Transient-Free Operations With Physics-Based Real-time Analysis and Control

Egemen Kolemen

David Eldon, Alex Glasser, Alan Glasser, John Ferron, Bill Eggert, K. Burrell, David Humphreys, Orso Meneghini

Mechanical & Aerospace Engineering

Jointly appointed with the Andlinger Center for Energy and the Environment

and the Plasma Physics Laboratory (PPPL)

PRINCETON UNIVERSITY

PPPL PRINCETON PLASMA PHYSICS LABORATORY

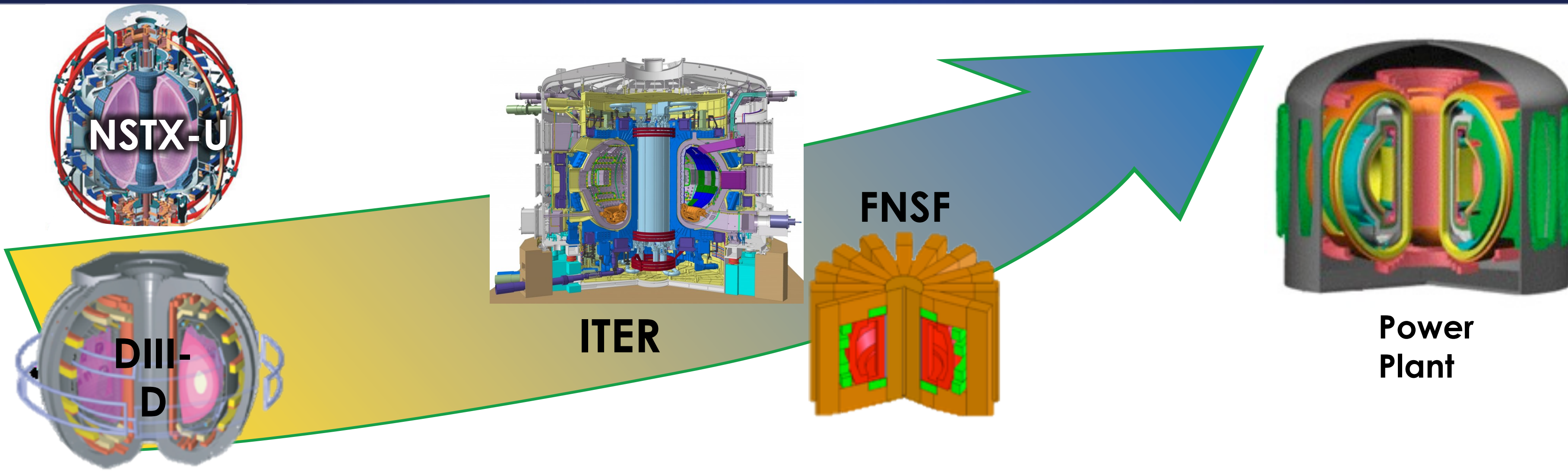
DIII-D NATIONAL FUSION FACILITY

1611-00446/1

This work supported in part by U.S. Department of Energy under DE-SC0015878 and DE-FC02-04ER54698.

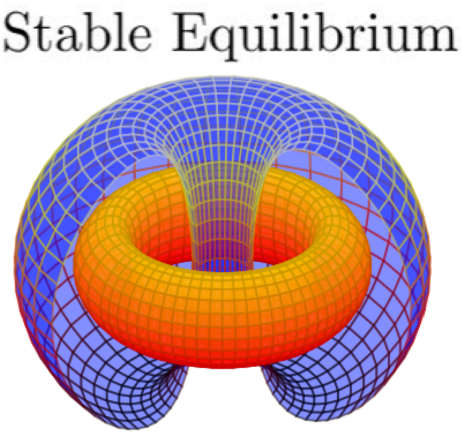
andlinger center for energy+the environment

Real-time Stability Calculations for Plasma Control and Disruption Avoidance



- For cost-effective commercial fusion power plants to be feasible, we need to **operate close to stability limits**
- There must also be almost **no disruption** in fusion power reactors and very few disruptions in ITER
- This necessitates **real-time stability analysis** for plasma control and disruption avoidance

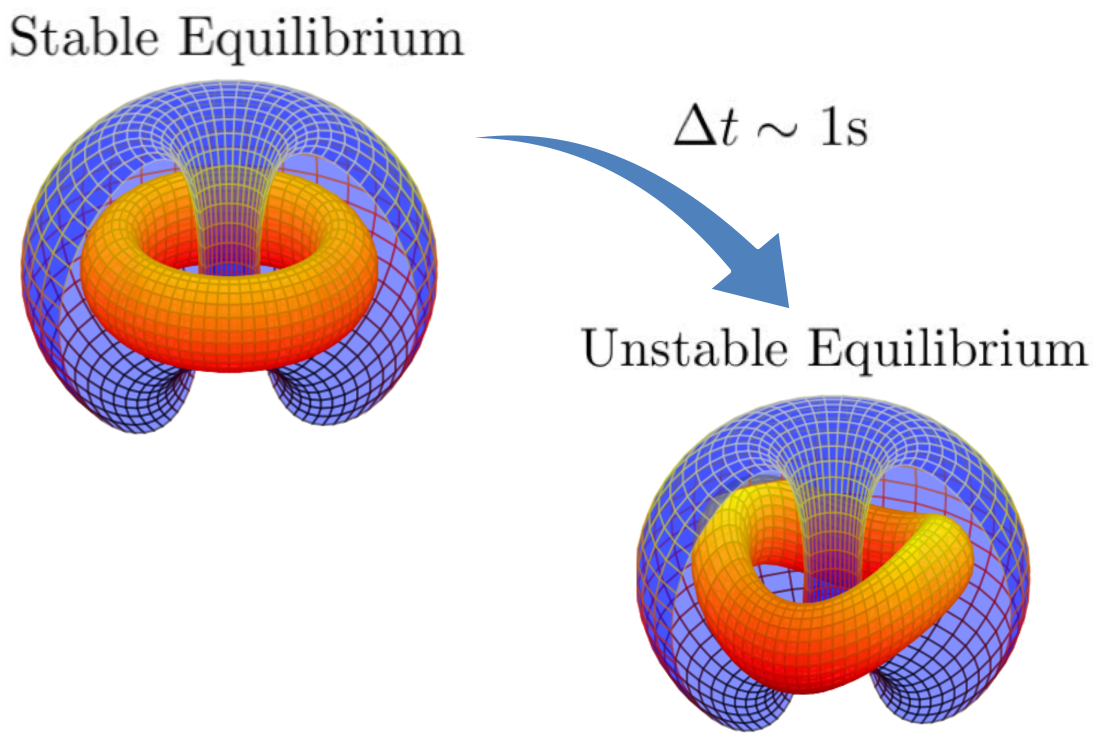
Steer the plasma away from unstable equilibria with Real-time Stability Analysis and Control



