Center for Simulation of Wave Interactions with MHD (SWIM)

FEST seminar
January 21, 2010
ORNL

D. B. Batchelor, L. A. Berry, E. F. Jaeger, D. A. Spong – *ORNL Fusion Energy*

D. E. Bernholdt, E. D'Azevedo, W. Elwasif, V. Lynch (NCCS) – *ORNL Computer Science*

S. C. Jardin, E. Feibush, D. McCune, J. Chen, L. P Ku, M. Chance, J. Breslau – *PPPL*

G. Abla, M. Choi, D. P. Schissel – *General Atomics* , **R. W. Harvey** – *CompX*

R. Bramley, S. Foley – *Indiana University,* **D. Keyes** – *Columbia University,* **D. Schnack, T. Jenkins** – *U. Wisconsin*

P. T. Bonoli, J. Ramos, J.Wright – *MIT,* **S. Kruger** – *TechX*, **G. Bateman** – *Lehigh University*

Unfunded participants:

L. Sugiyama – *MIT,* **J. D. Callen, C. C. Hegna, C. Sovinec** – *University of Wisconsin,* **E.**

H. St. John – *General Atomics,* **A. Kritz** – *Lehigh Univ.*

See our fun website at:

www.cswim.org

The ISOFS subcommittee (2002) recommended that the US undertake a major initiative – Fusion Simulation Project (FSP)

"The purpose is to make a significant advance toward the ultimate objective of fusion simulation – to predict in detail the behavior of any discharge in a toroidal magnetic fusion device on all important time and space scales."

What do we want our simulation to do?

Basic theory requires simulation – i.e. large scale computation

- **Needed to find the consequences of any theory in a *real* situation**
- **Needed to validate (or *invalidate*) a theory by experimental comparison**

Simulation directly supports experiments

- **Auxiliary system design**
- **Plasma scenario development**
- **Experimental (shot) design**
- **Experiment interpretation**

Validated simulation is essential to project to the next level of fusion device (DEMO)

What do we want our simulation to do?

Basic theory requires simulation – i.e. large scale computation

- **Needed to find the consequences of any theory in a *real* situation**
- **Needed to validate (or *invalidate*) a theory by experimental comparison**

Simulation directly supports experiments

- **Auxiliary system design**
- **Plasma scenario development**
- **Experimental (shot) design**
- **Experiment interpretation**

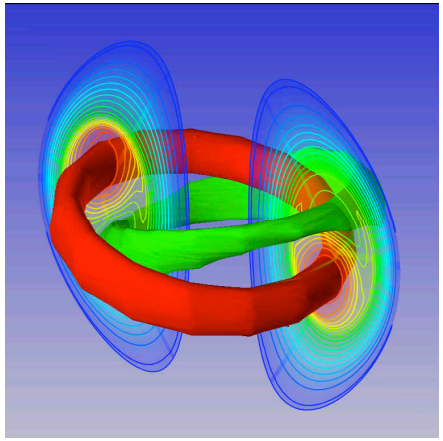
Validated simulation is essential to project to the next level of fusion device (DEMO)

Validated simulation, which may still contain elements of empiricism, embodies and demonstrates our understanding of the science and is the only way to reliably extrapolate to new devices

There are great many processes simultaneously at work in fusion plasmas → at the same time and in the same volume

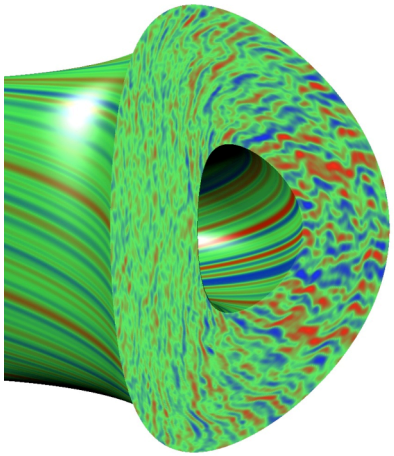
Hydromagnetic force balance of plasma pressure supported by J×B force

- MHD equilbrium
- Macroscopic fluid instability
- Current and magnetic field evolution



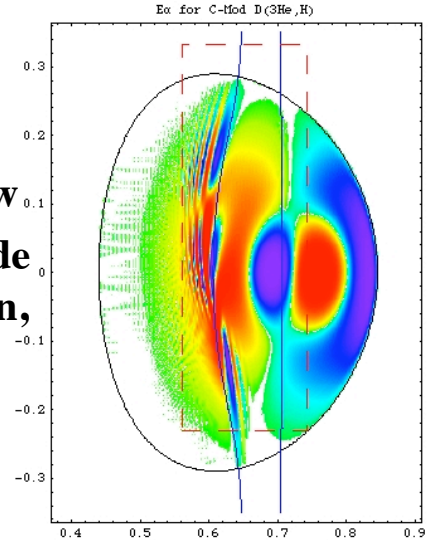
Kinetic stability and transport

- Micro-stability
- Turbulence and turbulent transport
- Long mean-free-path collisional transport
- Fusion heating



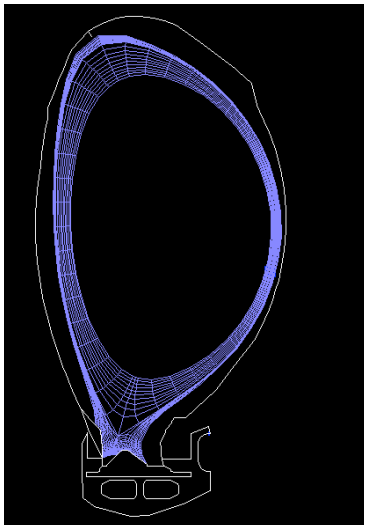
Injection of high-power waves or particle beams, magnetic flux

- Plasma heating
- Externally driven current or plasma flow
- Wave processes – mode conversion, absorption, reflection
- Non-Maxwellian particle distributions



Plasma/edge interactions

- Atomic physics processes
- Transition closed → open flux surfaces
- Transport on open field lines
- Turbulence
- Plasma/material interactions



All the plasma physics we need is determined by the kinetic equation + Maxwell's equations

$$\frac{\partial f_j}{\partial t} + \mathrm{v}\cdot\nabla f_j + \tfrac{q}{m}\left[E + \mathrm{v}\times B\right]\cdot\nabla_{\mathrm{v}} f_j = C(f_j) + S(\mathbf{x},\mathbf{v})$$

Particle sources

convection in space

convection in velocity space

Collisional relaxation toward Maxwellian in velocity space

$$q(\mathbf{x},t) = \sum_j q_j n_j(\mathbf{x},t) \equiv \sum_j q_j \int d^3\mathrm{v} f_j(\mathbf{x},\mathbf{v},t)$$

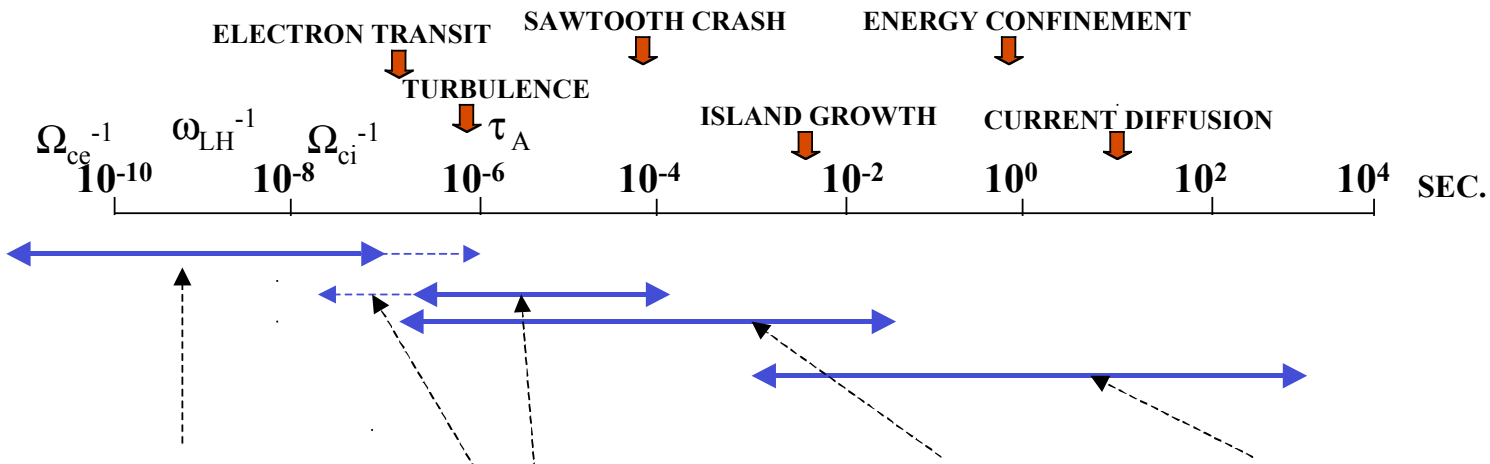
$$\mathbf{J}(\mathbf{x},t) = \sum_j q_j n_j(\mathbf{x},t)\mathbf{V}_j(\mathbf{x},t) \equiv \sum_j q_j \int d^3\mathrm{v}\,\mathbf{v} f_j(\mathbf{x},\mathbf{v},t)$$

Sources to Maxwell's equations for E, B

- **High dimensionality – f(x, v, t) is 7D, sensitive to geometric details**
- **Extreme range of time scales – wall equilibration/electron cyclotron O(10^{14})**
- **Extreme range of spatial scales – machine radius/electron gyroradius O(10^{4})**
- **Extreme anisotropy – mean free path in magnetic field parallel/perp O(10^{8})**
- **Non-linearity**

How do we get something computable out of this?

