extend beyond separatrix by developing a self-consistent core-pedestal-SOL model



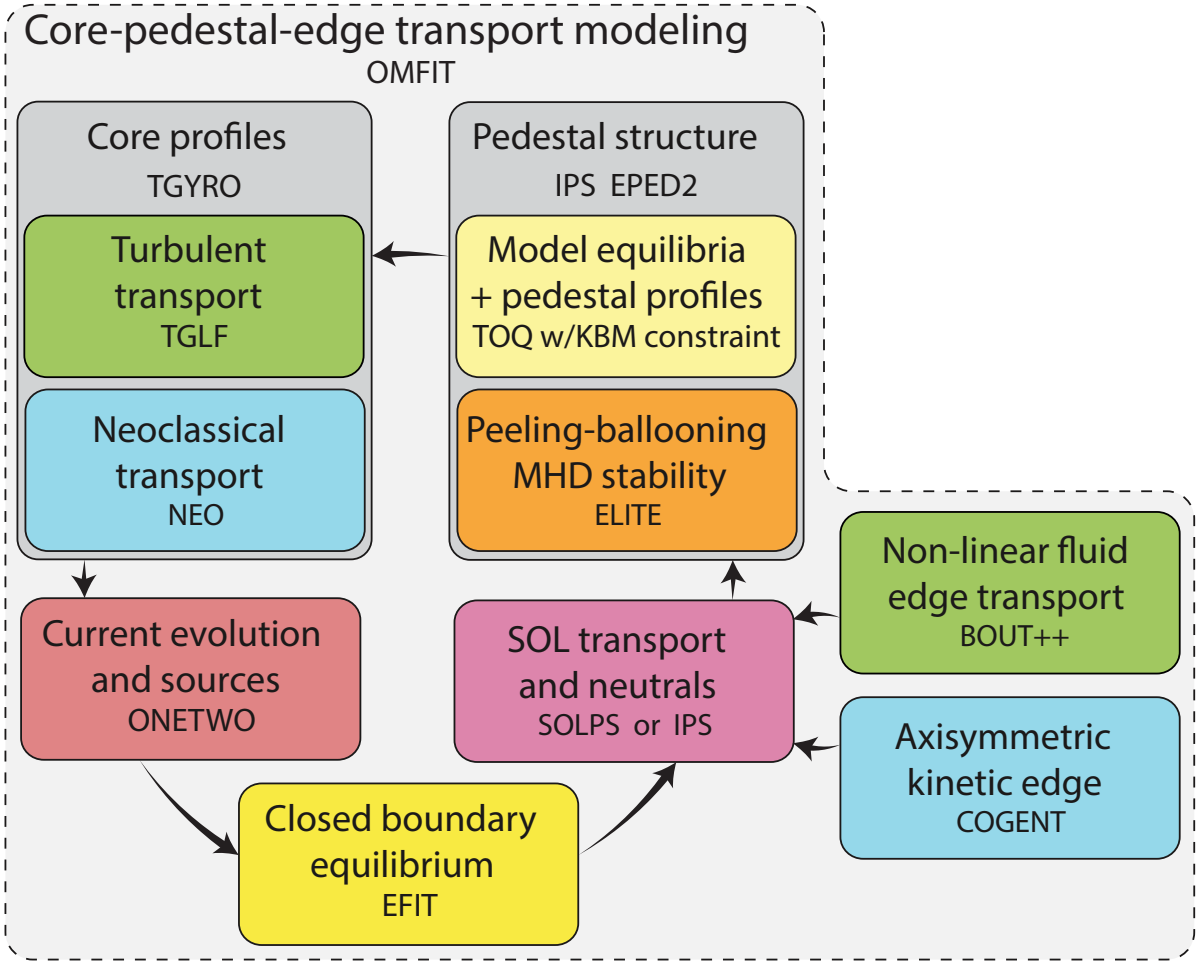
- Extend EPED1 to predict pedestal structure based on $n_{e,sep}$ and $T_{e,sep}$ rather than $n_{e,ped}$