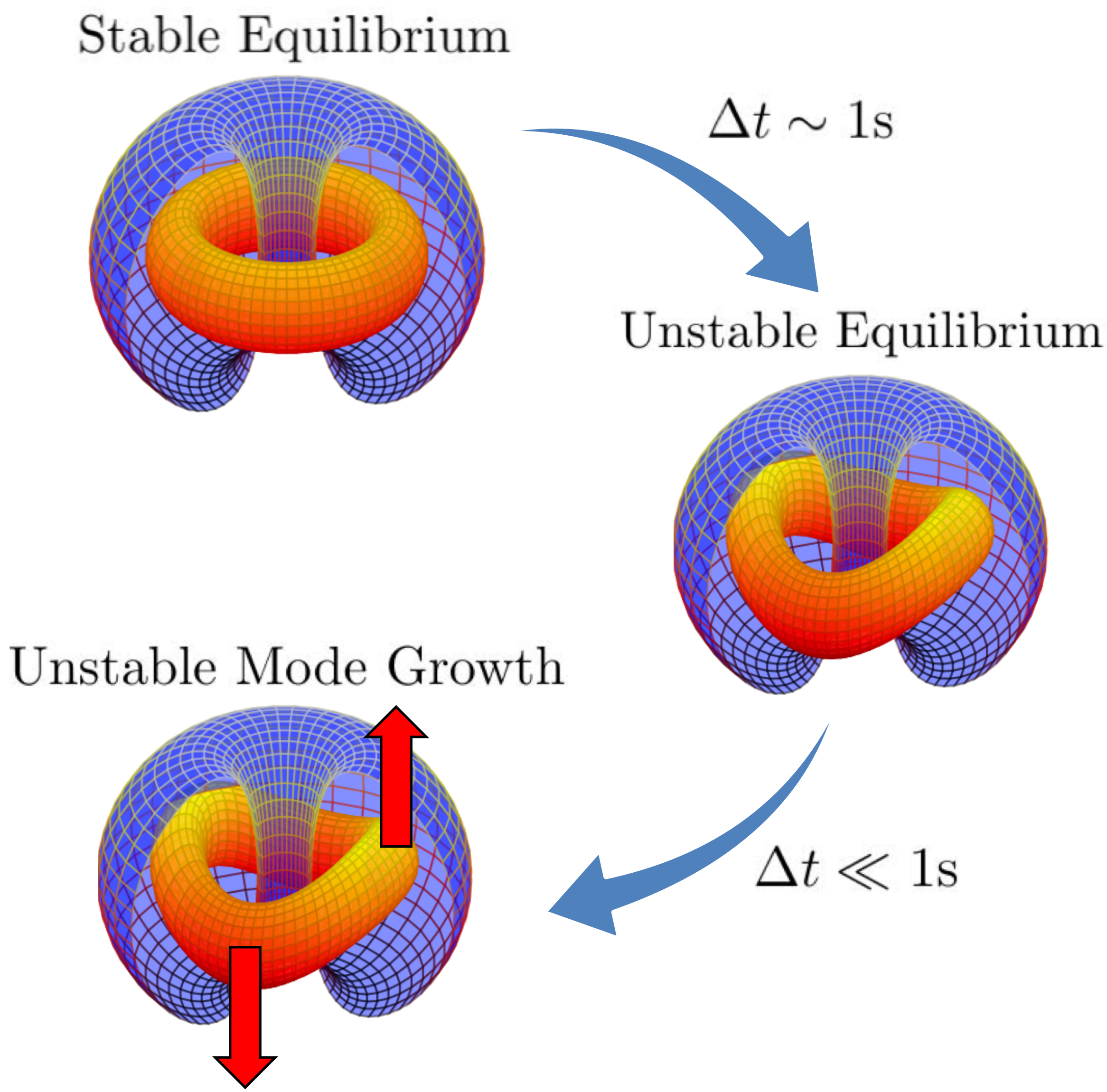
DIII-D NATIONAL FUSION FACILITY

1611-00446/3

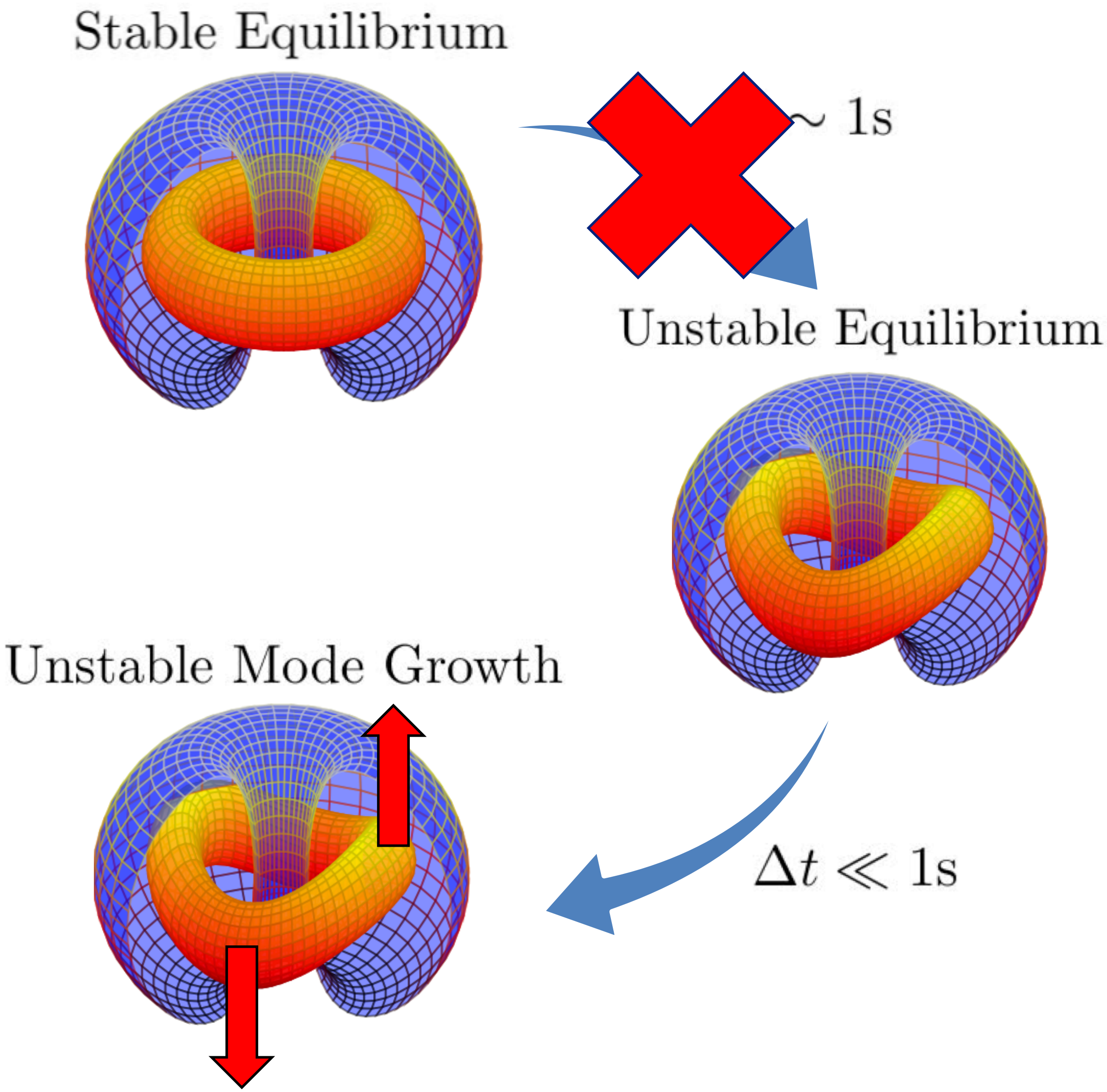
Steer the plasma away from unstable equilibria with Real-time Stability Analysis and Control



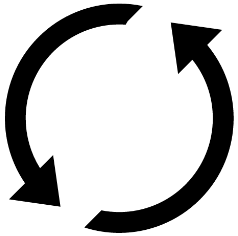
Steer the plasma away from unstable equilibria with Real-time Stability Analysis and Control