We make progress by taking advantage of time-scale separations and by averaging over dimensions



Single frequency and prescribed plasma background	Neglect displacement current, average over gyroangle, (some) with electrons	Neglect displacement current, integrate over velocity space, neglect electron inertia	Neglect displacement current, integrate over velocity space, average over surfaces, neglect ion & electron inertia
RF Codes	Gyrokinetics Codes	Extended MHD Codes	Transport Codes
wave-heating and current-drive	turbulent transport	device scale stability	discharge time-scale

The goal of integrated simulation is to bring together all of these essentially separate disciplines

Approach – start simple, pair-wise physics couplings → "Prototype FSP projects"

- **Address one or a few of the fundamental issues for physics integration** (wide range of time scales, range of space scales, extreme anisotropy, non-linearity, models of different dimensionality)

- **Address problems of immediate programmatic importance**

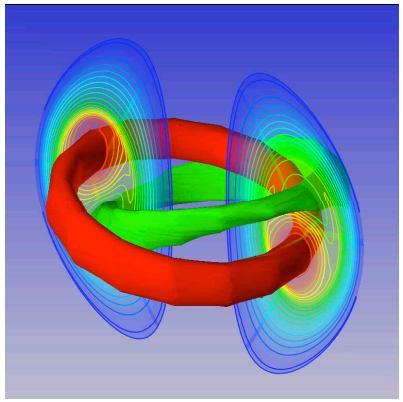
- **Test different approaches to computational framework/code integration that might ultimately lead to a basis for complete simulation**

SWIM is one three such prototype FSP projects sponsored jointly by DOE Fusion and CS/Math under SciDAC umbrella

SWIM brings together two mature sub-disciplines of fusion plasma physics, each with a demonstrated code base

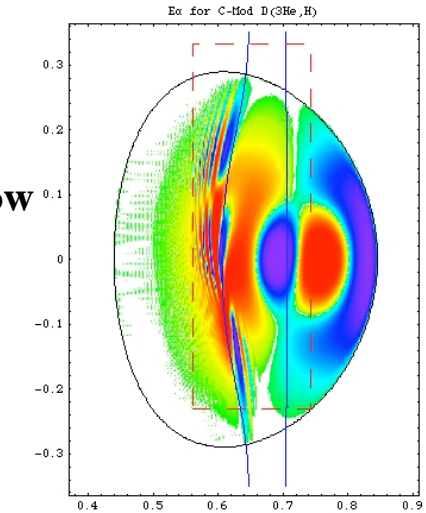
Extended MHD – CEMM

- MHD equilibrium
- Macroscopic fluid instability
- Current and magnetic field evolution



High power wave-plasma interactions – CSWPI

- Plasma heating
- Externally driven current or plasma flow
- Non-Maxwellian particle distributions



Fluid equations, extended to include non-ideal and kinetic effects (10^{-5} sec < τ_{MHD} < 10^{-1} sec)

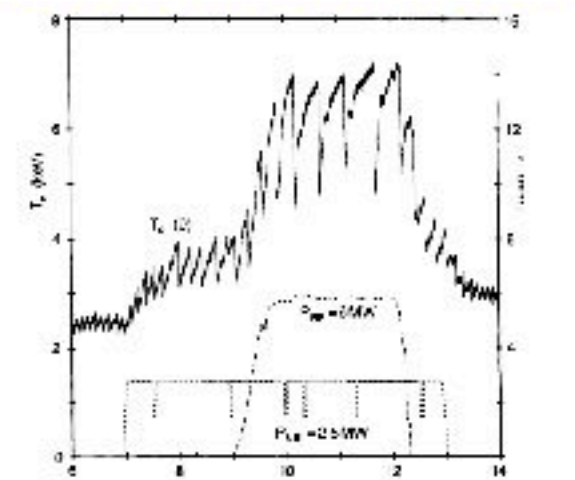
Plasma wave equation (τ_{RF} < 10^{-7} sec), coupled to slow evolution of plasma velocity distribution (τ_{FP} > 10^{-2} sec)

Why couple these particular two disciplines?

- Macroscopic instabilities can limit plasma performance
- RF waves can mitigate and control instabilities

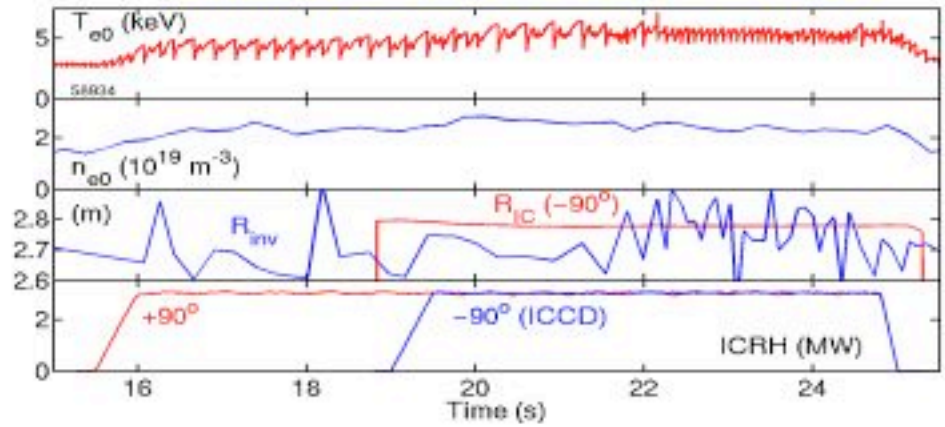
There are several experimentally demonstrated mechanisms by which RF waves can control the "sawtooth" instability

ICRF stabilization on JET



- ICRF heating can produce "monster" sawteeth – period and amplitude increased
- Likely stabilization mechanism – energetic particle production by RF

Sawtooth control with Minority Current Drive on JET

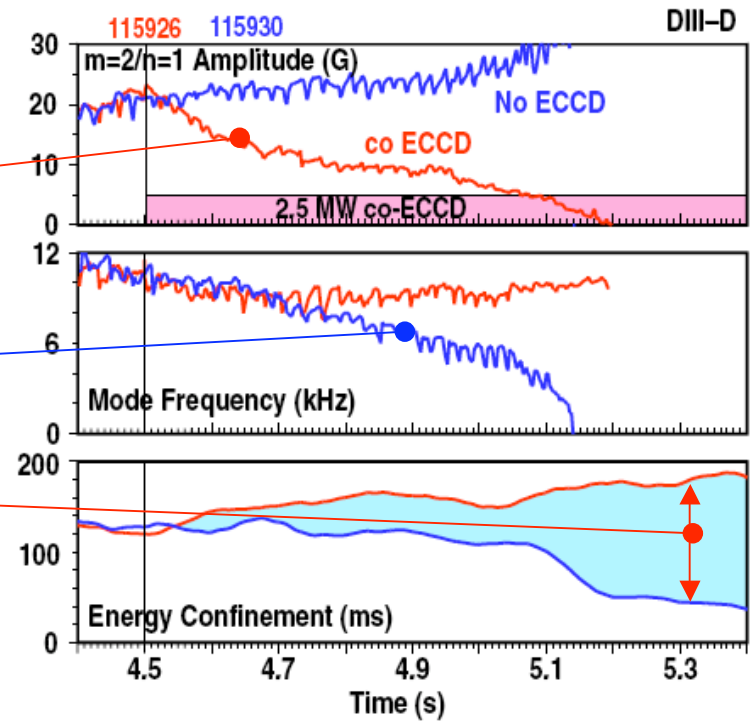


- ICRF minority current drive can either increase or decrease period and amplitude
- Likely stabilization/destabilization mechanism – RF modification of current profile

- Sawteeth can limit plasma performance themselves, or can trigger other instabilities – disruptions, neoclassical tearing modes
- Many physics processes interact – *qualitative* understanding exists but *quantitative* verification and *prediction* is lacking

Suppression of "neoclassical tearing modes" (NTM) by RF leads to improvement in confinement

- Electron cyclotron current drive drives down mode amplitude
- keeps mode rotating (no drop in frequency)
- improves energy confinement



R. Prater
APS 2003

- Empirical scaling of NTM pressure limits in ITER leave little margin in performance
- "Understanding the physics of neoclassical island modes and finding means for their avoidance or for limiting their impact on plasma performance are therefore important issues for reactor tokamks and ITER" – ITER Physics Basis (1999)

The SWIM project is carried out in two physics campaigns distinguished by the time scale of unstable MHD motion

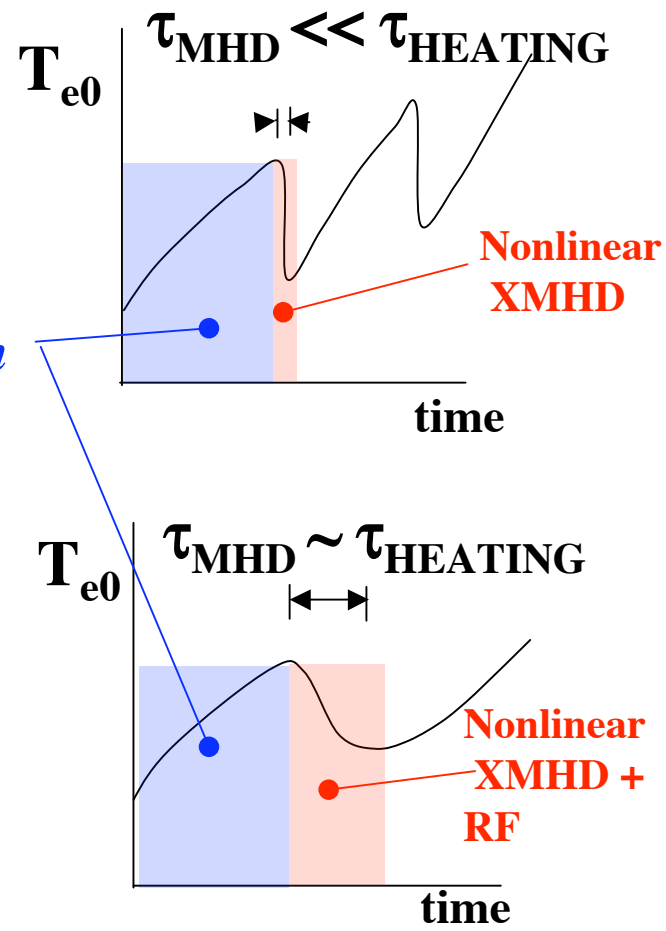
Fast MHD phenomena – separation of time scales

- Response of plasma to RF much slower than fast MHD motion – transport time-scale
- RF drives slow plasma evolution, sets initial conditions for fast MHD event
- Example: sawtooth crash

Slow plasma evolution

Slow MHD phenomena – no separation of time scales

- RF affects dynamics of MHD events ⇔ MHD modifications affect RF drive plasma evolution
- Deals with multi-scale issue of parallel kinetic closure including RF (mainly ECRH)
- Example: Neoclassical Tearing Mode

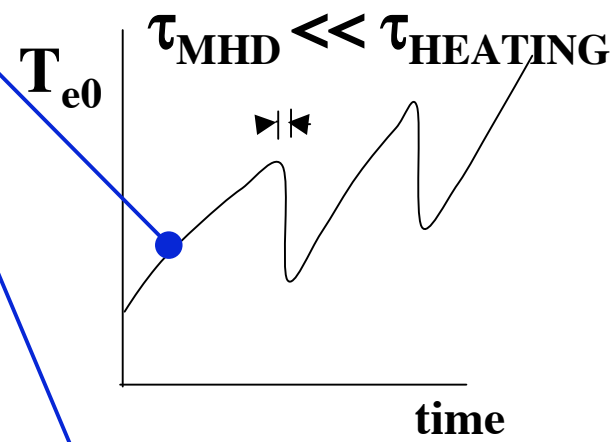


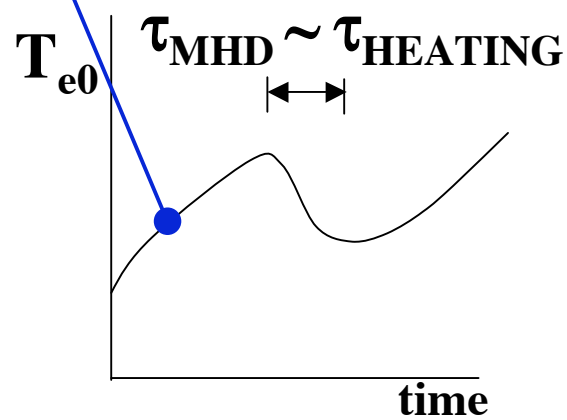
Software infrastructure: Integrated Plasma Simulator (IPS)