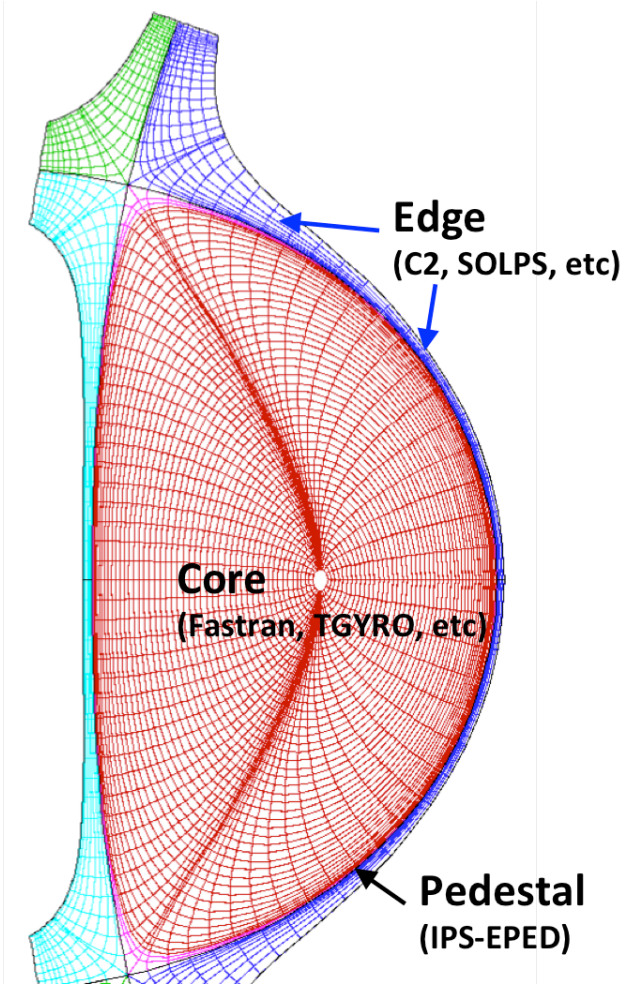


  - Can we use ETG critical gradient model?
  - Something else needed?

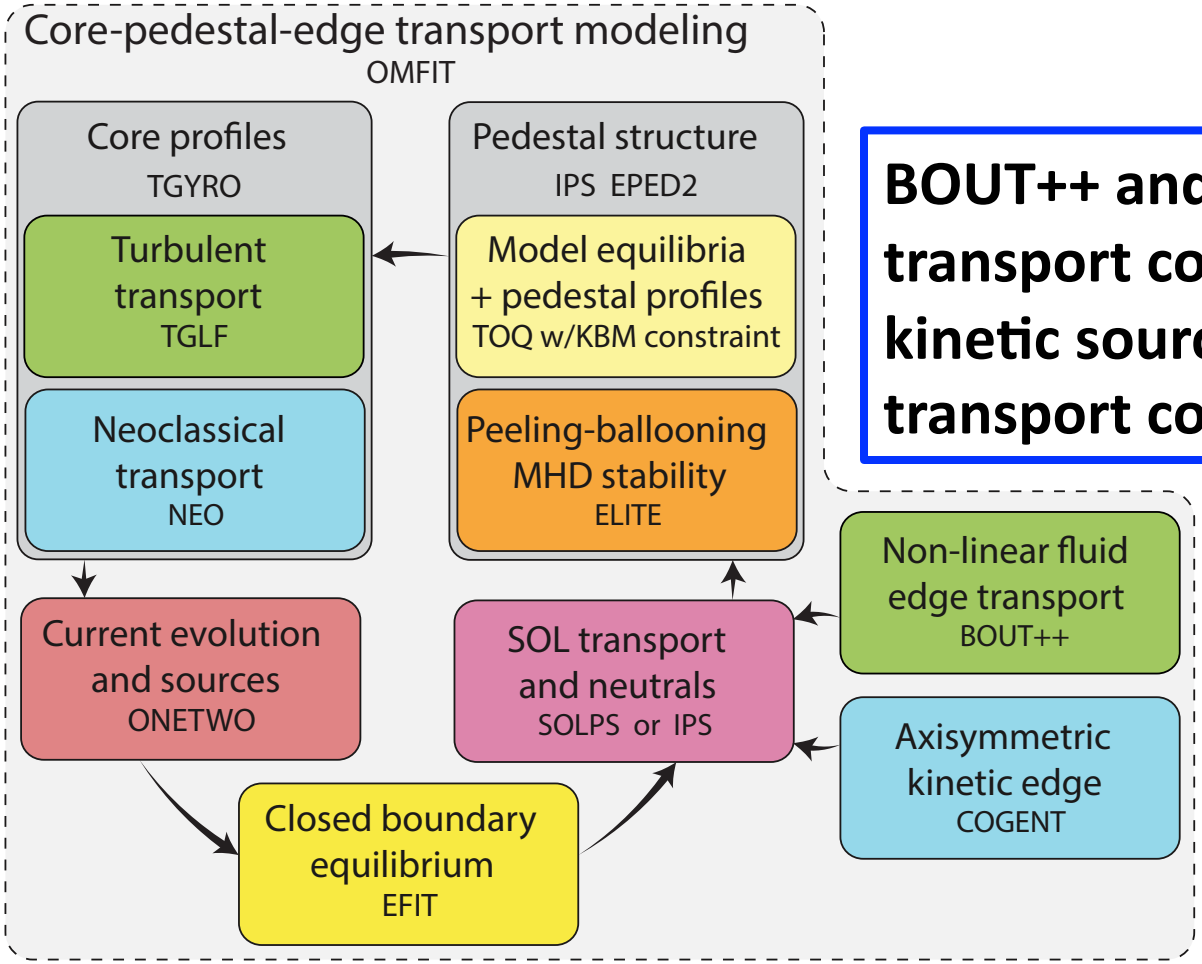
- Start with a simple 2-point SOL model to estimate $n_{e,sep}$ and $T_{e,sep}$