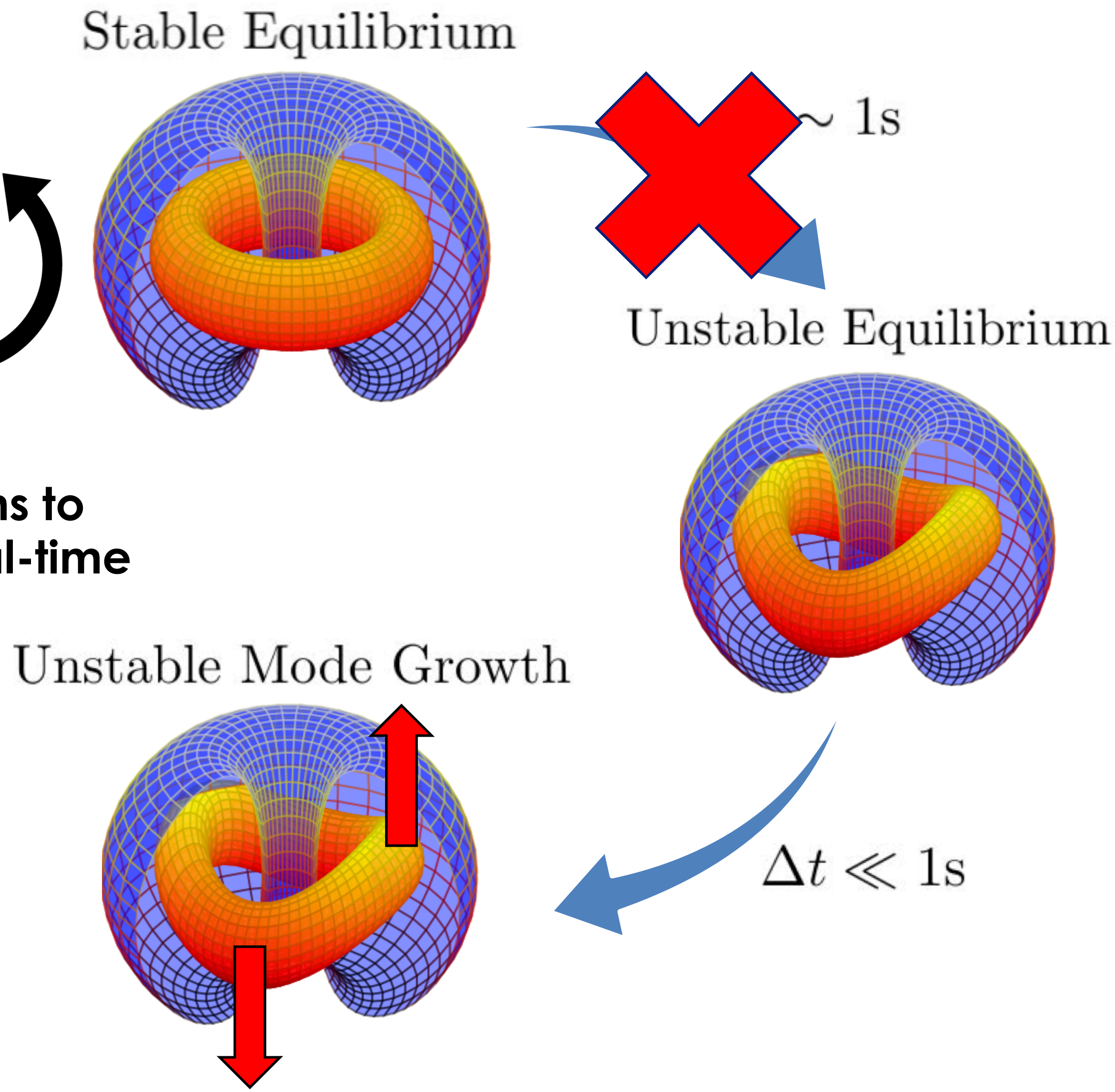
Steer the plasma away from unstable equilibria with Real-time Stability Analysis and Control



Steer the plasma away from unstable equilibria with Real-time Stability Analysis and Control



- **Develop fast algorithms to analyze stability in real-time**



Real-time Computational Analysis for Fusion Adds New Challenges

- In order to produce real-time stability computation useful for control, we need to:

1. >>99% reliability
   - Use a well known analysis and tested codes ➔ Ideal MHD
2. Take the physicist out of the loop
   - Automated Real-time Kinetic Equilibrium reconstruction

1. Real-Time Kinetic Equilibrium Reconstruction

2. Real-time Stability Calculations

1611-00446/9

Real Time Kinetic Equilibrium Reconstruction Will be Implemented by Adding P and J Constraints to EFIT

- EFIT solves the Grad-Shafranov Equation

$$\Delta^* \psi = -\mu_0 R^2 p' - \mu_0^2 f f'$$

$$J_R = -\frac{1}{R}\frac{\partial f}{\partial Z}$$

$$J_Z = \frac{1}{R}\frac{\partial f}{\partial R}$$

- ψ constrained by magnetics
- J constrained by magnetics and MSE

Real Time Kinetic Equilibrium Reconstruction Will be Implemented by Adding P and J Constraints to EFIT

- **EFIT solves the Grad-Shafranov Equation**

$$\Delta^*\psi = -\mu_0 R^2 p' - \mu_0^2 f f'$$

$$J_R = -\frac{1}{R}\frac{\partial f}{\partial Z}$$

$$J_Z = \frac{1}{R}\frac{\partial f}{\partial R}$$

- **ψ constrained by magnetics**
- **J constrained by magnetics and MSE**

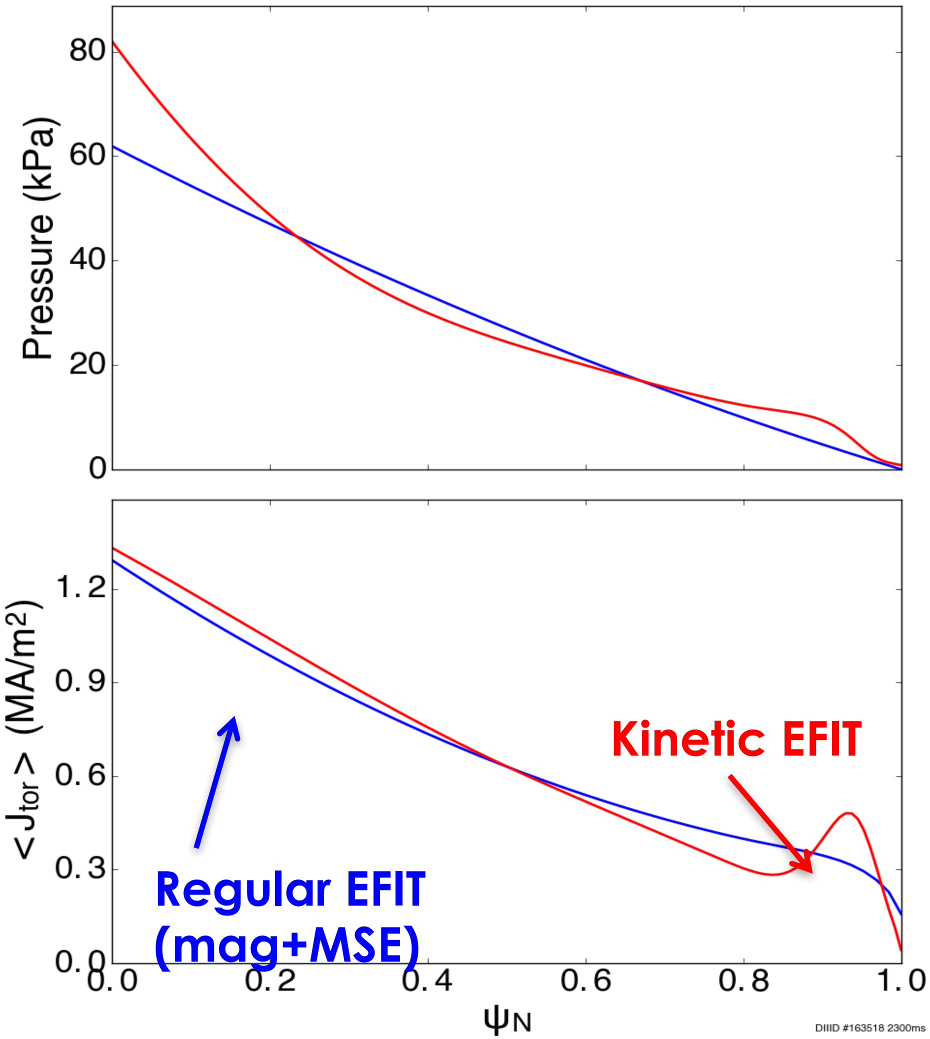
New
- **Additional constraints in a kinetic EFIT:**
  - p is constrained by TS, CER, and fast ion calculations
  - J is further constrained by $J_{BS} + J_{OHM} + J_{ECCD}$ calculations