A flexible, extensible computational framework capable of coupling state-of-the-art models for energy and particle sources, transport, and stability for tokamak core plasma

Simulation of plasma evolution requires complete model – Integrated Plasma Simulator (IPS)

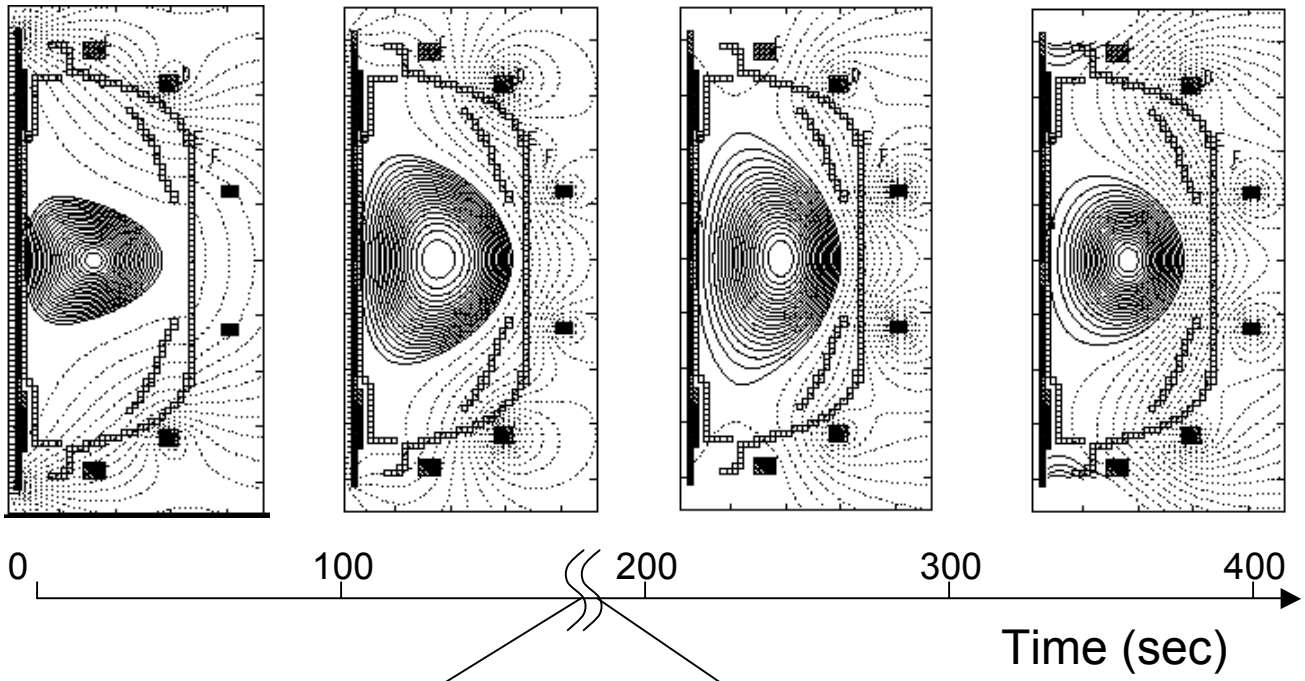
- Heating and current drive sources
- Particle sources
- Transport
- Magnetic field evolution
- Evolution of non-thermal distributions
- Monitoring of linear MHD stability



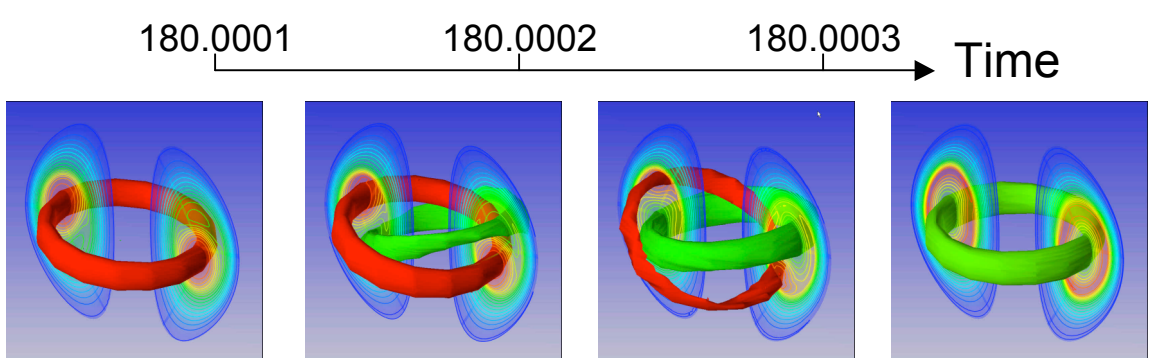


Integrated Plasma Simulator will allow coupling of virtually any fusion fusion code, not just RF and MHD, and should provide the framework for a full fusion simulation

Simulation of plasma evolution requires complete model – Integrated Plasma Simulator (IPS)



- Plasma evolves through a series of 2D axisymmetric equilibrium states
  - Heating and current drive sources
  - Particle sources
  - Transport
  - Magnetic field evolution
- Instabilities occur as instantaneous events



- 3D Extended MHD simulation starts and ends in axisymmetric state

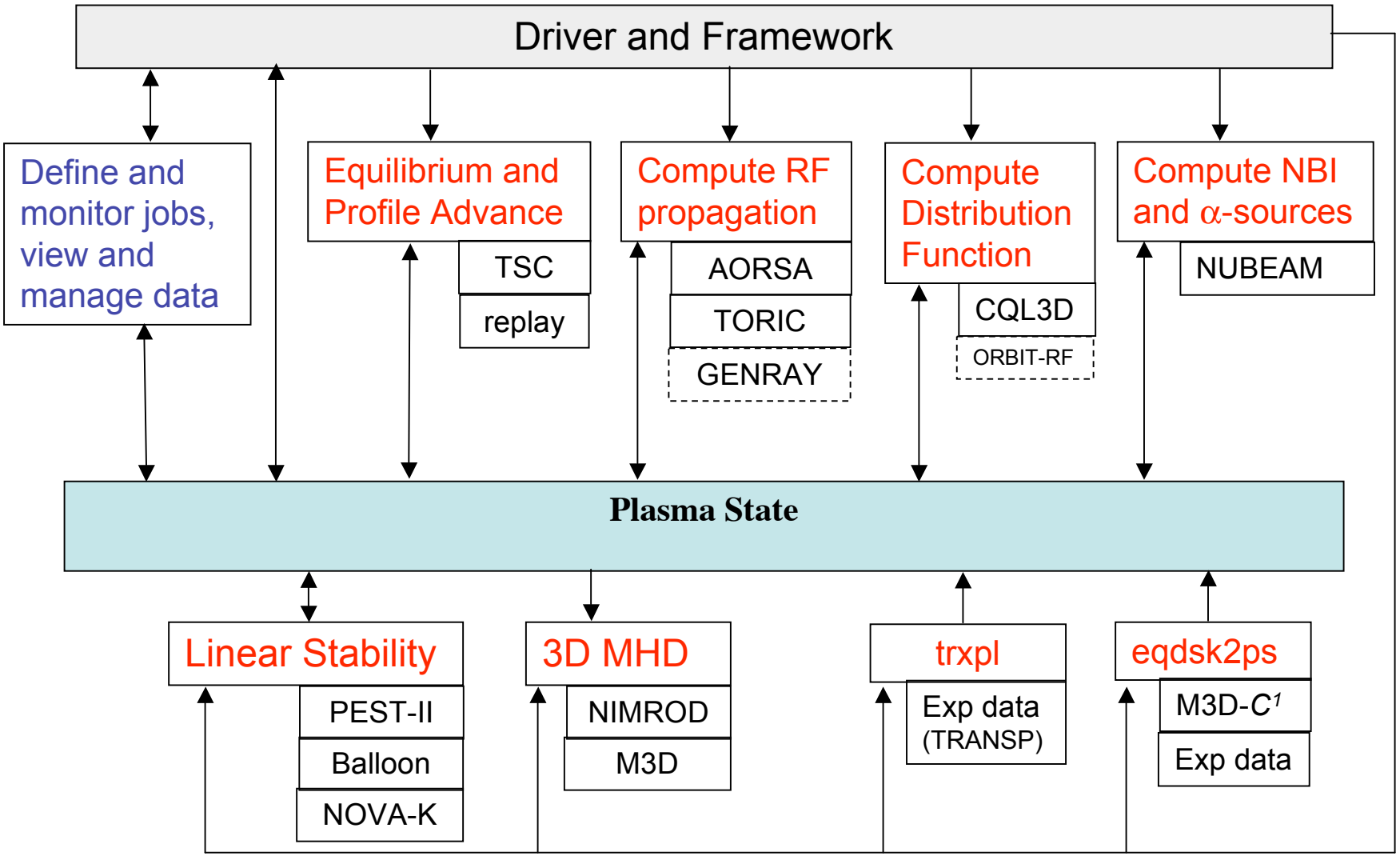
Project goals

- **Develop IPS so that it provides a computational environment satisfying the SWIM project's needs for concurrency, performance, and data management and is the tool of choice for those performing tokamak simulations**
- **Demonstrate capability of the SWIM system to address important questions of sawtooth instability behavior and their control by RF (Fast MHD campaign)**
- **Implement the coupling of ECCD, non-linear MHD and kinetic closure for study of RF stabilization of Neoclassical Tearing Modes (NTM) (Slow MHD campaign) and perform numerical simulations comparing with NTM experiments**
- **Provide a base of experience with framework/component architecture applied to integrated fusion simulation that can be factored into the design of a larger scale Fusion Simulation Project.**