Longer term: replace 2-point model with SOLPS+COGENT+BOUT++

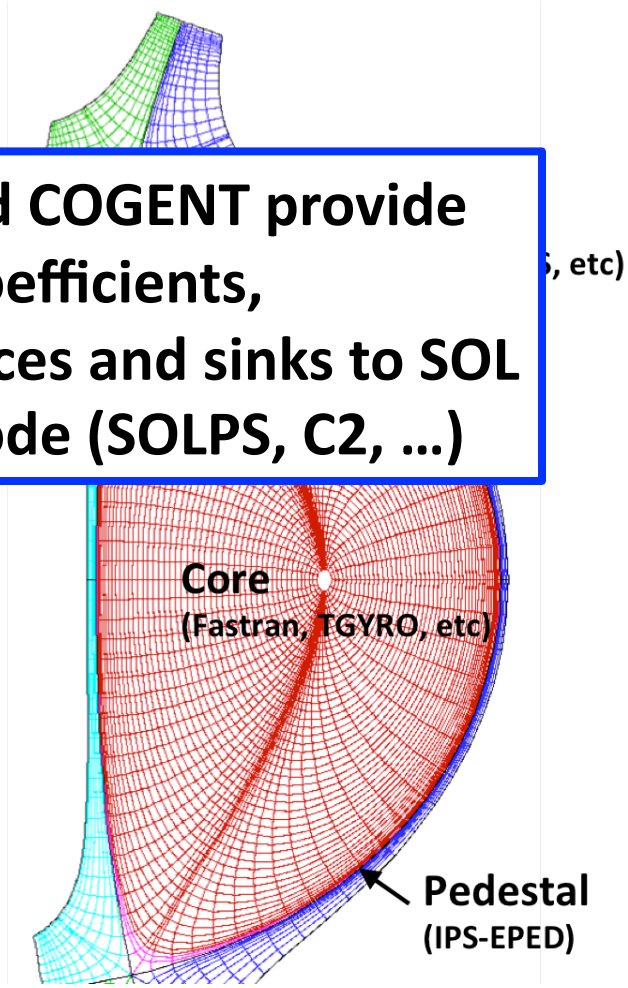




Longer term: replace 2-point model with SOLPS+COGENT+BOUT++



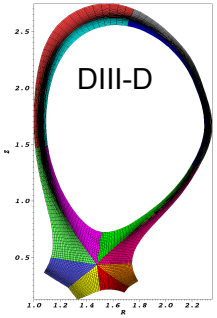
BOUT++ and COGENT provide transport coefficients, kinetic sources and sinks to SOL transport code (SOLPS, C2, …)

