Overview of the OMFIT framework

ITM code camp, Lisbon Portugal

O. Meneghini

Oak Ridge Associated Universities
General Atomics theory group

Nov 20^{th} 2013

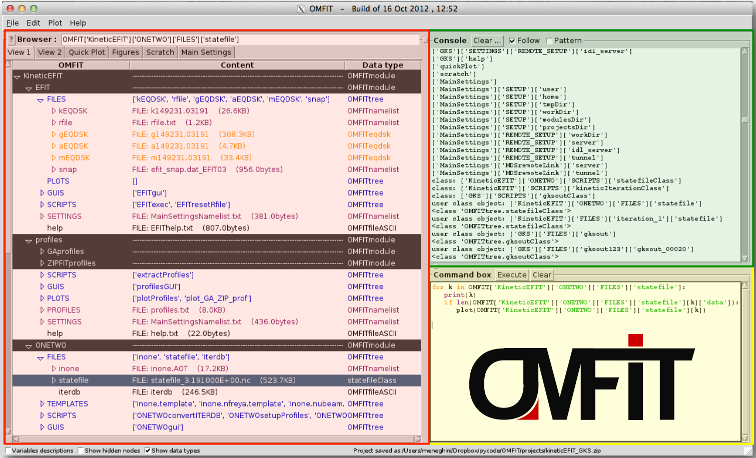
DIII-D NATIONAL FUSION FACILITY

GENERAL ATOMICS

Oak Ridge Institute for Science and Education

One Modeling Framework for Integrated Tasks

"A comprehensive framework designed to facilitate experimental data analysis and enable integrated simulations"



Main idea: collect data from different sources into a single, self-descriptive, hierarchical data structure (OMFIT tree)

Similar to the ITM CPO...

Unified structure enables communication among different codes

...but free-form

With no a-priory decision of what is stored and how

(like MDS+ or the filesystem on your laptop)

It's the difference between a **top-down** and a **bottom-up** approach

How could this possibly work!? Actually...

- Read/write of few scientific data formats enables interaction with many different codes
- Often codes need to exchange small amount of data
- Exploit existing integration efforts:
  - many codes already accept each others files
  - conversion utilities are aready available
- No need to modify codes and their I/O
  - No burden on developers of individual codes
  - Effort done by users interested in integrating
- Skips alltogether arguments about which data structure to use
- Does not exclude use of standard data structures when available

Other important characteristics of the OMFIT framework

- **Component based approach** and **Python scripting** allow building of complex workflows
- **Graphical user interfaces** ease execution of each component and their interaction
- **Power users retain full control** of code I/O files and execution
- **Local/remote** and **serial/parallel** codes execution
- **Lightweight, pure-Python framework** easy to install, maintain and expand
- **Integrated with experimental databases** for data analysis, generation of codes inputs and validation
- **Collaborative environment** promoting sharing code and testing
- Addition/improvement of features and components is **problem-driven**