With David Eldon and John Ferron

DIII-D NATIONAL FUSION FACILITY

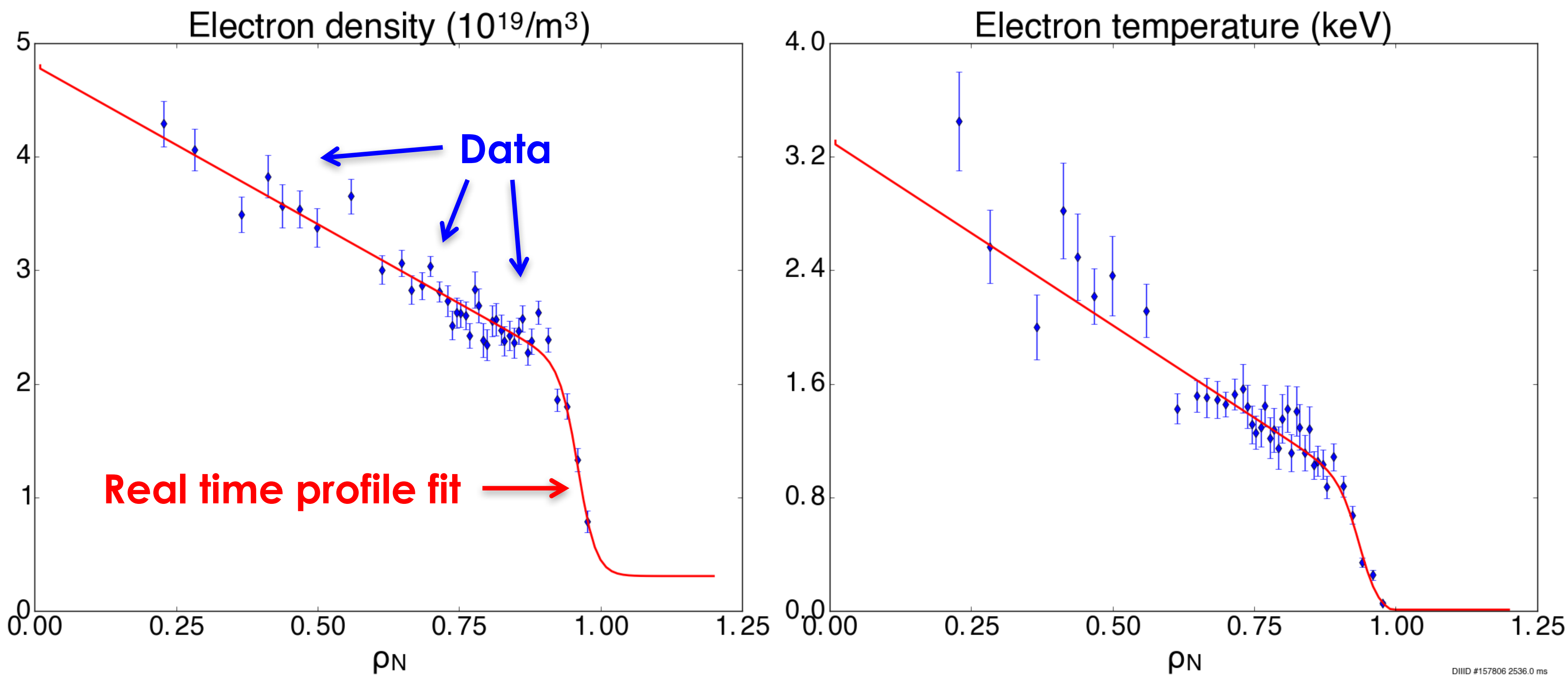
1611-00446/11

Real Time Kinetic Equilibrium Reconstruction Difference between regular versus kinetic-EFIT



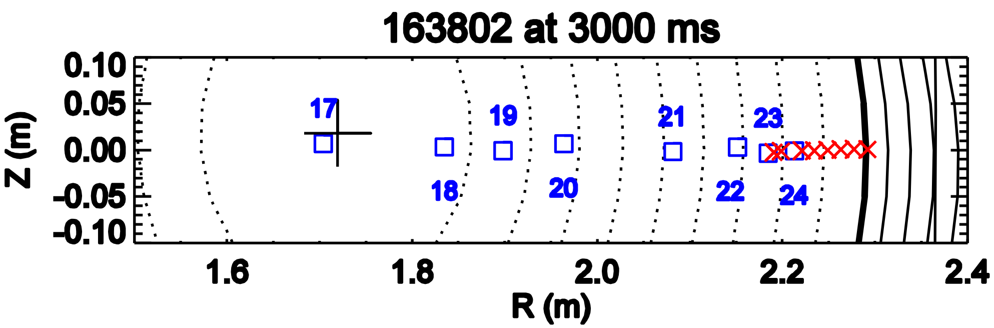
- **Pedestal pressure gradient and the resulting bootstrap current introduces errors throughout the profiles**
- **Critical to constrain equilibria for stability analysis**

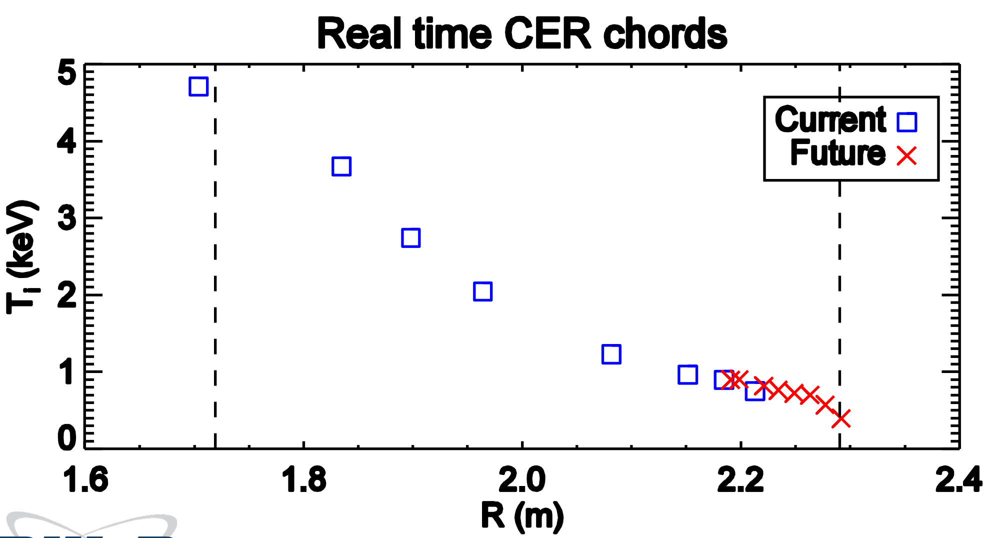
Real-time Thomson Working at DIII-D



- We acquire the Thomson data in real-time.
- Calibrate and fit it ➔ Input to auto-kefit