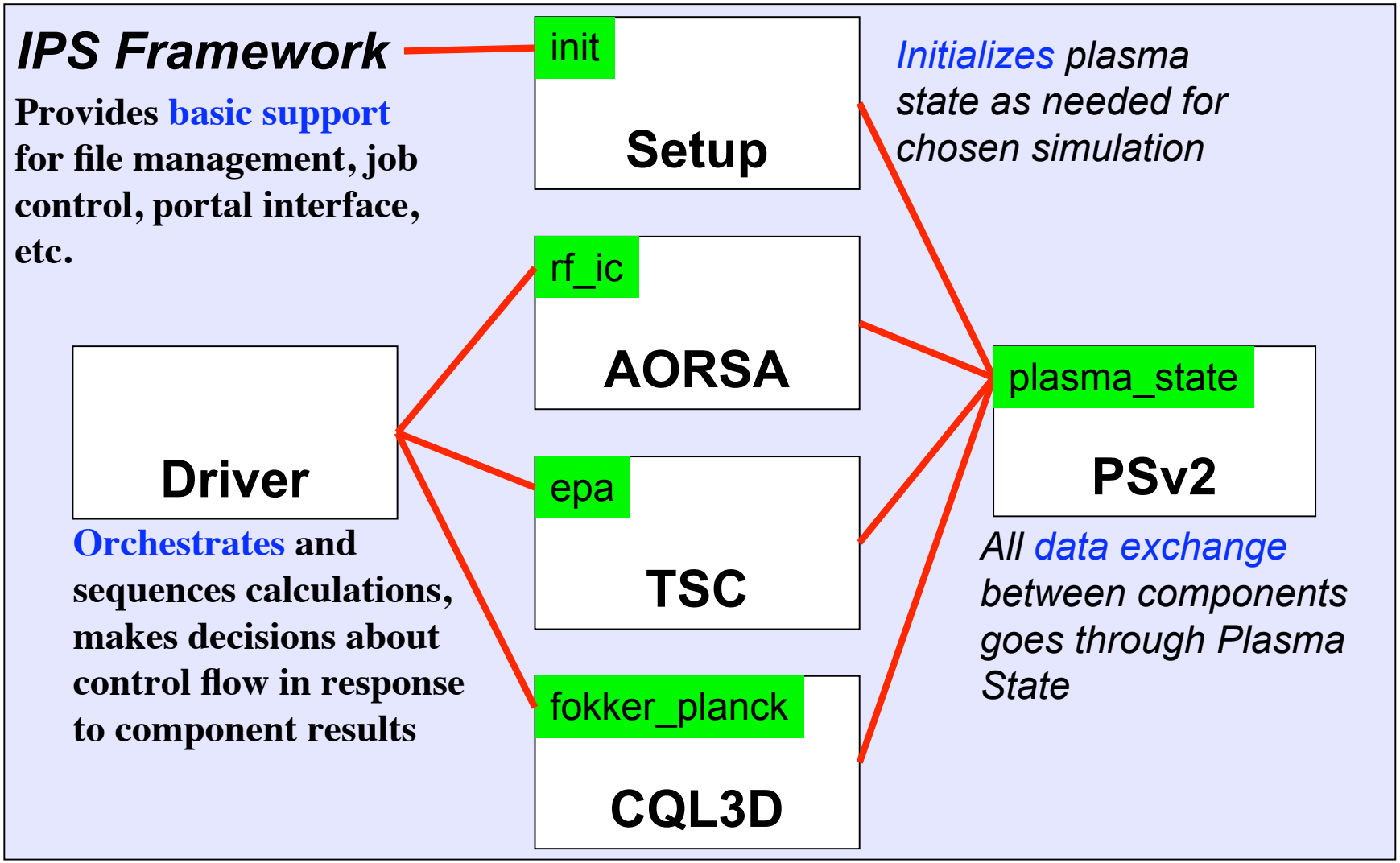
IPS Design Approach

- **Framework/component architecture – *written in Python***
  - **Effective division of labor – multiple groups can work independently, if they respect the interface**
  - **Flexibility – multiple codes can implement components interchangeably**
  - **Extensibility – easy to add components to framework**
- **Components implemented using existing whole codes *(usually in Fortran)* wrapped in standard component interface *(written in Python)***
  - **Rapid deployment – minimize changes to physics codes to adapt**
  - **Avoid bifurcation of physics modules – not different SWIM/stand-alone versions**
  - **Very simple component interface – init(), step(), finalize()**
- **File-based communication (→ in-memory data exchange now also)**
  - **Zero change to physics code – use existing I/O file structures**
  - **Avoid name-space/compiler/library incompatibilities between components**
- **Plasma State: official transfer mechanism for time-evolving data that must be transferred between components**
- **Target MPP computers from the start**

A physicists view of the Integrated Plasma Simulator
Implemented with existing well tested and validated codes

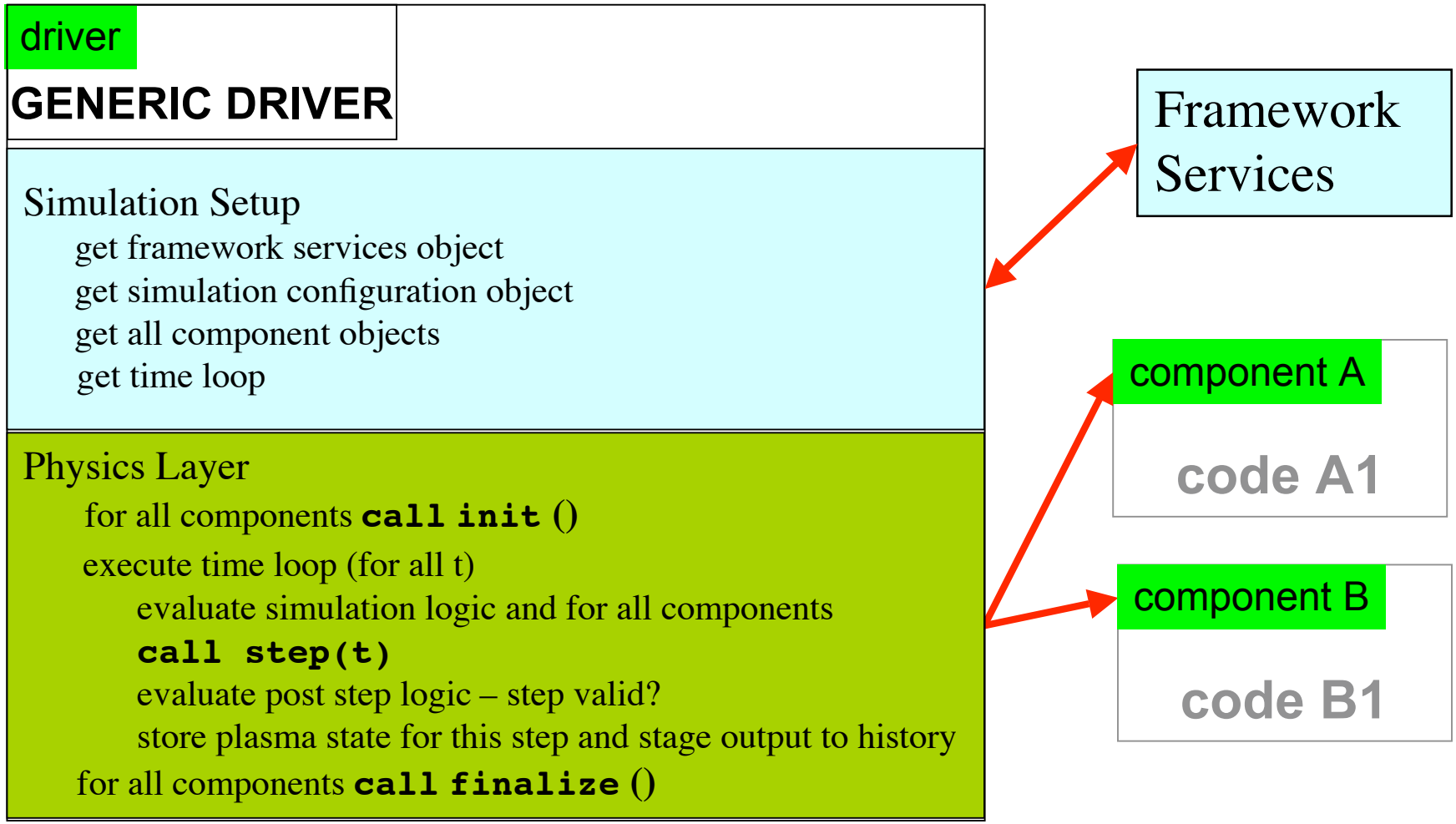


A Computer Scientists view – The IPS is a system for composing fusion simulation applications → Schematic of an IPS Application

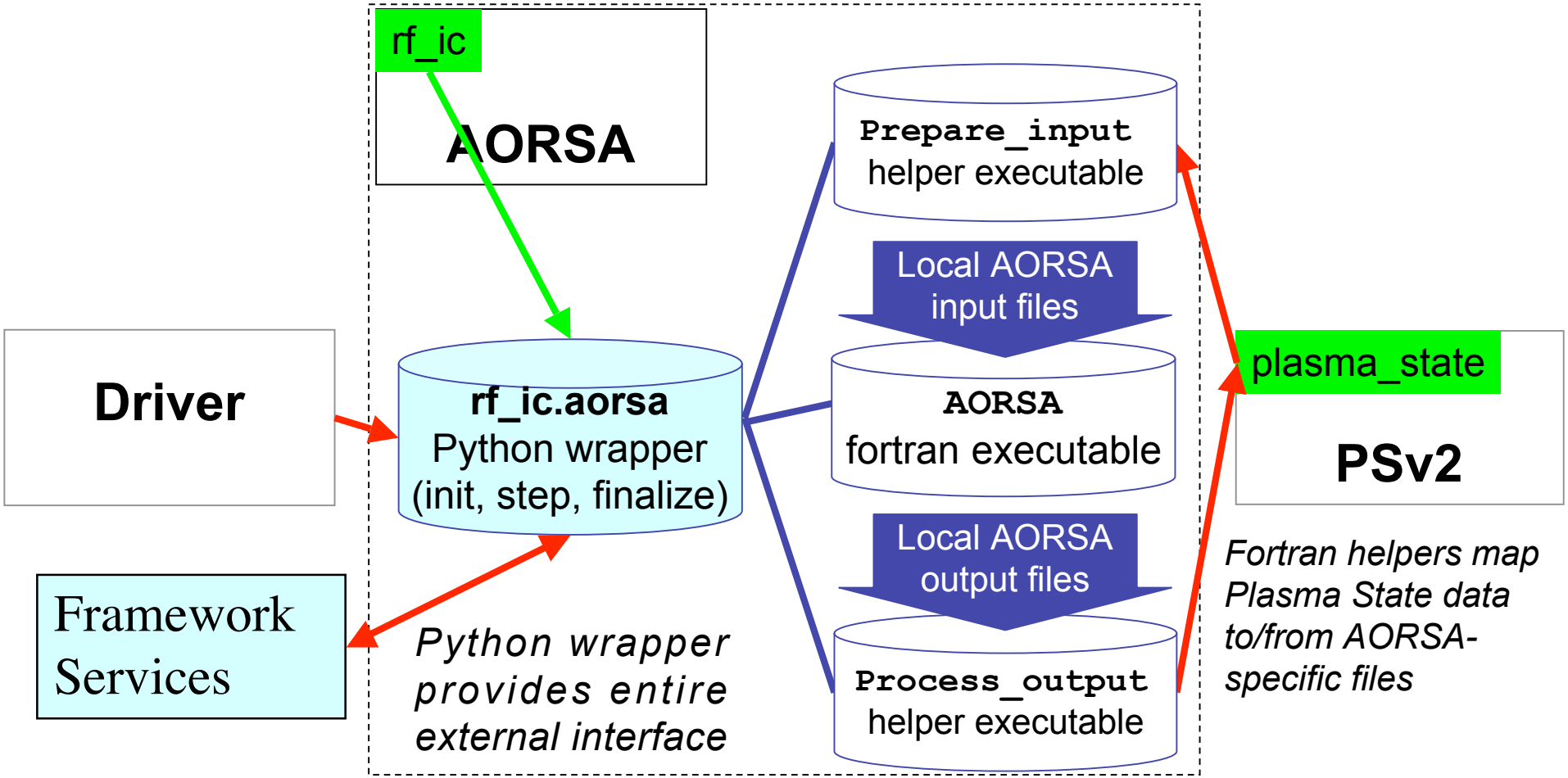


Components implement (one or more) specific interfaces.
A given interface may have multiple implementations.