OMFIT provides an increasing list of ever-improving modules

Easy to support new codes, especially if they use standard file formats

Equilibrium

EFIT

VARYPED

CORSICA

Experimental analysis

PROFILES

SCOPE

Gyro-kinetic

GYRO

TGLF

GKS

Heating

GENRAY

TORBEAM

NUBEAM

MHD

M3DC1

BOUT++

Transport

ONETWO

GCNMP

NEO

TGYRO

Stability

DCON

GATO

PEST3

ELITE

RMP

NTV

FLUTTER

OMFIT

OMFIT was used as part of many integrated modeling studies presented at 2013 APS

F. Turco *MARS-K Modeling Validation for Rotation and Fast-Ions Impact on RWM Stability in DIII-D Plasmas*

B. Grierson *Interpretive and Predictive Transport Analysis in DIII-D ITER Baseline and QH-Mode Discharges*

X. Wang *Off-diagonal Terms Connection Between Particle and Momentum Transport in DIII-D Plasma*

S. Mordijck *Changes in Particle Transport as a Function of Collisionality and Rotation*

C. Holland *Validation Metrics for Improving Our Understanding of Turbulent Transport* **(invited)**

C. Luna *Prediction of Transport Phenomena with Neural Networks*

S. Smith *Magnetic Flutter Plasma Transport Induced by 3D Fields in DIII-D* **(invited)**

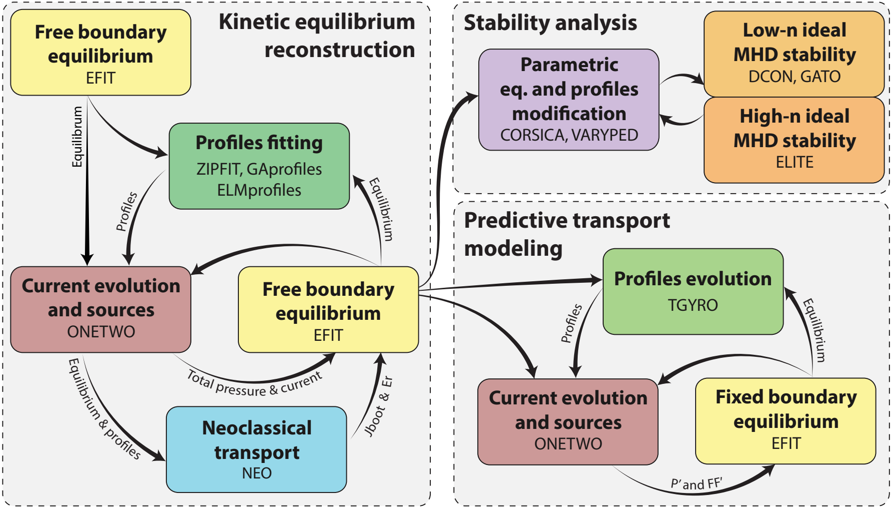
C. Chrystal *Testing Neoclassical and Turbulent Effects on Poloidal Rotation in the Core of DIII-D* **(invited)**

E. Bellie *Neoclassical Flows, Transport, and Non-Axisymmetric Effects in the Tokamak Plasma Edge* **(invited)**

A. Garofalo *Modeling of Steady-state Scenarios for the FNSF, Advanced Tokamak Approach*

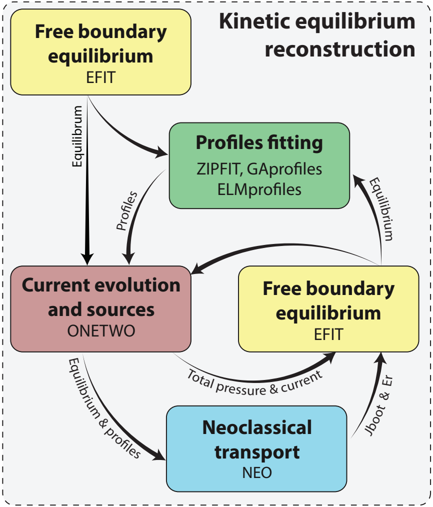
OMFIT manages the complexity of many codes interacting with each other in complicated workflows

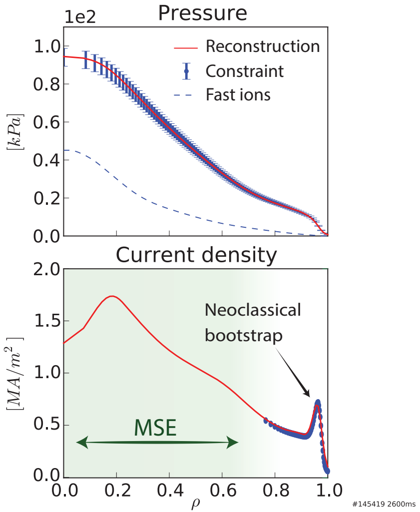
Routinely used for DIII-D equilibrium, stability and transport analyses



OMFIT streamlines kinetic equilibrium reconstructions which are at the foundation of most physics studies

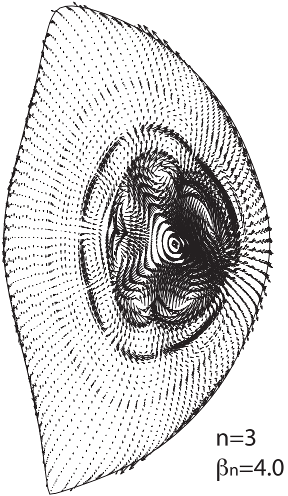
Measurements and models (J_b, NBI, ECH) used to constrain P and J