RT-CER constraints on the Current and Pressure



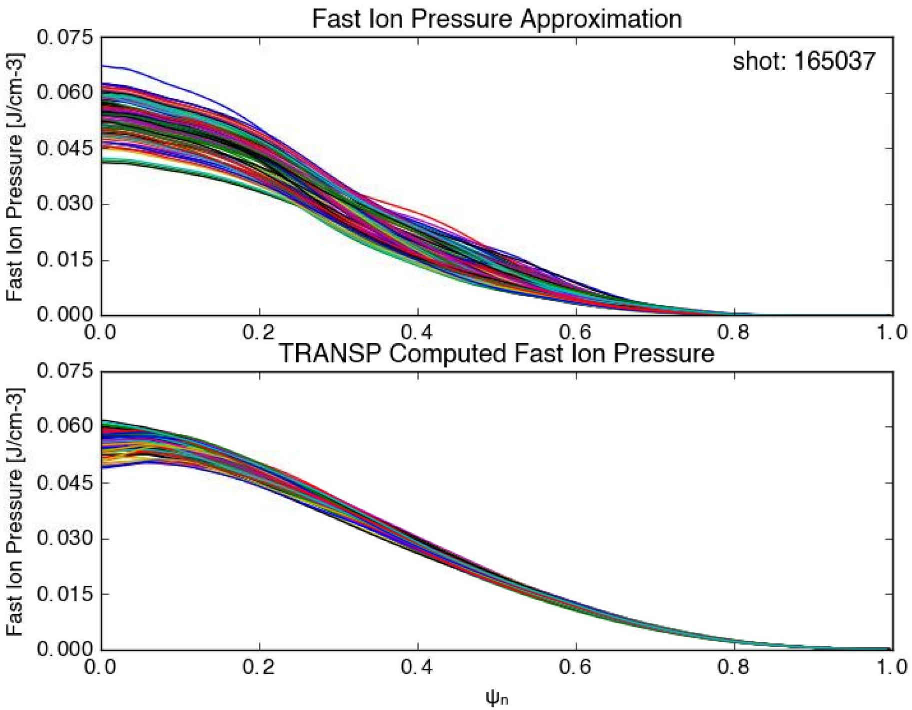


□ Core CER channels already acquired

✕ Edge CER chords for pedestal are added this year

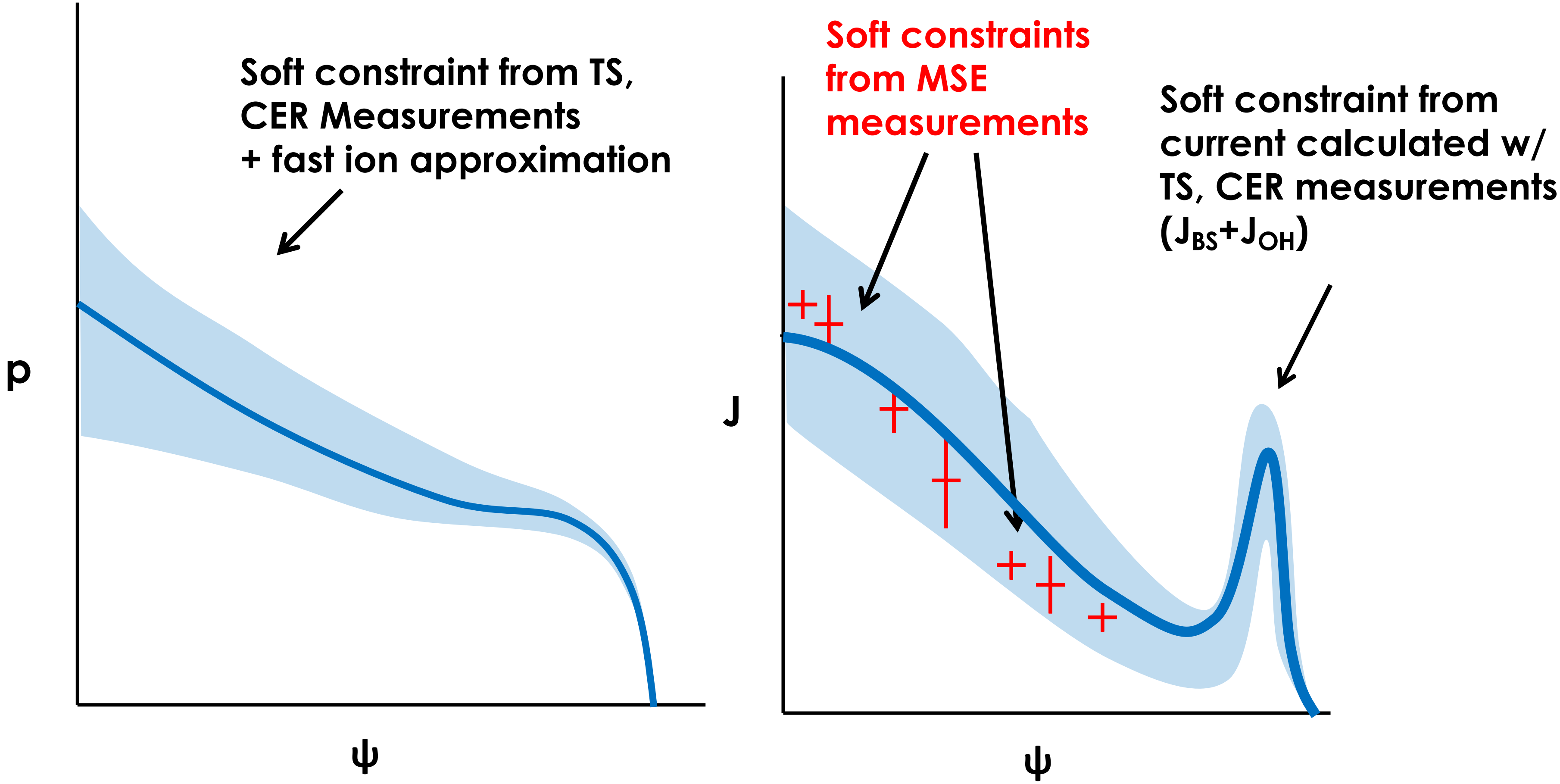
- Calibration and fitting for getting Ion temperature and density are to be tested this run campaign

Fast Ion Approximation Tool (FIAT) provides quick access to fast ion profiles (Bill Eggert)



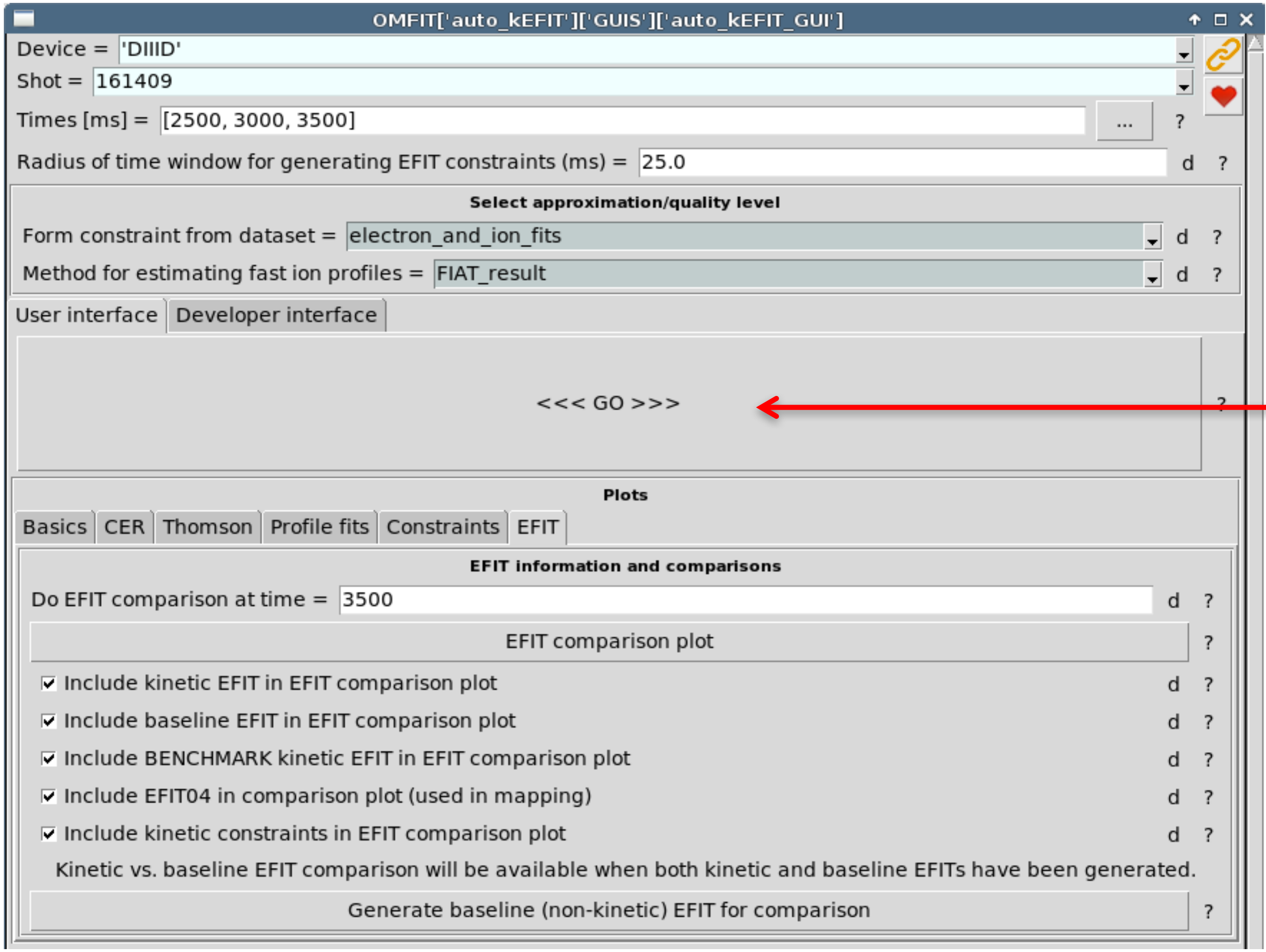
- **Fast ion pressure:**
- ***FIAT* surveys 1000s of shots**
  - Read fast ion profiles from ONETWO results
  - Fit fast ion profiles w/ Gaussian & record amplitude and width
- **➔ Fast function for estimating fast ion profiles then can be used in real-time**

Add soft constraints on the Current and Pressure to RT EFIT



- RT-Thomson (two years ago), RT-CER (this year)

auto_kEFIT is Built on the Assumption that There is No User Input During the Workflow (David Eldon)



- **Offline Testing Started ➔ Online Development**
- **Pick your shot & timing and press GO!**
- **Aim: No Human Intervention requirement**

The developer interface with one GO button

1. RT-Kinetic EFIT

2. Real-time Stability Calculations

How can we achieve rt-Stability calculations (Alex Glasser)