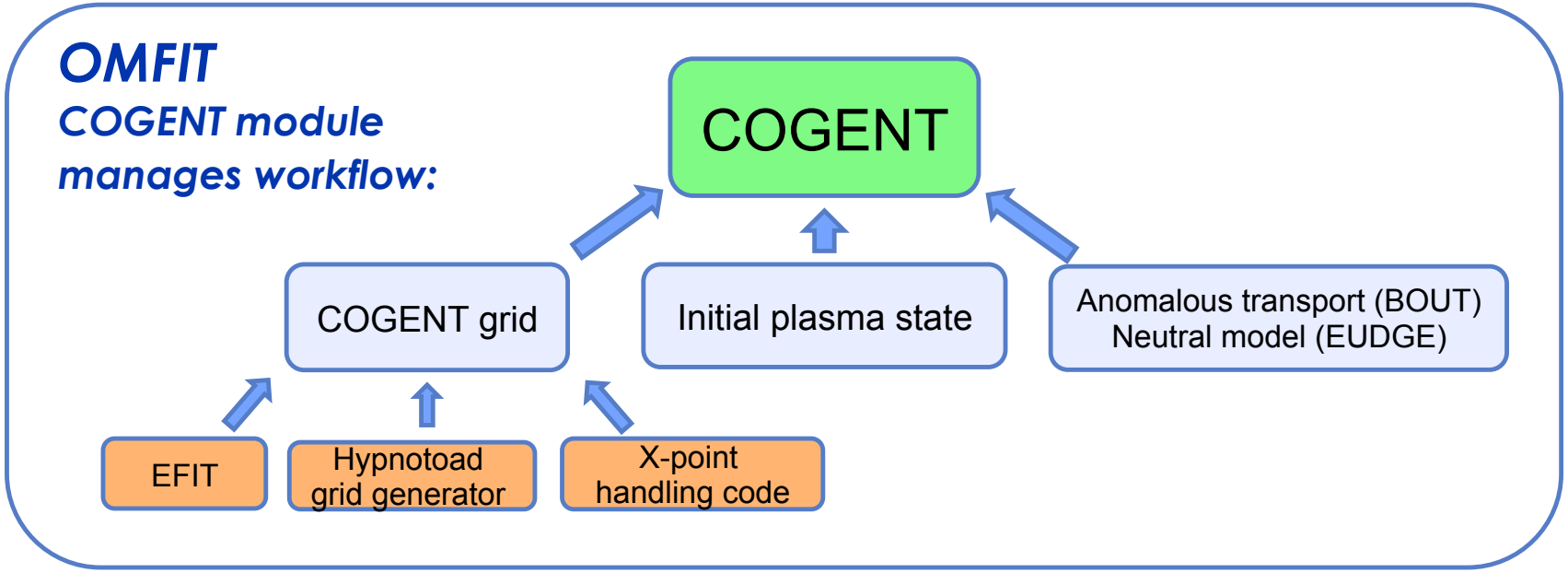


COGENT module structure

COGENT code

- Full-F gyro-kinetic equation coupled to the long wavelength gyro-Poisson equation in a divertor geometry. Presently 4D (axisymmetric), 5D in progress
- 4th order finite-volume discretization combined with the mapped multiblock approach to handle the X point with high accuracy