Schematic of typical Driver component

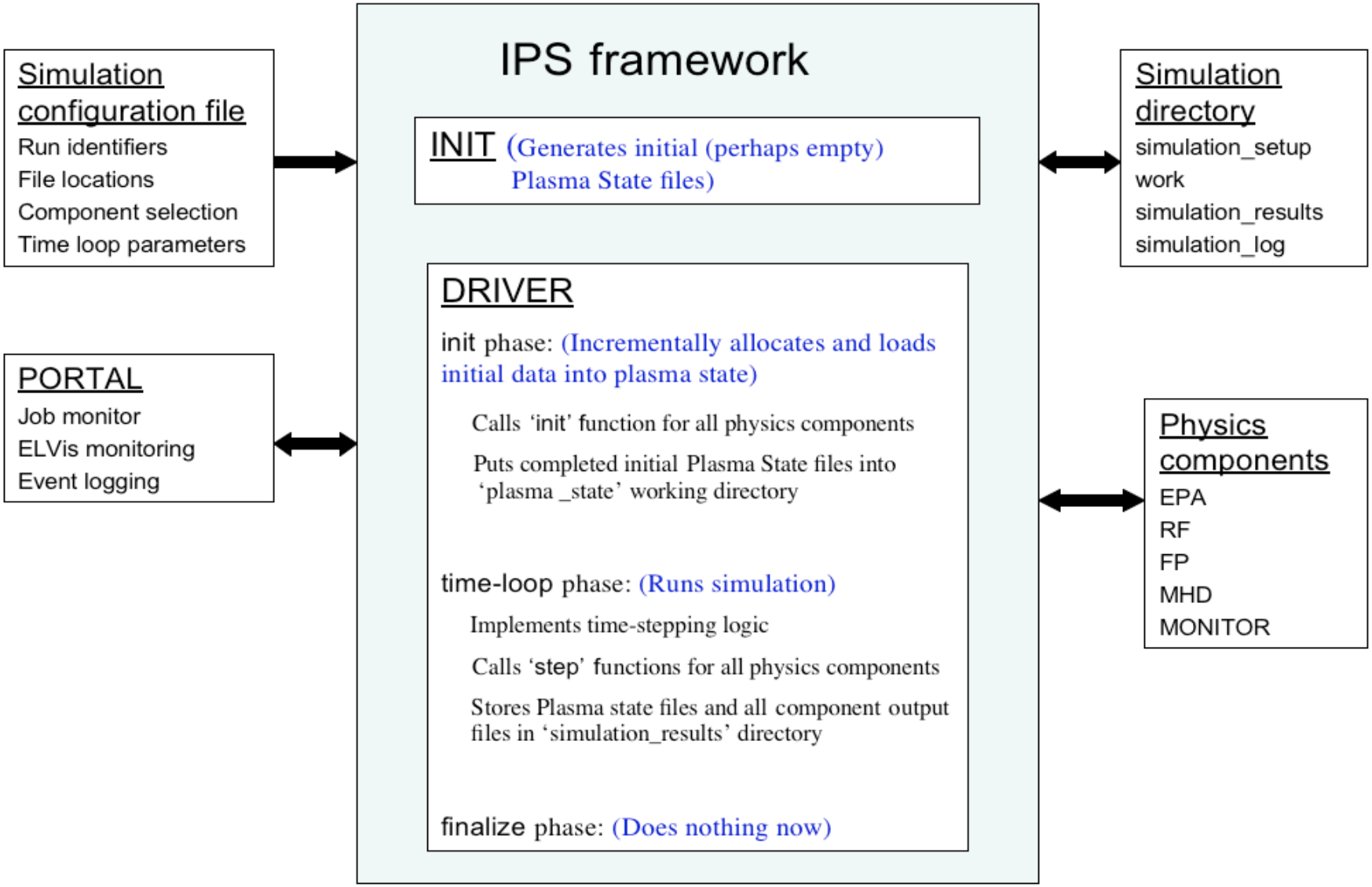


Schematic typical component internals – rf_ic implemented by AORSA



IPS design/specifications say nothing about internal implementation of components.

What does a driver do?



All Simulation data exchanged between components goes through Plasma State – but components can produce other files

In a narrow sense "Plasma State" is a flat data structure

- **Fortran 90 Module – now at version 2.023 (also C++ interface)**
  - **Distributed by NTCC, used in TRANSP, PTRANSP, FACETS …**
  - **netCDF for backend storage**
- **Supports multiple state instances (very important!) → e.g. current/prior state, pre-/post-sawtooth, etc**
- **PS data conventions (names, units, etc.) for IPS determined by (benevolent) dictator → extended as needed**
- **Data stored "as produced" → Consumer is responsible for adapting units/grids as needed**
- **Code is automatically generated from state specification text file → ease and accuracy of update**
- **Some types of data we don't know how to deal with yet → distribution functions are just code dependent filenames**

In a broader sense "Plasma State" of a simulation is a collection of files that are moved as needed by the framework

- **Current Plasma State file**
- **Previous Plasma State file**
- **Next Plasma State file**
- **Current eqdsk file**
- **Current dql file (quasilinear operator file used by AORSA and CQL3D)**
- **Current cql file (distribution function file used by AORSA and CQL3D)**
- **Current jsdsk file (used by linear MHD component)**

What does the Plasma State software consist of?

A data structure that contains data shared by components

- **Time evolving data – scalars, 1D profiles, 2D profiles, file paths to more complicated data (e.g. eqdsk file)**
  - **Species densities and temperatures, magnetic field, source profiles like heat deposition or driven current**
  - **A plasma state instance contains one set of data (time slice) not a time history. A time history consists of many plasma states**
  - **Data is grouped by component and all data is "owned" by a specific component**
  - **During a time step each component updates its own data as it finishes – data is only 'consistent' at the beginning and end of a valid time step**
- **Static data that is common between components – tokamak geometry, definition of auxiliary systems such as RF and neutral beam characteristics …**

A collection of routines defining the user interface

- **Functions to store and access state data → get, store, update, commit**
- **Functions to initialize plasma state structure and to manipulate data → allocate state arrays, copy plasma state objects, grid interpolation**

The basic interface is simple – simple things are easy

Initializing some plasma state arrays and storing them

```
! Get plasma state definition
    USE plasma_state_mod

! Get the current plasma state
    call ps_get_plasma_state(ierr, trim(cur_state_file))

! Set dimensions of some state arrays and allocate them
        ps%nspec_th = nspec_th
        ps%nrho = nrho

        CALL    ps_alloc_plasma_state(ierr)

! Set values for thermal species temperature profiles
            DO i = 1, nspec_th
                CALL profgen( ti0, ti_edge, alpha_Ti, zone_center, ps%Ts(:,i) )
            END DO

! write plasma state data file
        CALL    ps_store_plasma_state(ierr, cur_state_file)
```

This gets you all plasma state data declarations and access to all functions

This gets you all plasma state data

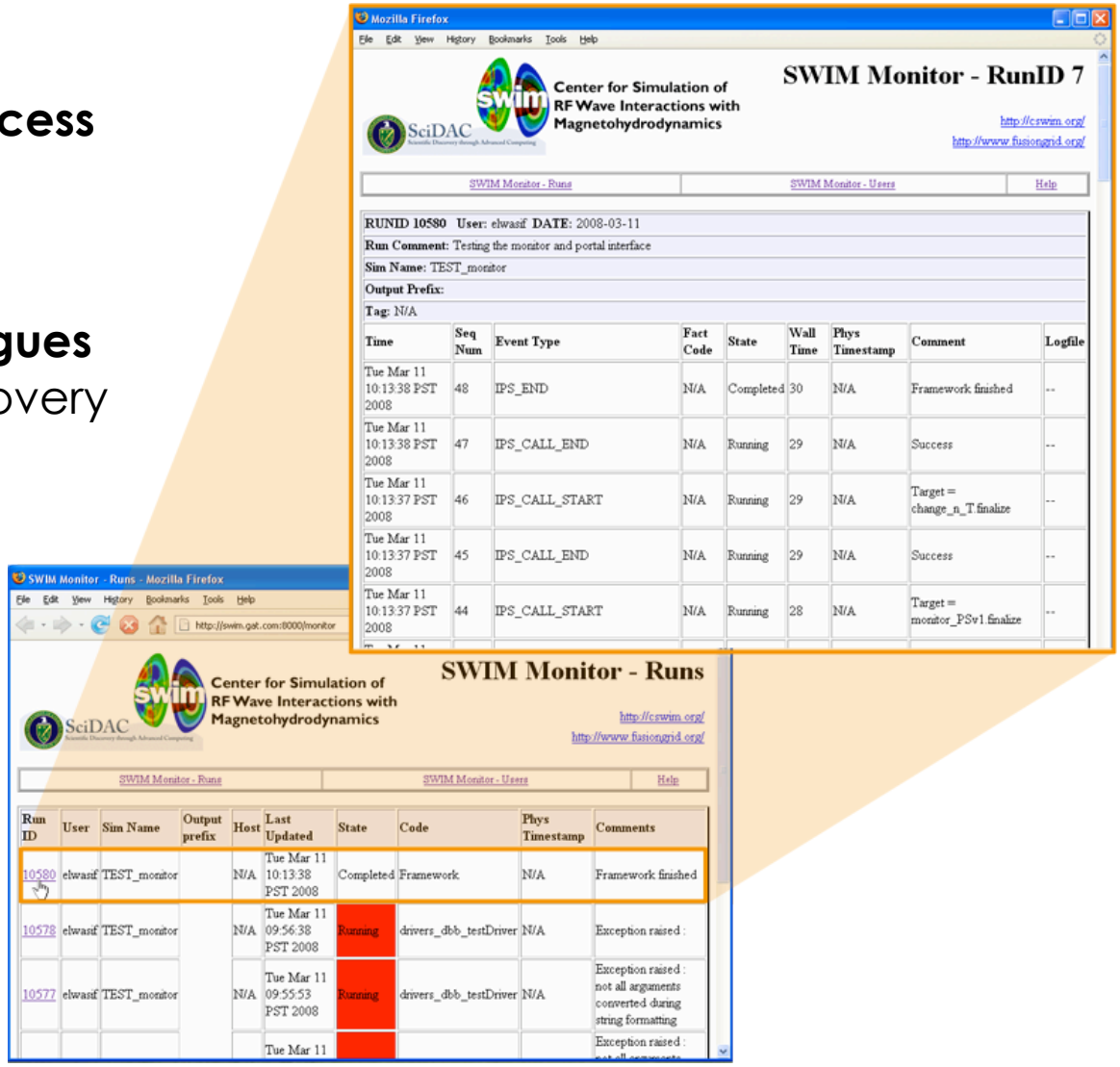
State arrays are allocated incrementally. This allocates arrays with newly assigned non-zero dimensions

This stores a complete Plasma State in a netcdf file

But there are dozens of routines to do more specialized things

SWIM Monitoring Portal – Real-time status

- **Web portal, worldwide access**
  - Status of all runs
  - Details on specific runs
- **Search meta data catalogues**
  - Rapid search and discovery