- **Need for Control:**
  - Two time scales of crucial importance are
    - Energy confinement time, τ_E, ➔ pressure profile to equilibrate
    - Current relaxation time, τ_R, ➔ plasma current density profile to equilibrate
  - In DIII–D, τ_E ~200 ms & τ_R is ~2 s, in ITER both > seconds.
- **How to get it:**
  - The fastest Stability Calculations: Single core DCON 5 s for *n*=1 and 10 s for *n*=2.
  - Parallelizing DCON
    - Parallelize the coordinate transfer
    - Parallelize into subdomains (ODE)
  - Initial results show we can get to ~200 ms computation time

Stability Analysis: Using RT-EFIT + DCON

- Start with non-resistive DCON with the wall
- Solves the ideal MHD (low toroidal number)
- Using the Energy Principle

$$\delta W = \frac{1}{2}\int_{\Omega} d\mathbf{x}\,[Q^2 + \mathbf{J}\cdot\xi\times\mathbf{Q} + (\xi\cdot\nabla P)(\nabla\cdot\xi) + \gamma P(\nabla\cdot\xi)^2]$$

- The Ξ_ψ which minimizes the 'action' δW_P is seen to satisfy the Euler-Lagrange equation:

$$\left(\mathbf{F}\Xi'_\psi + \mathbf{K}\Xi_\psi\right)' - \left(\mathbf{K}^\dagger\Xi'_\psi + \mathbf{G}\Xi_\psi\right) = 0$$

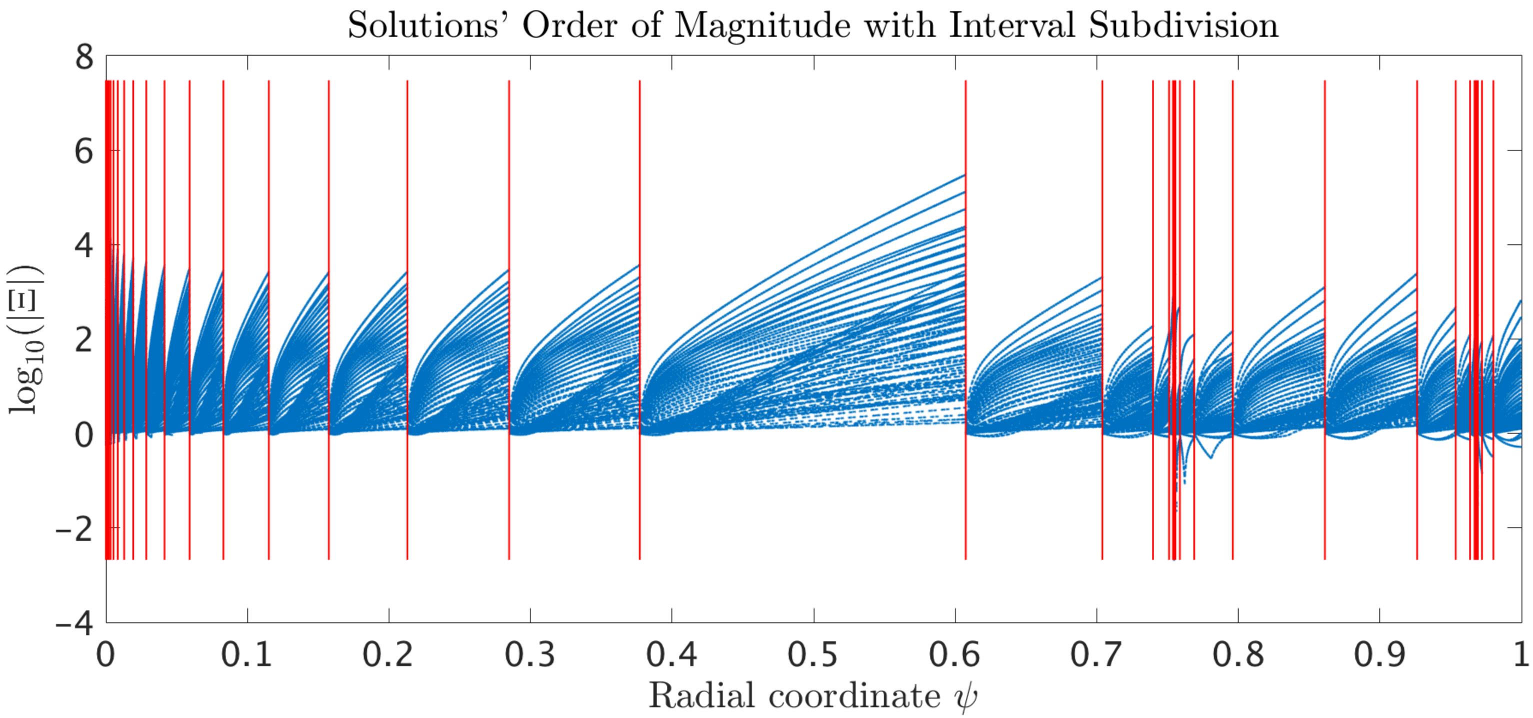
- Convert the problem to a 2-point BVP (with analyticity condition at singular points) because it allows fast computation methods

$$\begin{pmatrix}\dot{\mathbf{q}}\\ \dot{\mathbf{p}}\end{pmatrix} = \mathbf{A}(\psi)\begin{pmatrix}\mathbf{q}\\ \mathbf{p}\end{pmatrix} \text{ with } \mathbf{q}(0)=0,\ \mathbf{p}(1)=0, \text{ and } \mathbf{K}^\dagger\mathbf{F}^{-1}\mathbf{K}\mathbf{q}\Big|_{m_s} = 0$$

Parallelize integration for multiple cores with state transition matrix and domain decomposition

$$\dot{\mathbf{\Phi}}(\psi) = \mathbf{A}(\psi)\mathbf{\Phi}(\psi) \quad \text{with} \quad \mathbf{\Phi}(\psi_0) = \mathbb{1}$$

$$\begin{pmatrix} \mathbf{q}_1 \\ 0 \end{pmatrix} = \mathbf{\Phi}(\psi_n, \psi_{n-1}) \cdots \mathbf{\Phi}(\psi_2, \psi_1)\mathbf{\Phi}(\psi_1, \psi_0) \begin{pmatrix} 0 \\ \mathbf{p}_0 \end{pmatrix}$$