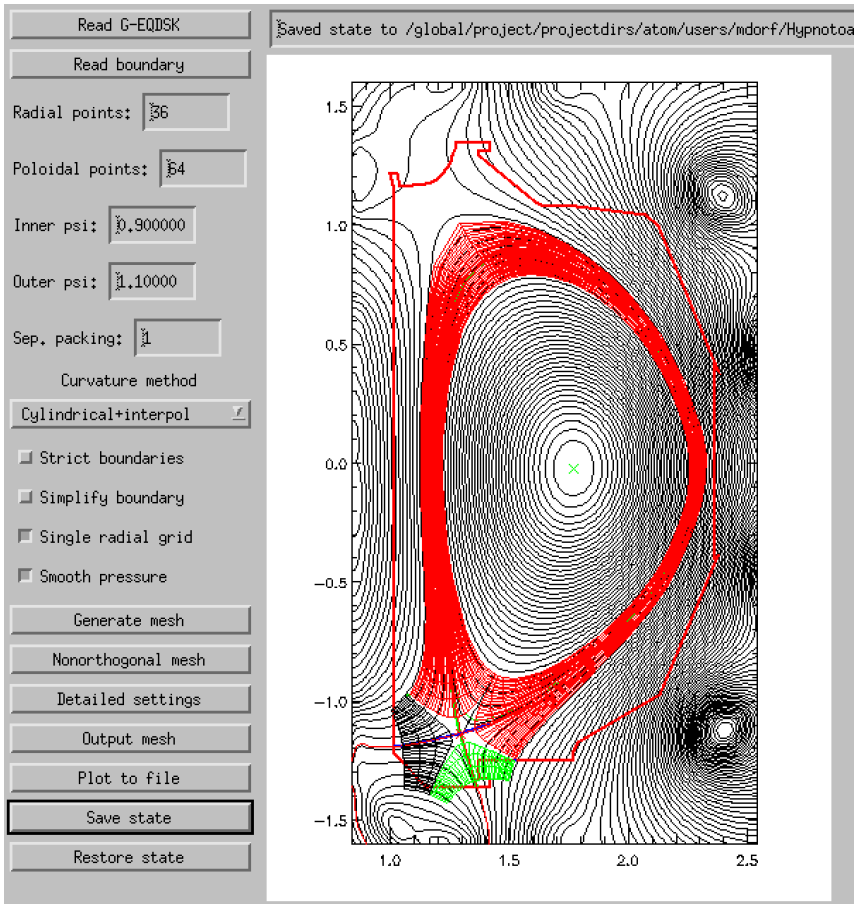


Creating COGENT grid (Already in OMFIT)

Hypnotoad: A field-aligned structured mesh generator

Developer: B.D.Dudson (University of York)

Publically available on github / used by the BOUT community



Step 1 (done)

Run Hypnotoad

- *Input: EFIT g-files*
- *Output: grid cell-centers / magnetic field data*

Step 2 (done)

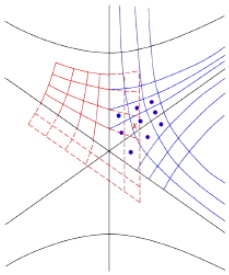
Run the Hypnotoad -based grid postprocessor

- *Output: COGENT-structured grid vertices*

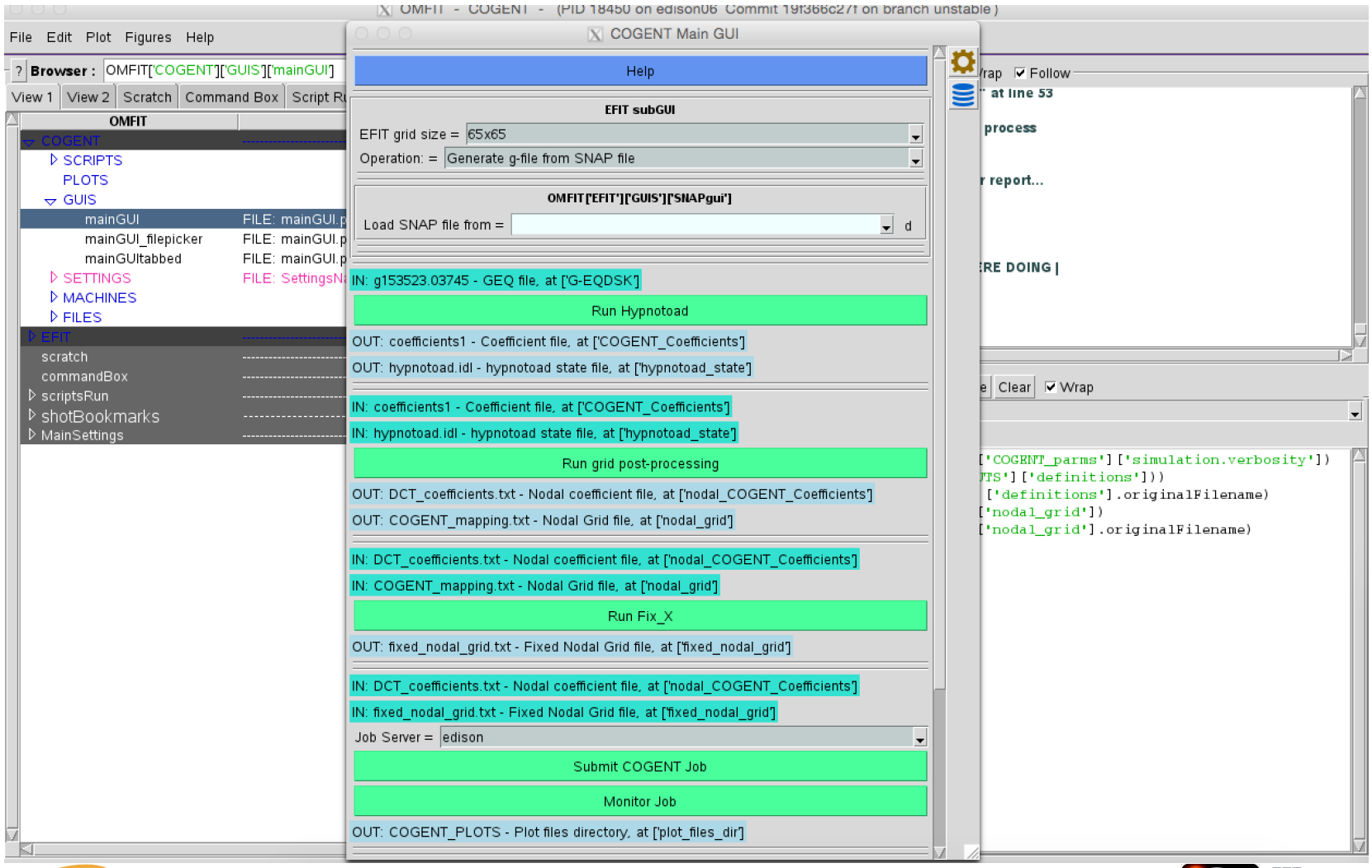
Step 3 (in progress)