Q. What do you get out of an IPS run?
A. The output directory + monitor file in W3_dir

/Simulation_output (example `Run_1_CMOD_8021`)

- /work – current working files for this time step
  - /plasma_state – all current plasma state files
  - /epa
    - /tsc – all tsc input files, plasma state files → all tsc output files
  - /rf
    - /aorsa – all aorsa input files, plasma state files → all aorsa output files
  - /… other physics components
- /simulation_results/history – archived data from all time steps
  - /plasma_state – plasma state files for all time stamps
  - /<time 0>/components
    - /epa
      - /tsc – all tsc output files
    - /rf
      - /aorsa – all aorsa output files
  - /<time 1>/components
  - /… other time steps
- sim.config file
- /simulation_setup – all input files at simulation initialization
- /simulation_log – events published to the portal by the framework

- The framework builds the name, path and structure for the output directory from information in the config file – so you can change it if you want

Physics studies with IPS

- **ITER discharge simulations with massively parallel RF and neutral beam components (ORNL)**
- **Studies of RF driven energetic tail formation on Alcator C-mod (ORNL)**
- **Onset of saturated n = 1, m = 1,2 modes in NSTX – coupling of IPS to M3D (PPPL)**
- **Use of IPS to study control of sawtooth onset time with lower hybrid waves on C-mod (MIT/CompX)**
- **Use of IPS to study ECCD resistive tearing mode stabilization and motion of flux surfaces – coupling to GENRAY ECH ray tracing to NIMROD nonlinear MHD (U. Wisc./TechX)**

IPS is supporting ITER simulations for International Tokamak Physics Activity (ITPA) tasks

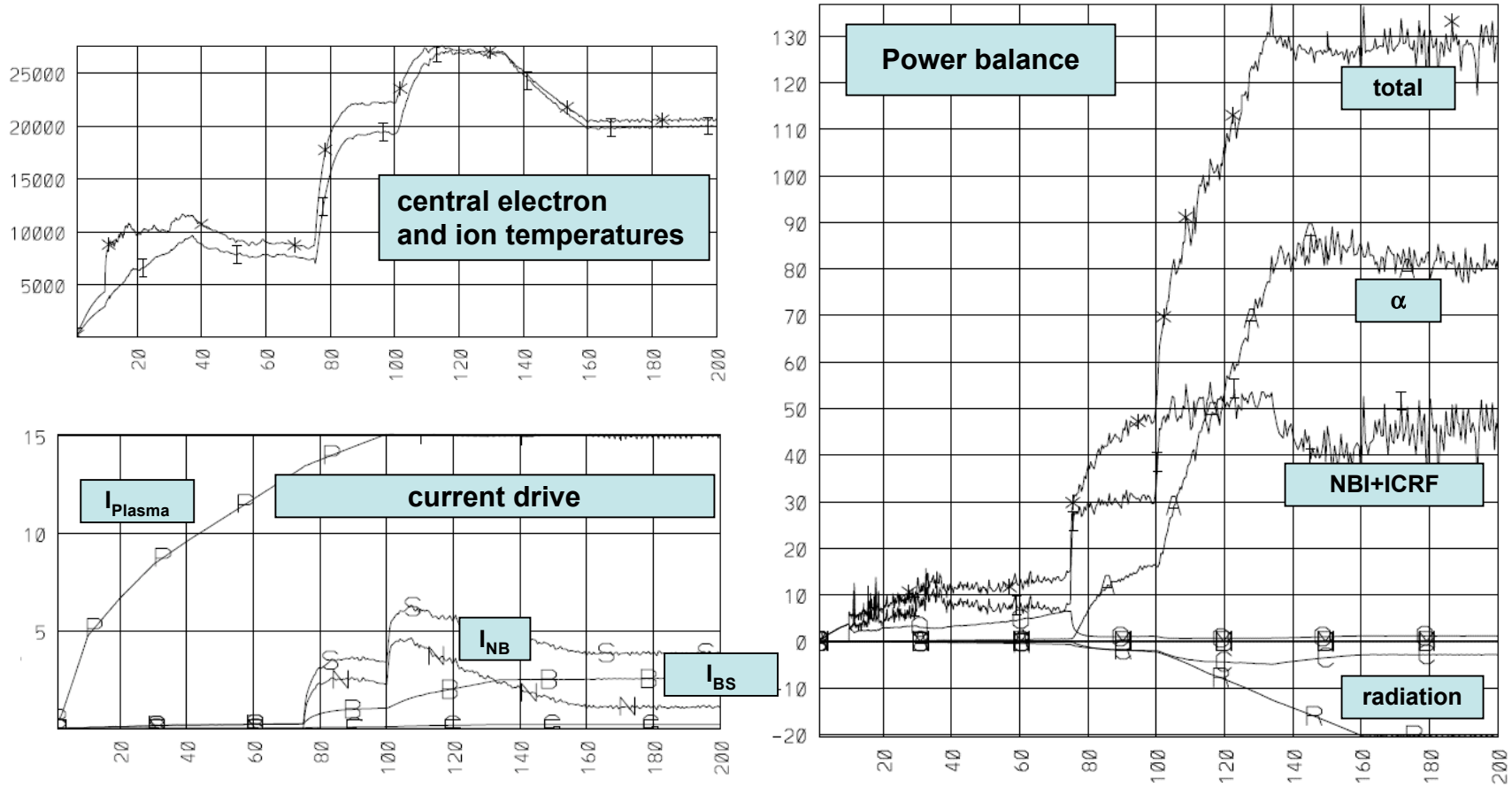
There are a number of planned operational scenario of ITER, e.g.the "*hybrid mode*" → achieve high fusion yield for long discharge time

These raise a number of critical questions:

- **Are these states achievable with the heating and current drive systems available?**
- **Are such states controllable to maintain stationary current, density and temperature profiles?**
- **Are such states stable and which sorts of instabilities are most dangerous?**
- **What are the beta (plasma pressure) limits and how close to the limits is it possible to operate?**
- **How sensitive is the performance to assumptions?**
  - **Energy transport**
  - **Pedestal properties**
  - **Plasma edge conditions**

Integrated simulations, in conjunction with experiments on present day experiments provide the physics basis for such planned scenarios

A single run such as below with minimal level of physics detail can take 6 weeks with present technology



TSC (Free-Boundary Equilibrium and Profile Advance)

TORIC (RF Ion Cyclotron) – 32 poloidal Fourier modes (poorly converged)

NUBEAM (neutral beam injection) – 1,000 Monte Carlo particles (poor statistics)

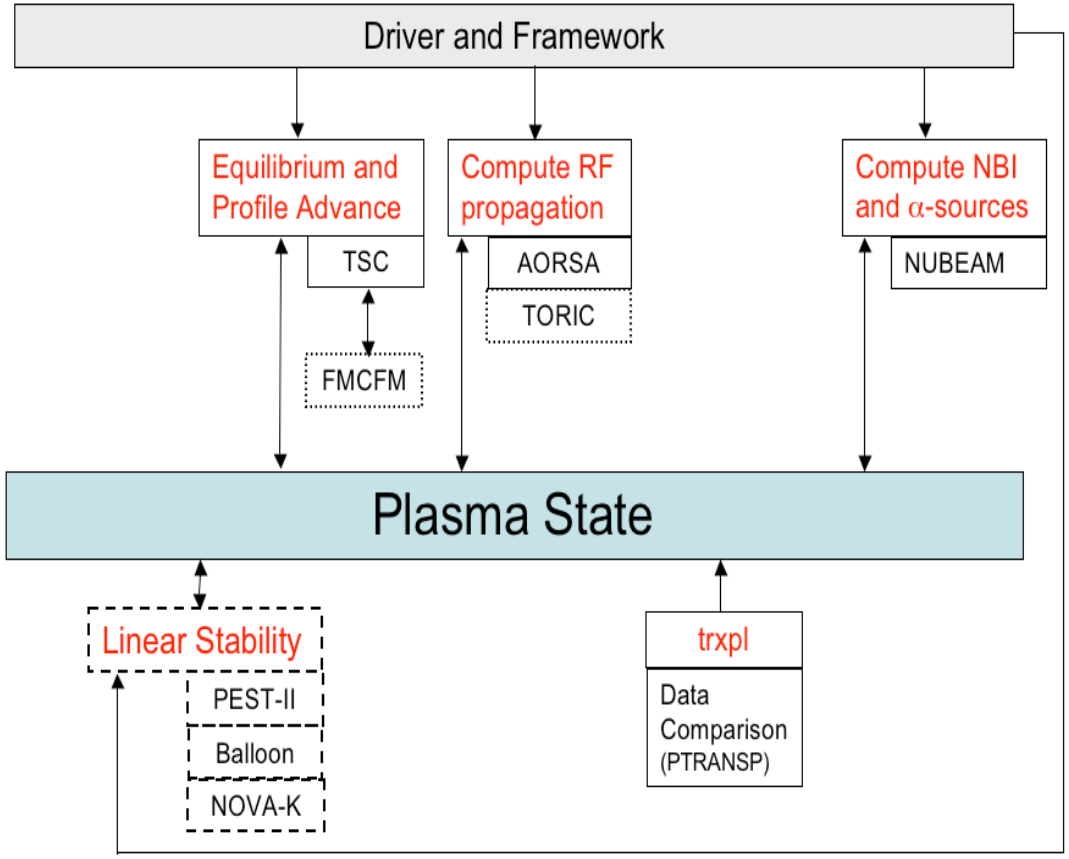
Near term goal: demonstrating the capability to use massively-parallel computers to greatly accelerate the integrated simulation process for ITER, while also improving the level of physics fidelity of the simulations.