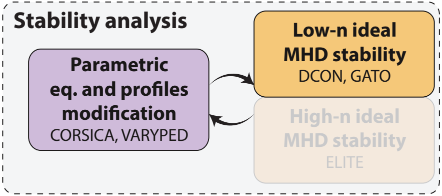


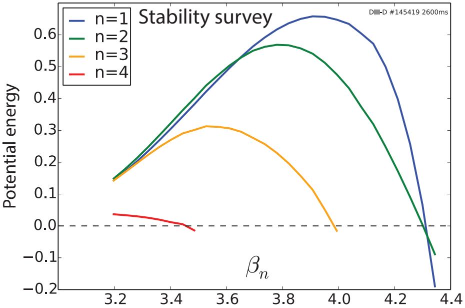


OMFIT can efficiently investigate ideal MHD stability of the core plasma

DCON finds unstable β_n, growth rate and mode structure with GATO

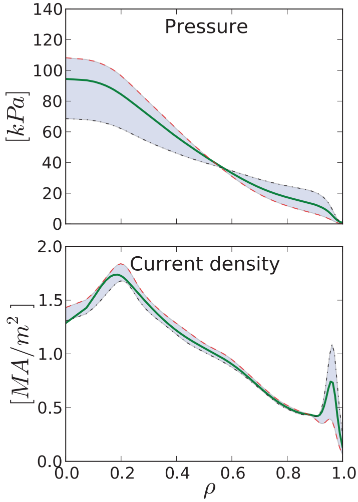


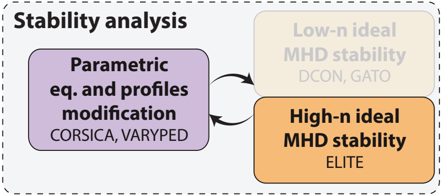


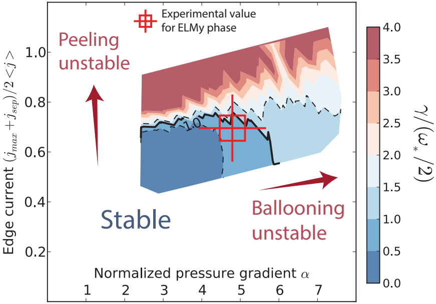


OMFIT can conveniently generate edge stability diagrams

Peeling-ballooning stability strongly depends on edge ∇P and ∇J

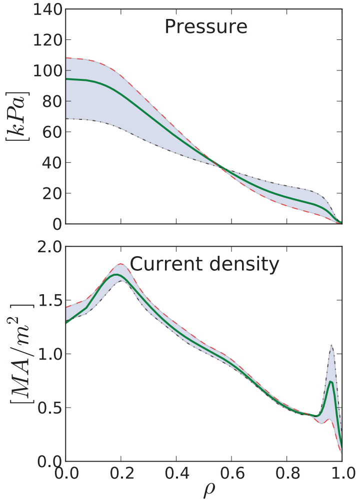


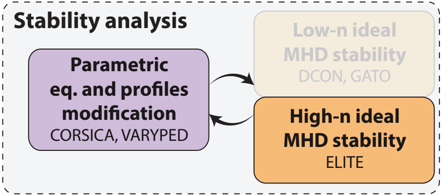


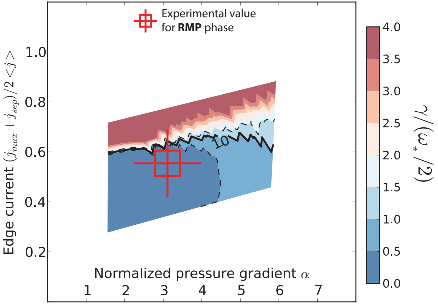


OMFIT can conveniently generate edge stability diagrams

Peeling-ballooning stability strongly depends on edge ∇P and ∇J







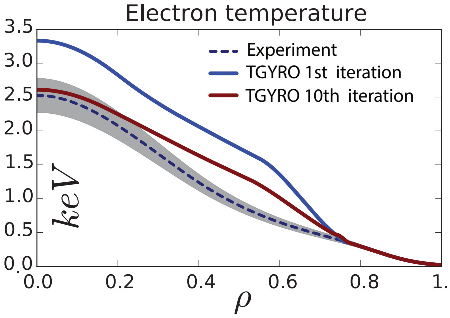
Self-consitent steady-state predictive models are efficiently obtained as an extension of the kinetic EFIT workflow

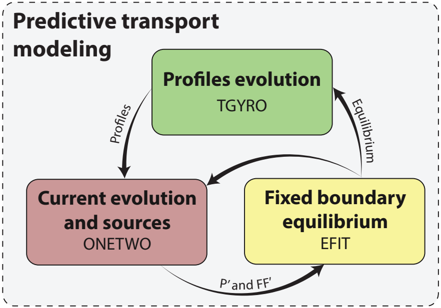
Substitute: kinetic profiles **fitting** → kinetic profiles **prediction**