Run the grid-smoothing tool (de-aligns grid near the X-point)

- *Manual (developed)*
- *Automatic (in progress)*



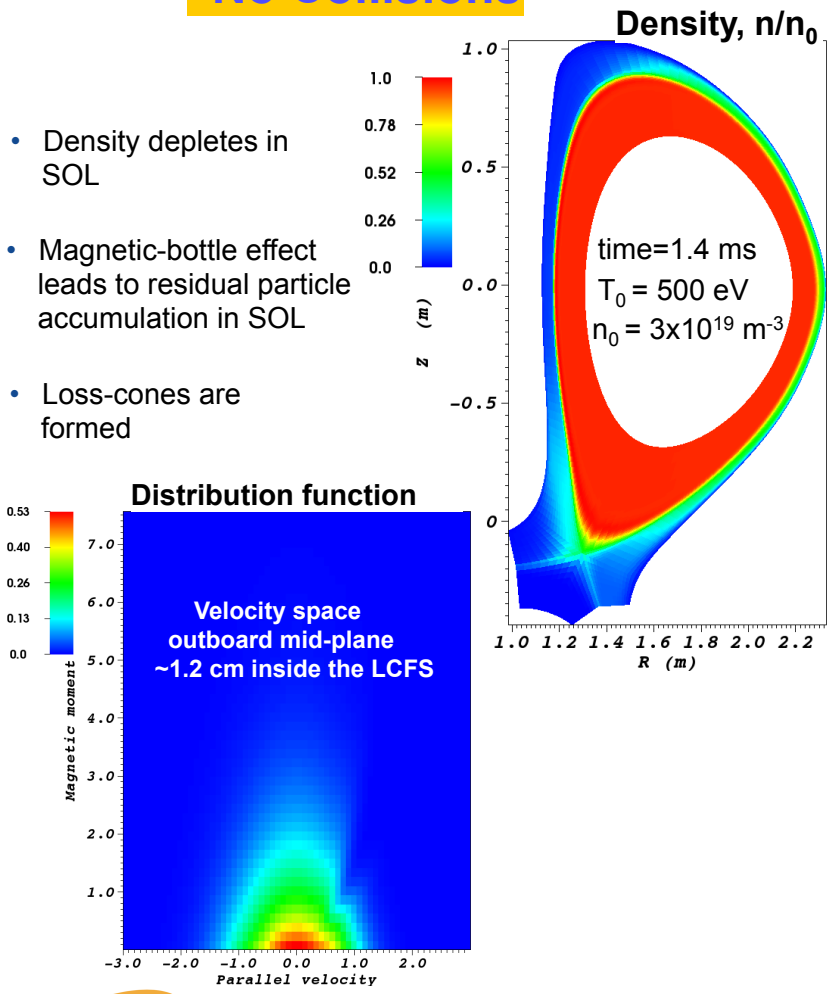
COGENT module GUI (screen shot)