Coupling of TSC (Free-Boundary Equilibrium and Profile Advance)

AORSA (massively parallel RF Ion Cyclotron solver) - 256×256 poloidal Fourier modes

TORIC (semi-spectral ICRF solver) – 149 poloidal modes, 409 radial nodes

NUBEAM (parallel neutral beam injection) – 1,000,000 Monte Carlo particles



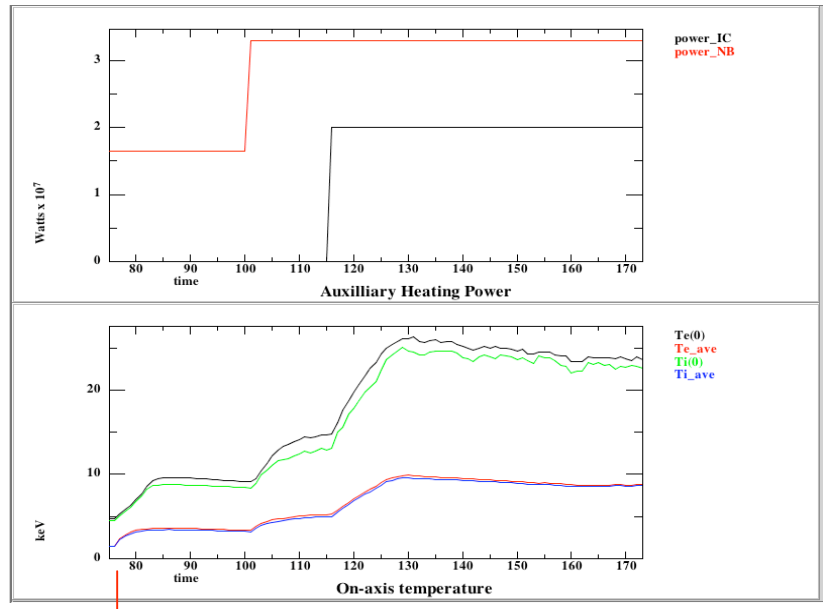
We are now carrying out simulations of the ITER device in a few days with high levels of levels of physics resolution

IPS simulation – ***1 day running time***

TSC – GLF23 transport

parallel AORSA – 256×256 (R, Z modes)

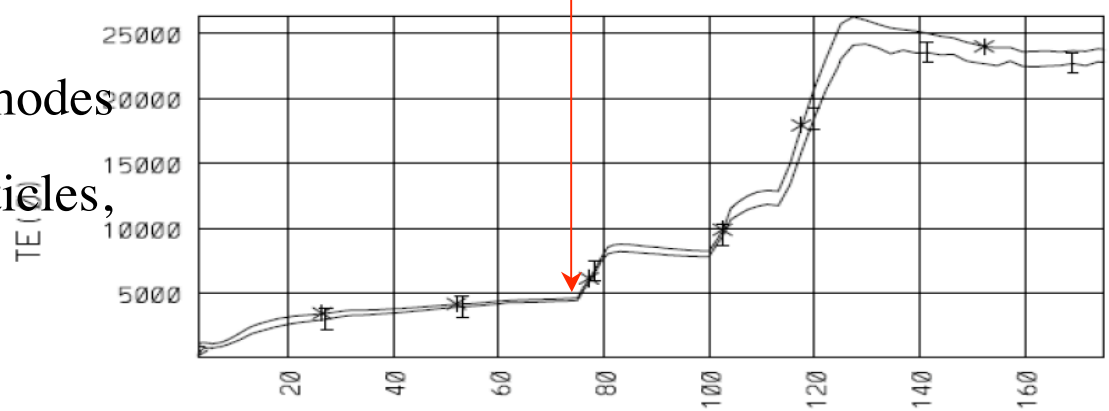
parallel nubeam – 10^6 NBI particles, 10^6 fusion particles



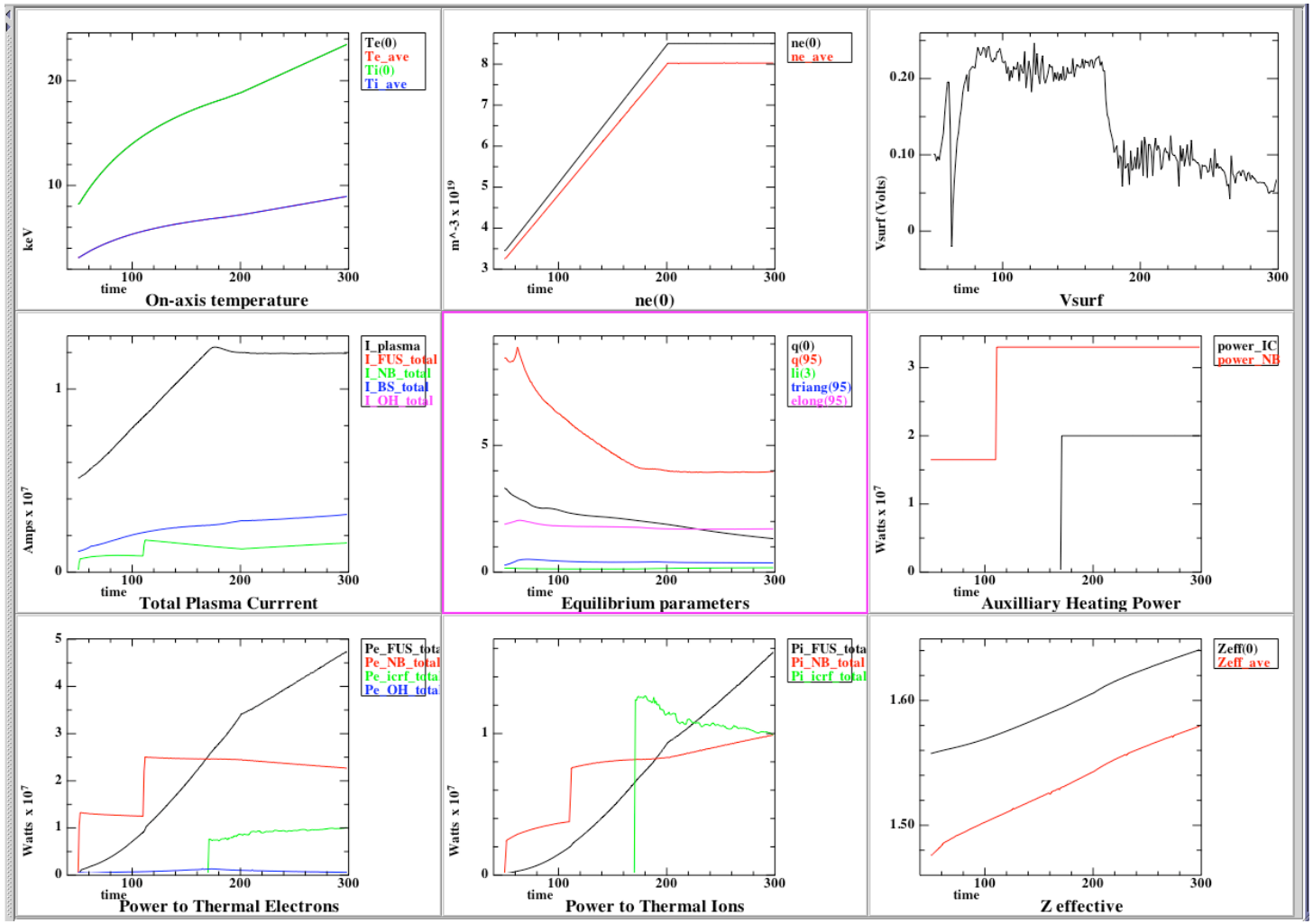
Previous TSC simulation – ***full 1,000 sec run, 6 week running time***

serial TORIC – 31 poloidal modes

serial nubeam – 10^3 NBI particles, 10^3 fusion particles



ITER 40700T10 benchmark – TSC, TORIC (147 poloidal modes), NUBEAM (10^6 MC particles for NBI & fusion) → 20 hr on 1,000 processors



We have repeated this calculation with NUBEAM and TORIC running simultaneously

Physics issues to be addressed

Scope – Demonstrate effectiveness of IPS for integrated simulations *(not to become the ITER simulation service)*

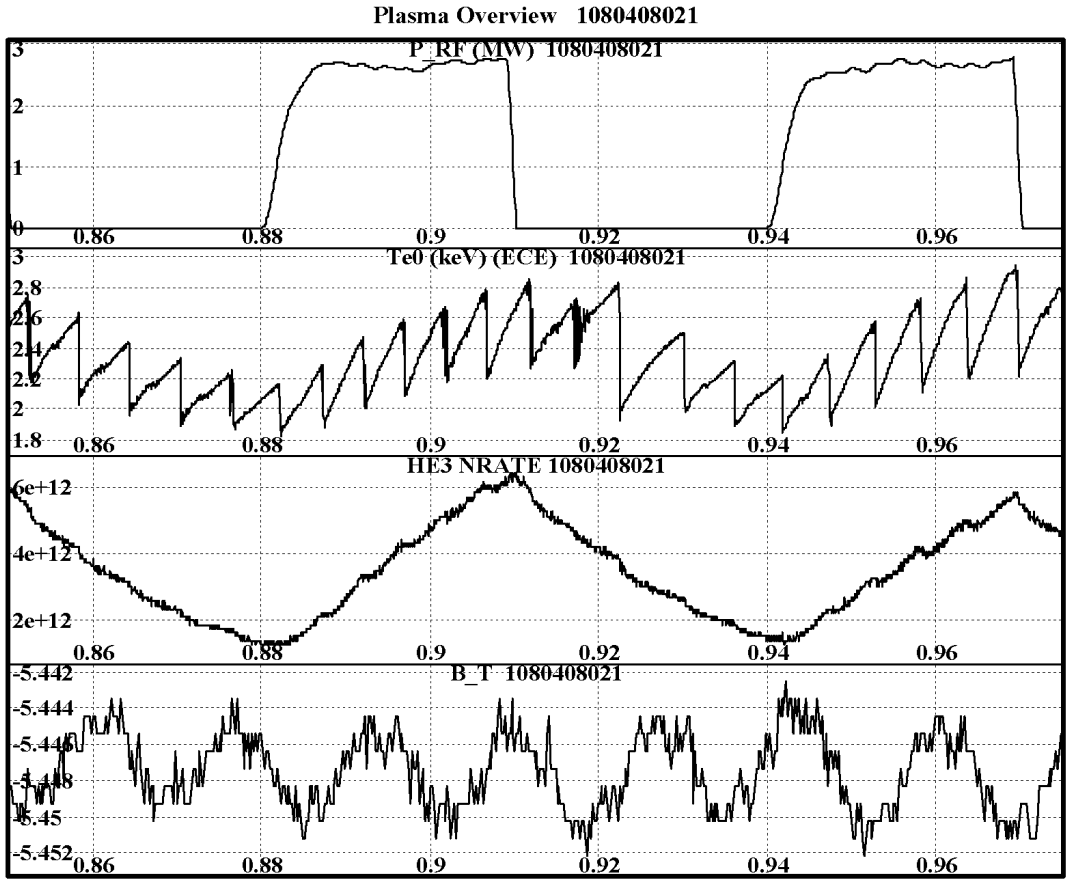
- **Participate in ITER benchmark activities as a V&V activity for IPS**
- **Support professional modelers (like Kessel, Budny, Murakami, Casper…) and get feedback on needs and design**
- **Exploit MPP and MCMD to get rapid simulation turn-around. Use MCDM to set up scans for optimization studies**

Components required – and desired improvements

- **TSC – parallel GLF23**
- **AORSA/TORIC – optimization, bench-marks in IPS, non-maxwellian beams and alphas (beyond $E_{||}$, $E_{\perp}$), distributions from particle list**
- **NUBEAM – optimization,**
- **GENRAY (EC) – complete component**
- **MCMD framework – restart capability**
- **CQL3D – if minority distribution evolution is needed**

Time dependent modeling of ICRF energetic minority tail formation on Alcator C-Mod