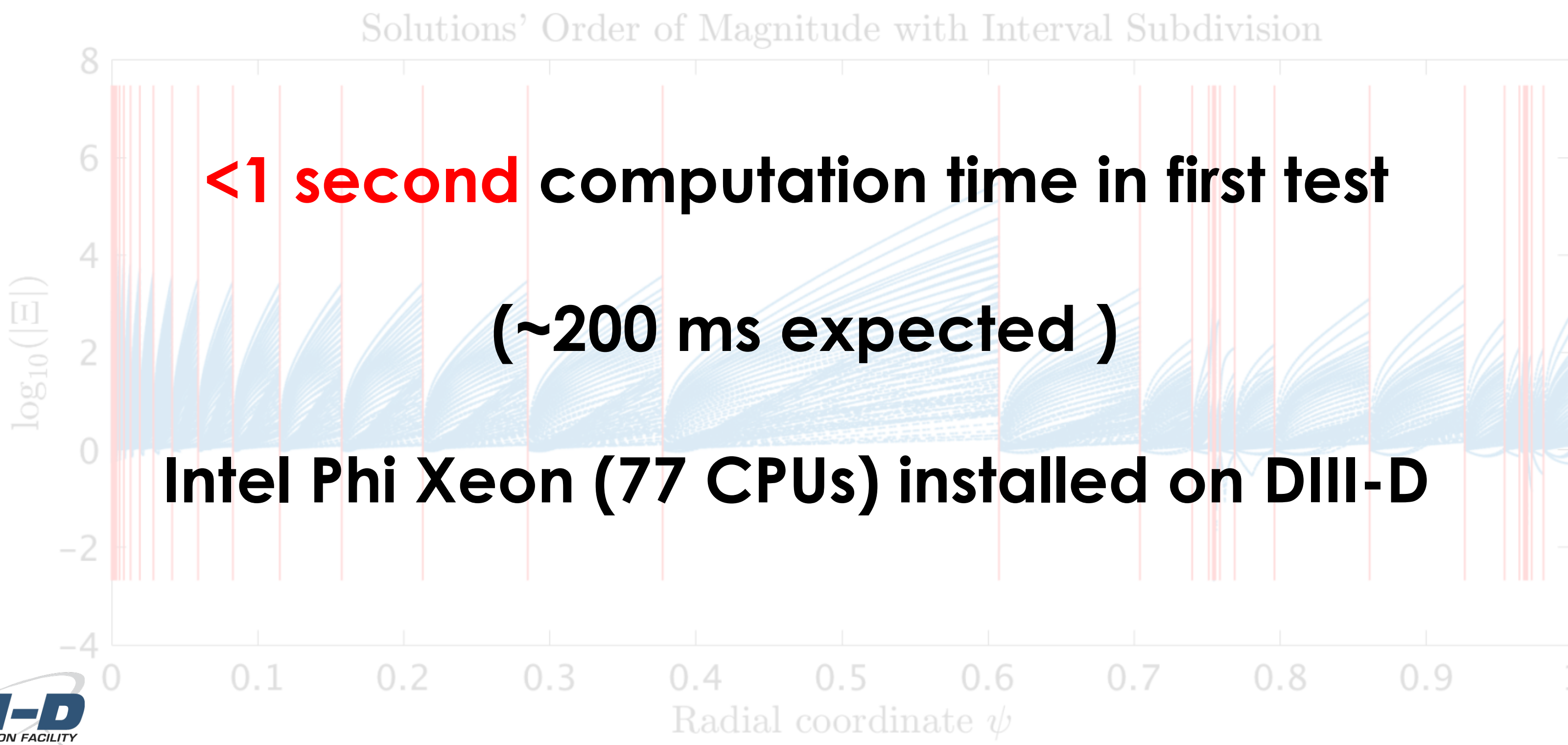
DIII-D NATIONAL FUSION FACILITY

1611-00446/21

Parallelize integration for multiple cores with state transition matrix and domain decomposition

$$\dot{\mathbf{\Phi}}(\psi) = \mathbf{A}(\psi)\mathbf{\Phi}(\psi) \quad \text{with} \quad \mathbf{\Phi}(\psi_0) = \mathbb{1}$$

$$\begin{pmatrix} \mathbf{q}_1 \\ 0 \end{pmatrix} = \mathbf{\Phi}(\psi_n, \psi_{n-1}) \cdots \mathbf{\Phi}(\psi_2, \psi_1)\mathbf{\Phi}(\psi_1, \psi_0) \begin{pmatrix} 0 \\ \mathbf{p}_0 \end{pmatrix}$$



<1 second computation time in first test

(~200 ms expected)

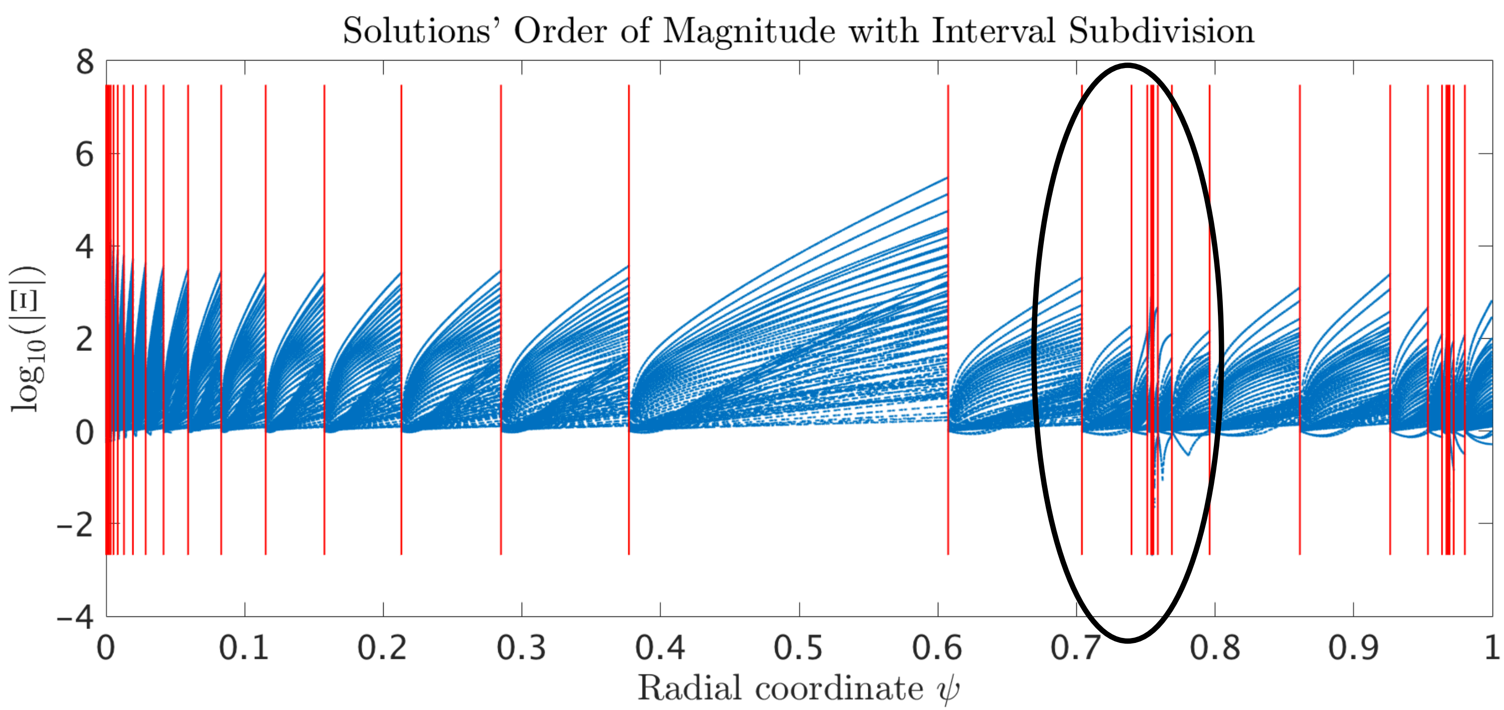
Intel Phi Xeon (77 CPUs) installed on DIII-D

DIII-D NATIONAL FUSION FACILITY

Parallelize integration for multiple cores with state transition matrix and domain decomposition

$$\dot{\boldsymbol{\Phi}}(\psi) = \mathbf{A}(\psi)\boldsymbol{\Phi}(\psi) \quad \text{with} \quad \boldsymbol{\Phi}(\psi_0) = \mathbb{1}$$

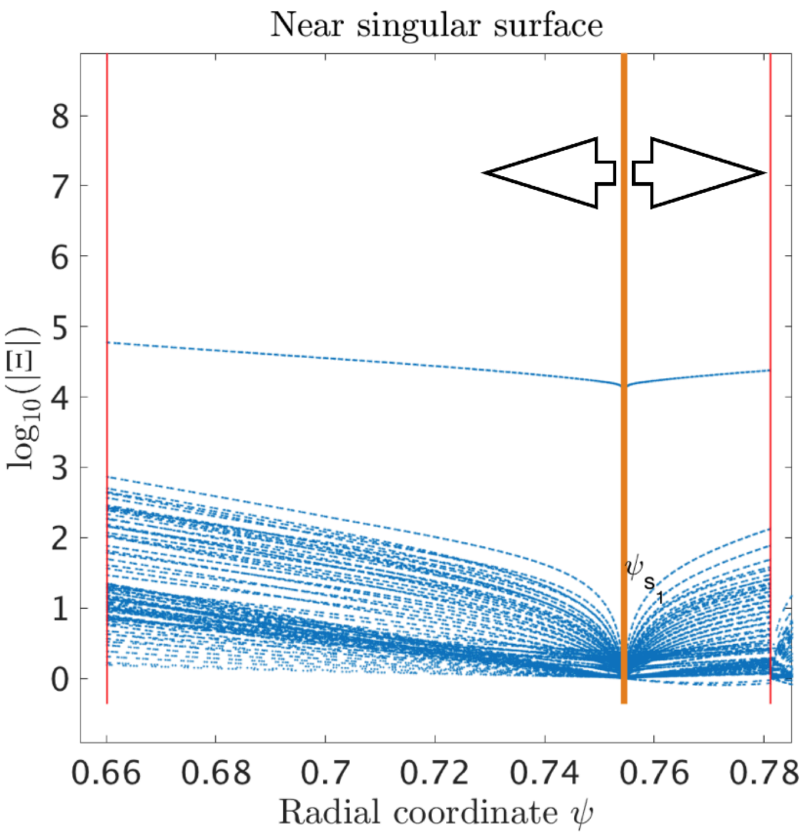
$$\begin{pmatrix} \mathbf{q}_1 \\ 0 \end{pmatrix} = \boldsymbol{\Phi}(\psi_n, \psi_{n-1}) \cdots \boldsymbol{\Phi}(\psi_2, \psi_1)\boldsymbol{\Phi}(\psi_1, \psi_0) \begin{pmatrix} 0 \\ \mathbf{p}_0 \end{pmatrix}$$