TGYRO efficiently solves the steady-state transport equation:

$$\Gamma_{neo}(x) + \Gamma_{turb}(x) = \Gamma_{target}(x) = \int_0^x V'(r)\, S(r)\, dr$$

- Neoclassical from NEO and turbulent from either TGLF or GYRO





The next step: integrating OMFIT with ITM

Strategy:

1. Enable manipulation of CPO data
   - R/W of data from/to the UAL using available Python bindings

2. Execution of kepler actors
   - “standalone” kepler actors use text files for I/O

Achieved so far:

- Wrote OMFIT Python class for read/write of I/O files of standalone kepler actors
- Can automatically create OMFIT-ITM interface and execute standalone actor
- Can use UAL but little more work is needed for seamless integration in the OMFIT tree

Live demo

PLEASE WEAR YOUR 3D GLASSES

Conclusions

- Comprehensive OMFIT framework developed and used to support DIII-D with many applications

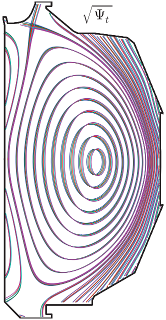
- Integration with ITM-UAL will allow seamless execution of the codes adhering with the ITM-TF standards

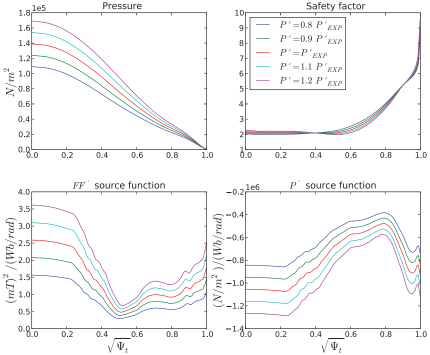
- OMFIT-ITM integration prepares ground for GA integrated modeling of ITER

Extra slides

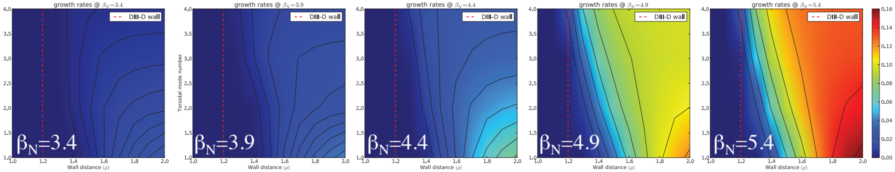
Survey of ideal MHD stability at increased β_n with GATO

Pressure scanned by scaling of P' and ideal MHD stability evaluated for different toroidal mode numbers n and wall distances (conformal wall)



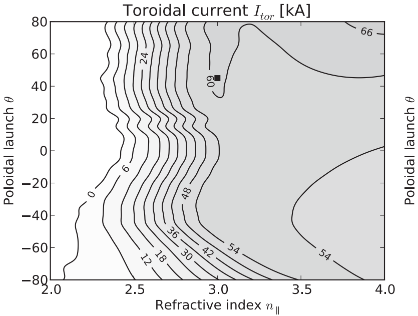


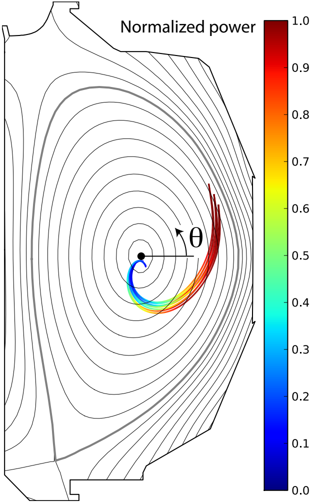
220 GATO simulations run 20 at a time in parallel on 3 different remote machines



Evaluation of whistler waves (also known as ‘helicons‘) current drive efficiency and location with GENRAY

- DIII-D target discharge #122976 with $\beta_n = 3.9$ (high β needed for absorption)
- Automated scan of launched $n_{\parallel}$ and poloidal angle θ of wave injection
- Target compares favorably $(60\ kA/MW)$ with respect to EC $(16\ kA/MW)$ and NBI $(26\ kA/MW)$