Running COGENT for the AToM show case: Ion orbit loss test

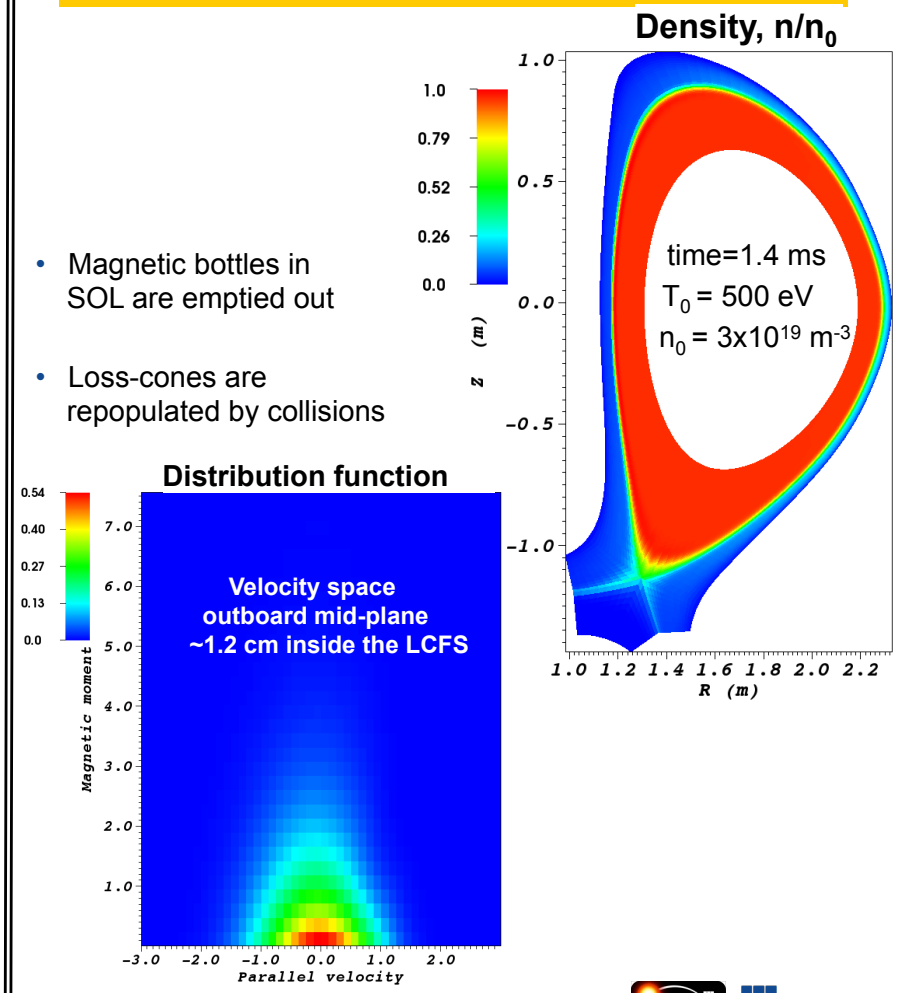
No Collisions

- Density depletes in SOL
- Magnetic-bottle effect leads to residual particle accumulation in SOL
- Loss-cones are formed