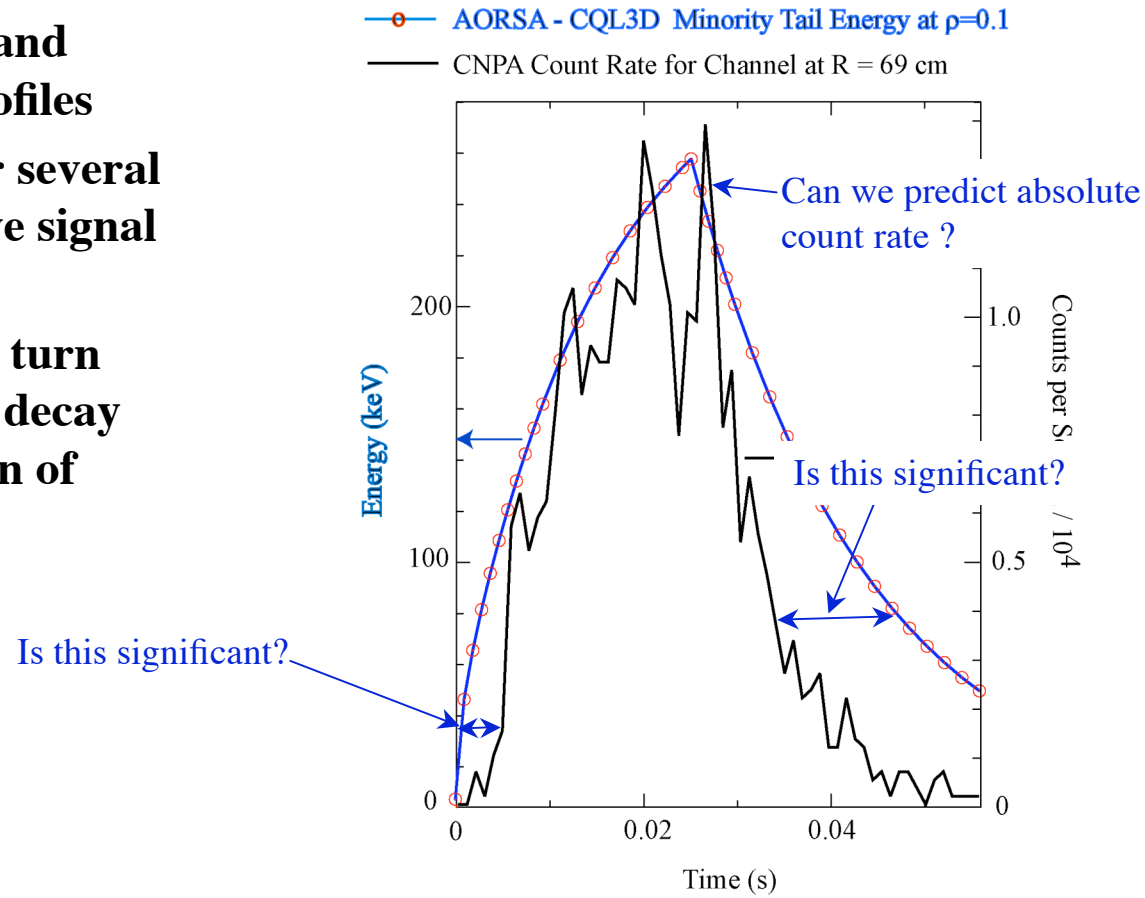
- Recent experiments with pulsed P_{ICRF} = 2.65 MW
- Performed at our request (mutually beneficial experimental collaboration)
- We are modeling these experiments for validation of the RF, Fokker Plank, transport, and sawtooth



Motivation is to improve understanding of RF energetic particle formation, sawtooth behavior and transport.

Comparison of simulated peak tail energy with experimental Neutral Particle Analyzer count rate

- **Simulation using AORSA2D and CQL3D with fixed plasma profiles**
- **CNPA count rate binned over several square-wave pulses to improve signal to noise ratio**
- **Reasonable agreement on tail turn on, not so good agreement on decay time (profile effect?, indication of energetic particle loss? …)**



This is directed toward fully coupled simulations with TSC, AORSA2D, CQL3D, MHD stability, with synthetic diagnostic of NPA measurements

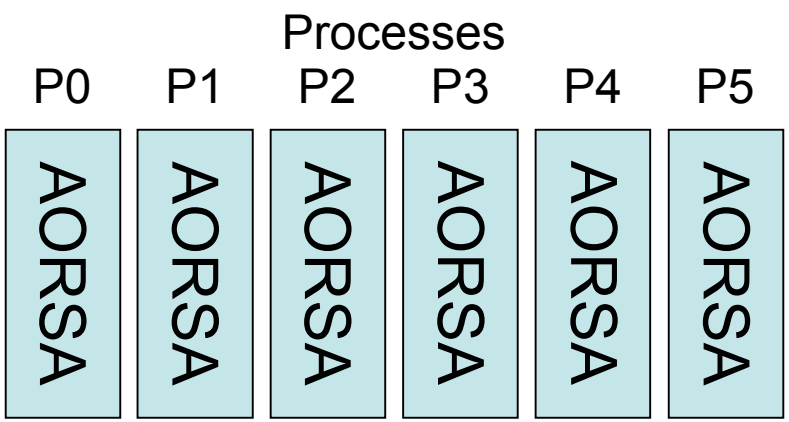
Framework stuff to do in the near term

- **Implement restart capability**
  - **Add checkpoint and restart methods to components**
- **Implement formal regression testing**
- **Improve documentation**
  - **See draft users guide by Samantha Foley**
- **Replace cumbersome "next plasma state" method of cross-component communication with messaging services**
- **Collect simulation input data in svn "data" tree**
- **Improve portal data management and meta-data handling**

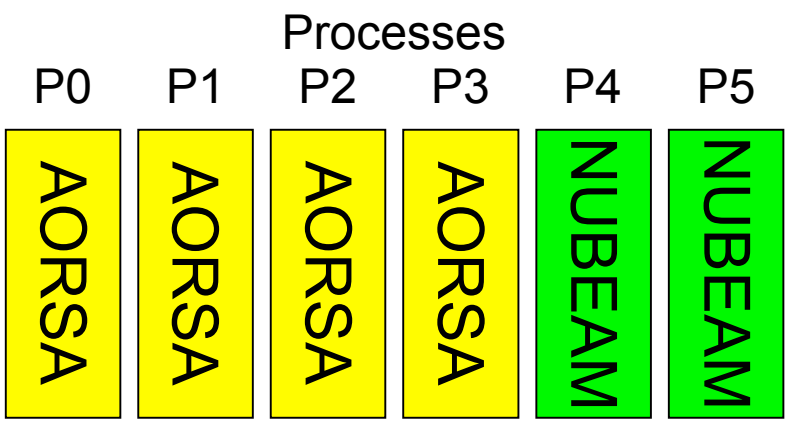
Extending IPS with infrastructure to address load balance issues in simulations with coupled coarse-granularity components – MCMD?

- **The *multiple-component multiple-data* execution model is the component equivalent of multiple-program multiple-data (MPMD)**
  - **Most individual parallel programs are *single*-program multiple-data (SPMD) or SCMD if they are component-based**
  - **A term used in the Common Component Architecture community**
- **An execution model that allows multiple *parallel tasks* (components) to execute *concurrently***
- **Simple pictorial example:**

Single Parallel Task

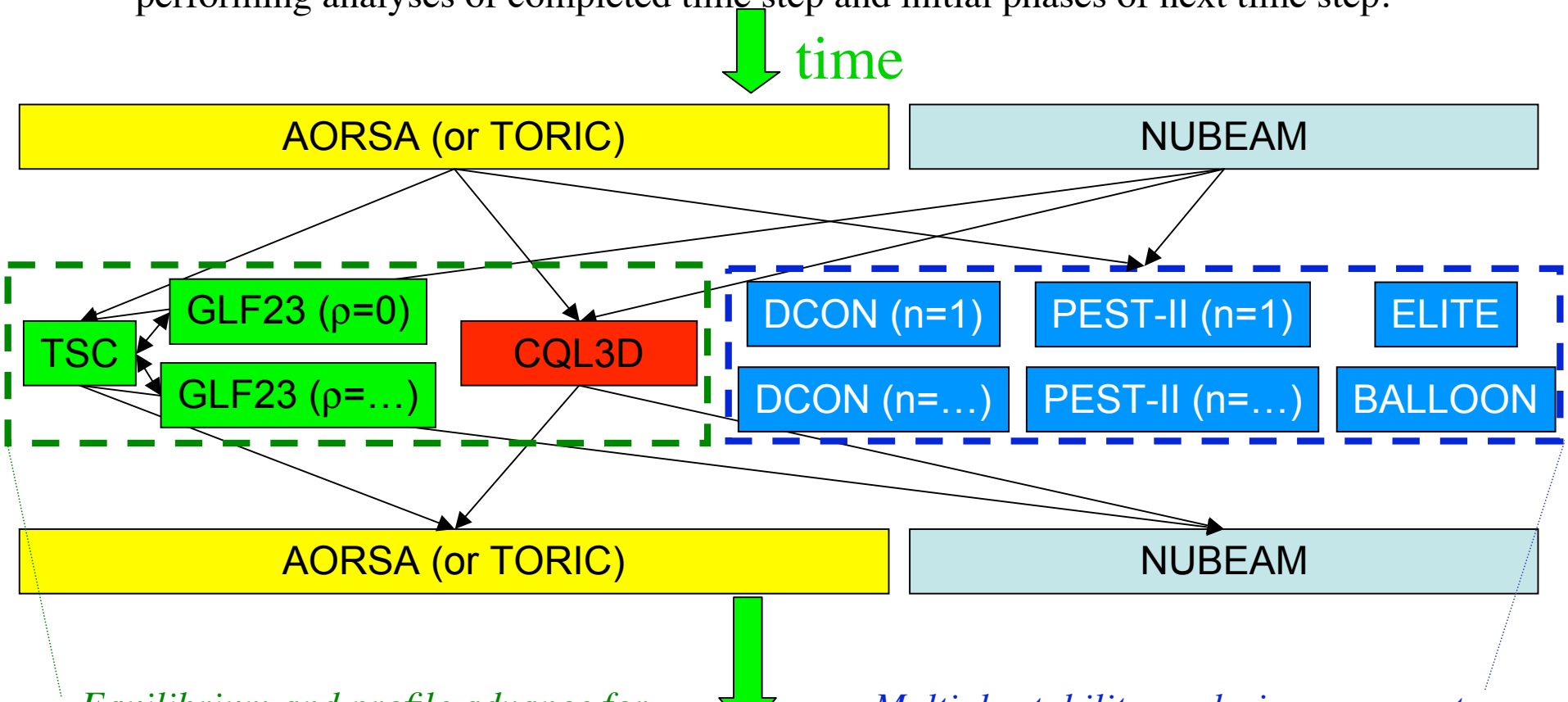


Concurrent Parallel Tasks (MCMD)



Simulations such as ITER scenario investigations can benefit from MCMD by minimizing time in (near) serial operations

Highly parallel components alternating with many smaller components, performing analyses of completed time step and initial phases of next time step.



Equilibrium and profile advance for step t, including parallel anomalous transport tasks for each flux surface, all running concurrently with the Fokker Planck component.

Multiple stability analysis components running on multiple toroidal modes, all running concurrently on t-1 results.

Stuff to do in the longer term

- **Expand physics studies and complete those already on the stack**
- **Explore more complex time stepping approaches**
  - **Scheduled and unscheduled events during a time step**
  - **Inserting intermediate time steps**
- **Explore more advanced uses of concurrency**
  - **Parallel in time and variants**
  - **Staggering of source updates**
  - **Multiple concurrent simulations**