Integration at singular surface (1)

- **State transition matrices may be inverted to integrate out from singular surfaces rather than in, ensuring only analytic solutions are integrated**

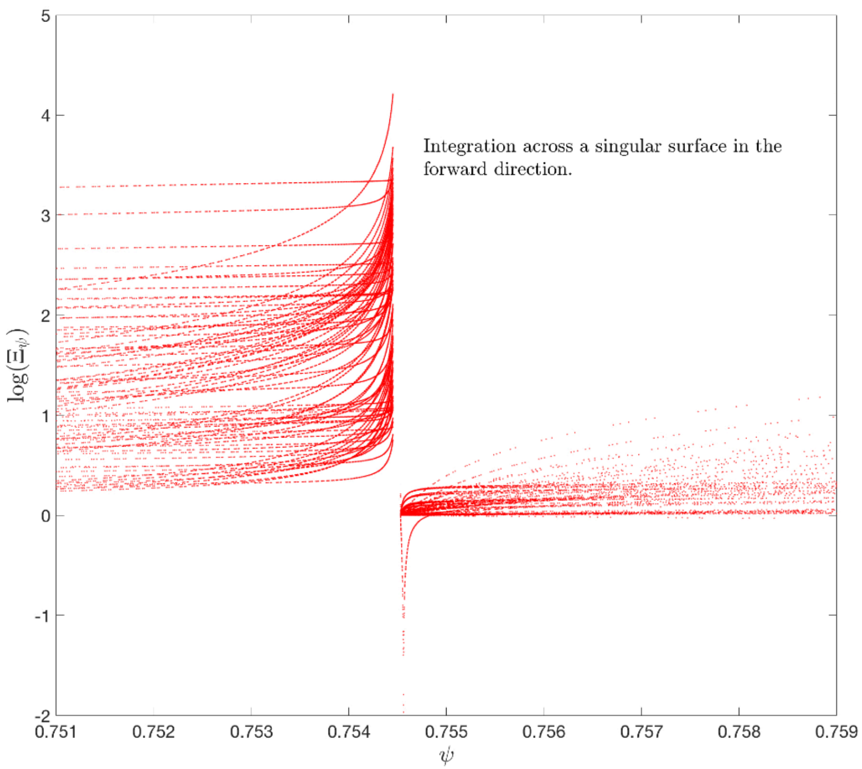
$$\mathbf{\Phi}^{-1}(\psi_1, \psi_2) = \mathbf{\Phi}(\psi_2, \psi_1)$$

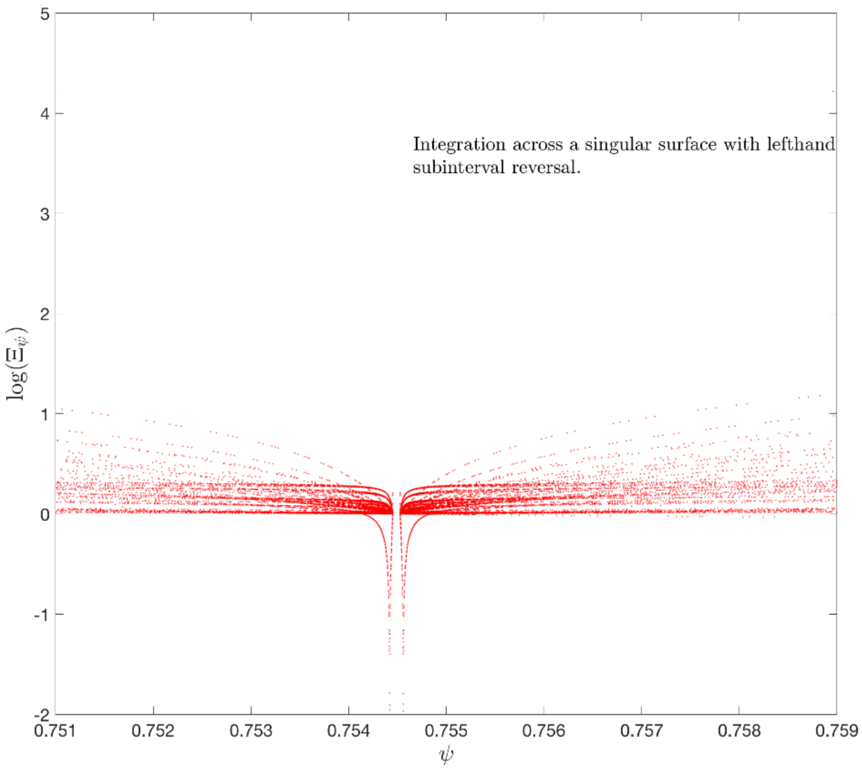


Integration at singular surface (2)

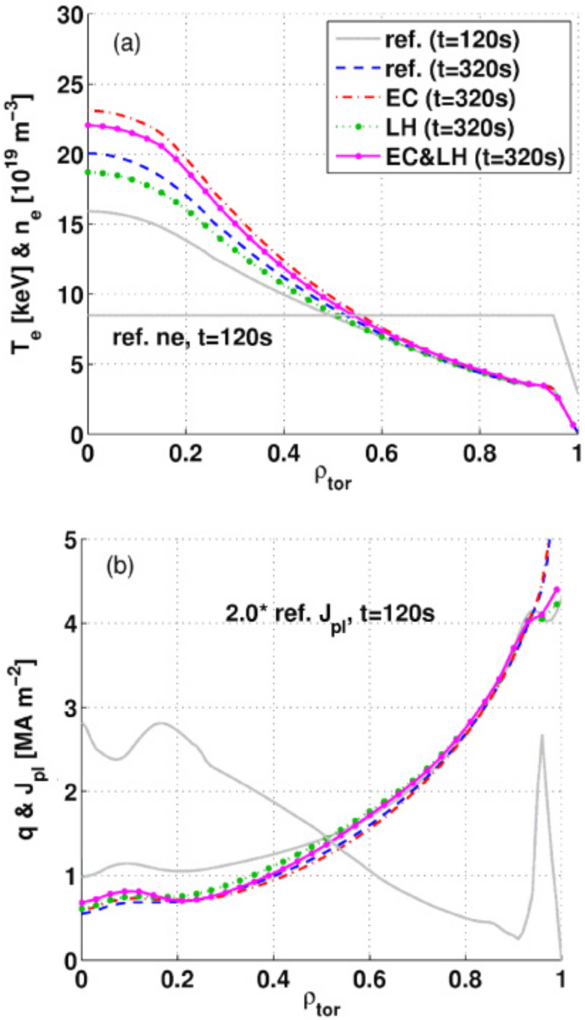
- **State transition matrices may be inverted to integrate <u>out</u> from singular surfaces rather than <u>in</u>, ensuring only analytic solutions are integrated**

$$\mathbf{\Phi}^{-1}(\psi_1, \psi_2) = \mathbf{\Phi}(\psi_2, \psi_1)$$





Vision: Real-time Stability Calculations for Plasma Control and Disruption Avoidance



ITER Profile Control Simulation (Kim and Lister NF 2012)

- **Develop the system for ITER on current machines**
- **Each server will run a different variation of a profile parameter (e.g. increase NBI power, reduce edge current etc.)**
- **Project the stability in the future.**
- **If approaching stability boundary change/control the profiles**
  - Calculate multiple profile variations ➔ choose the best path

Conclusions

- **Automated kinetic equilibrium reconstructions is implemented at DIII-D ➔ Real-time under development**
- **Real-time methods for ideal stability calculations based on DCON are under development at DIII-D**
- **Aim: Control profiles evolution to keep the plasma away form the stability boundaries**
  - **First to be tested at DIII-D**
  - **Then, implemented at ITER**