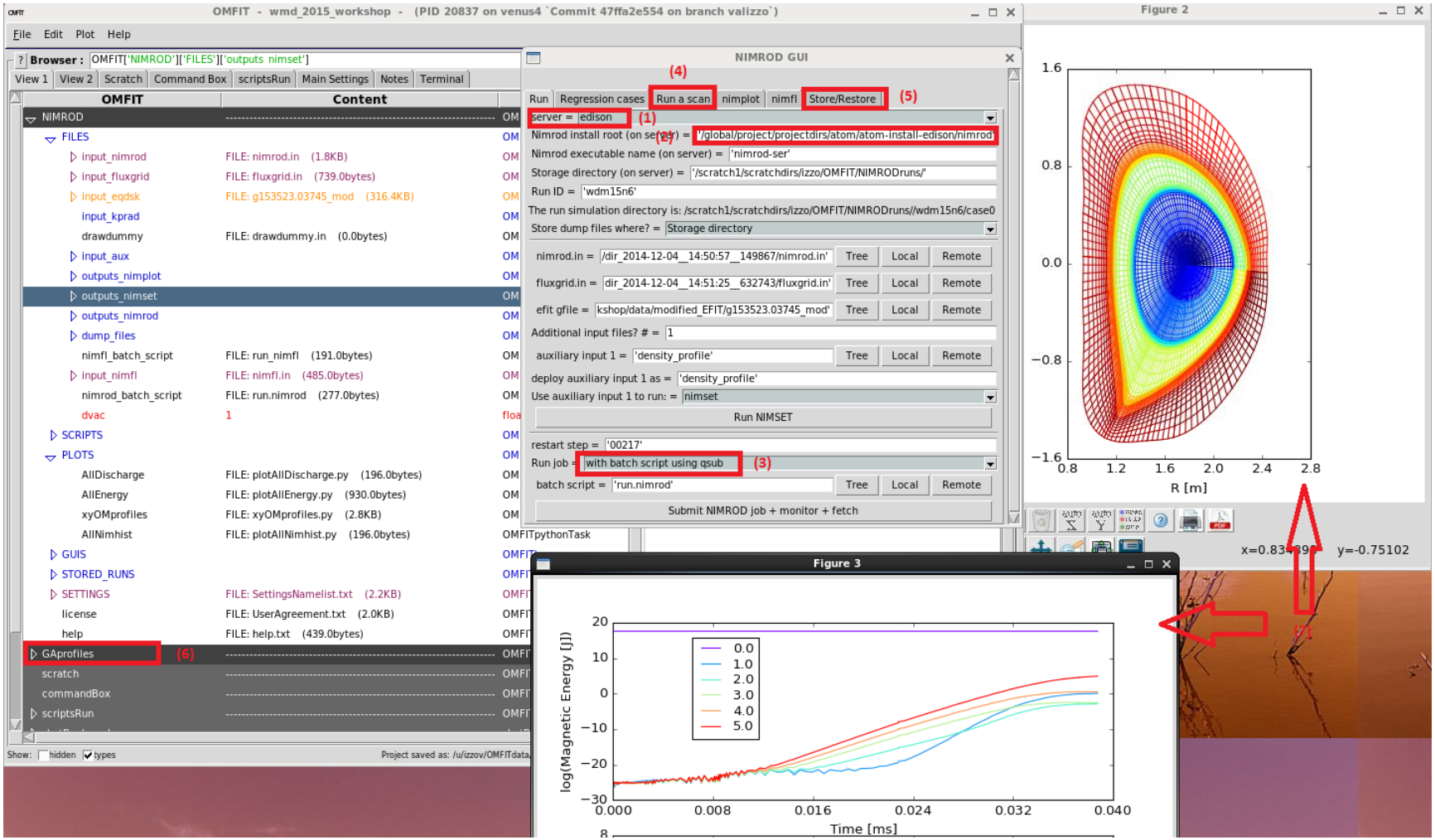


Full nonlinear FP collisions

- Magnetic bottles in SOL are emptied out
- Loss-cones are repopulated by collisions



NIMROD module available, building workflows for remote code execution, data analysis, archiving at NERSC



Other BOUT++ opportunities

- Verification and Validation
  - Ability to execute and analyze multiple codes across different local and remote platforms along with experimental data at multiple machines
- Help extend pedestal models to predict rotation at top of pedestal
  - Big issue for predicting ITER/reactor plasmas- any hope of getting better performance from rotation
- Long term AToM goal: incorporate ELM dynamics into time-dependent predictive integrated modeling
  - How does this work? What would it look like?
- Intermediate-scale core turbulence model
  - Replace GYRO or TGLF instances with set of 3+1 BOUT++ gyrofluid runs
  - Investigate multiscale ITG/ETG?
  - Develop ETG subgrid models? Overlaps with edge needs