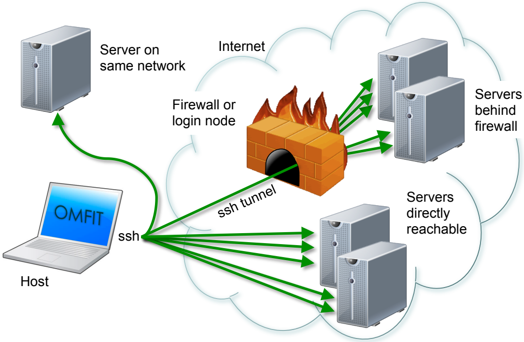




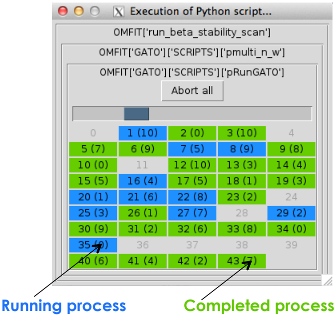
$500\ MHz \quad n_{\parallel} \sim 3.4 \quad \theta \sim 40\ deg$

High level Python APIs allow users to: execute tasks remotely and in parallel

- Seamless execute codes and and manage files remotely
  - Let codes run codes where they already work!
  - Machine running OMFIT directs and stores data in OMFIT tree
- Parallel execution of the same task with different input parameters, on multiple remote machines
- Real-time monitoring of local / remote and serial / parallel tasks



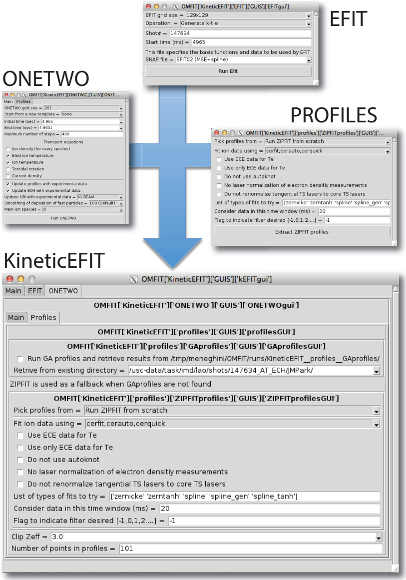
Monitor progress of parallel execution



High level Python APIs allow users to: create Graphical Users Interfaces (GUIs)

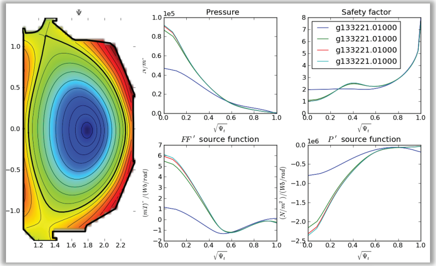
User GUIs speed-up routine analysis and hide many of the underlying complexities to inexperienced users

- GUIs are python scripts and are created by users themselves
- Quick and easy! For each GUI entry need to specify the OMFIT tree location associated with it
- GUIs can be nested to create comprehensive GUIs, while ensuring consistency

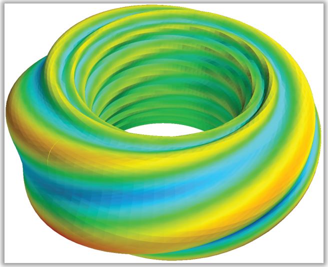


Quickly visualize data in the OMFIT tree or create publication quality graphics with Python scripts

Kinetic EFIT iterations



M3D-C1 simulation of RMP pressure perturbation



1D/2D arrays are (over)-plotted with the push of a button

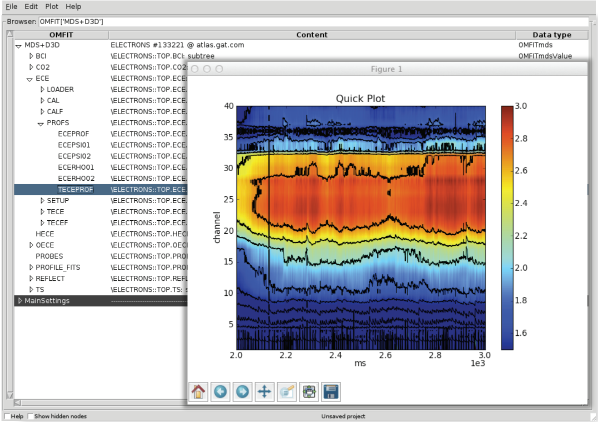
- Inspect inputs/outputs of different analyses / codes / iterations / ...
- Plots are interactive and can be customized (à la MATLAB)

More sophisticated plots are scripted in Python

- Matplotlib library very similar to MATLAB and IDL plot commands
- Plotting scripts can be assigned to specific objects

Access MDS+ data, PTDATA signals and D3DRDB tables directly from the OMFIT tree

- Browse, search, plot and manipulate experimental data interactively or in scripts
- Creation of codes inputs: profiles, power, angles,..
- Validation: compare modeling results with experiments



MDS+ traverser