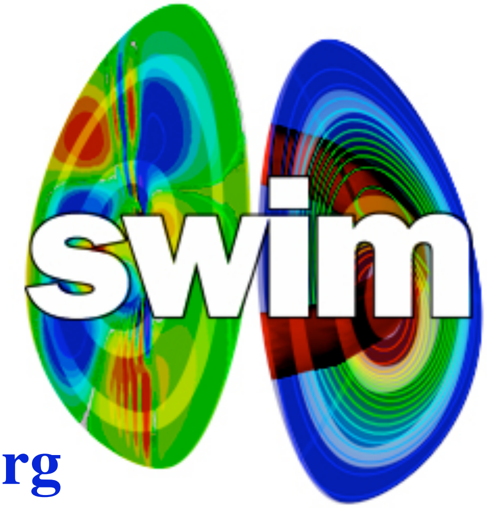
Center for Simulation of Wave Interactions with MHD (SWIM)

PSACI PAC Meeting
June 7-8, 2007

D. B. Batchelor, L. A. Berry, S. P. Hirshman, W. A. Houlberg, E. F. Jaeger, R. Sanchez – *ORNL Fusion Energy*
D. E. Bernholdt, E. D'Azevedo, W. Elwasif, S. Klasky– *ORNL Computer Science and Mathematics*
S. C. Jardin, E. Feibush, D. McCune, J. Chen, L. P Ku, M. Chance, J. Breslau – *PPPL*
R. Bramley – *Indiana University,* **D. Keyes** – *Columbia University,* **D. Aswath, M. Choi, D. P. Schissel** – *General Atomics,*
R. W. Harvey – *CompX,* **D. Schnack** – *U. Wisconsin,* **J. Ramos, P. T. Bonol, J.Wright** – *MIT*
S. Kruger – *TechX*, **G. Bateman** – *Lehigh University,*
Unfunded participants:
L. Sugiyama – *MIT,* **C. C. Hegna** – *University of Wisconsin,* **H. Strauss** – *New York University,* **P. Collela** – *LBNL*
H. St. John – *General Atomics,* **A. Kritz** – *Lehigh Univ.*

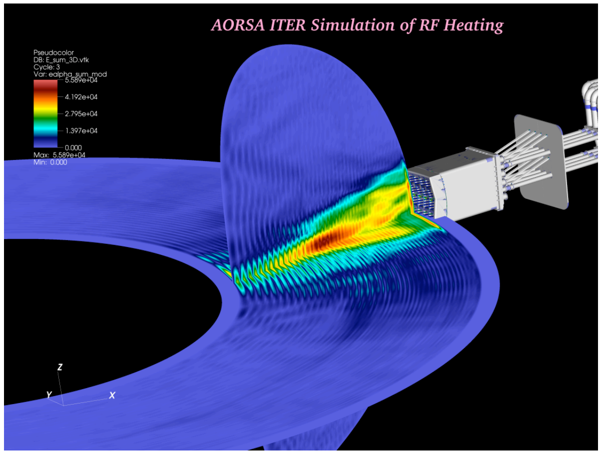
- What is this project? (for non-fusion types)
- Progress on scientific goals
- Role of collaborations
- Role of leadership-class computing resources
- Plans to address 3-year project goals



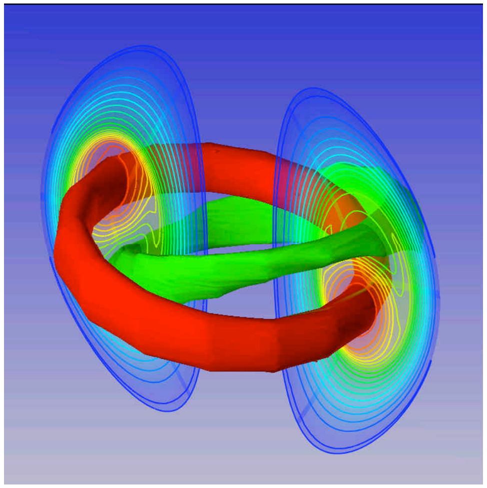
See our fun website at: www.cswim.org

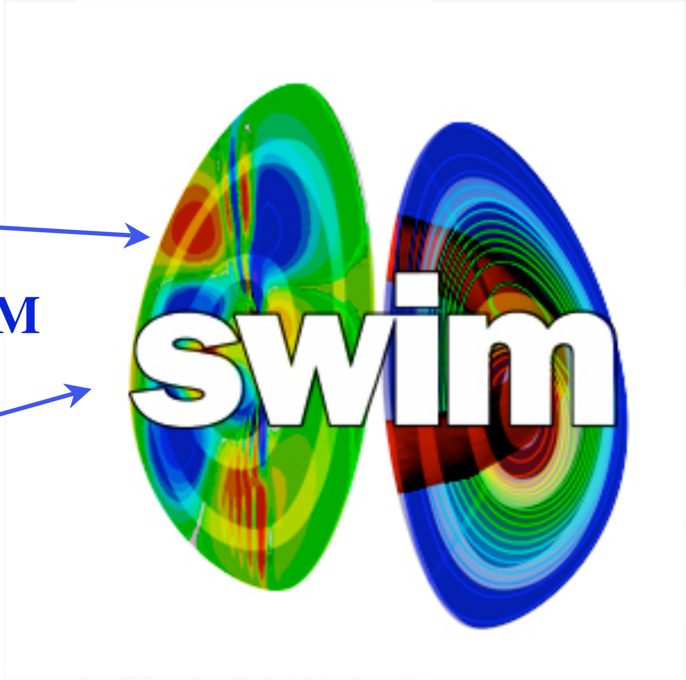
SWIM brings together two mature fields of fusion computation

High power wave-plasma interactions – CSWPI



Extended MHD – CEMM



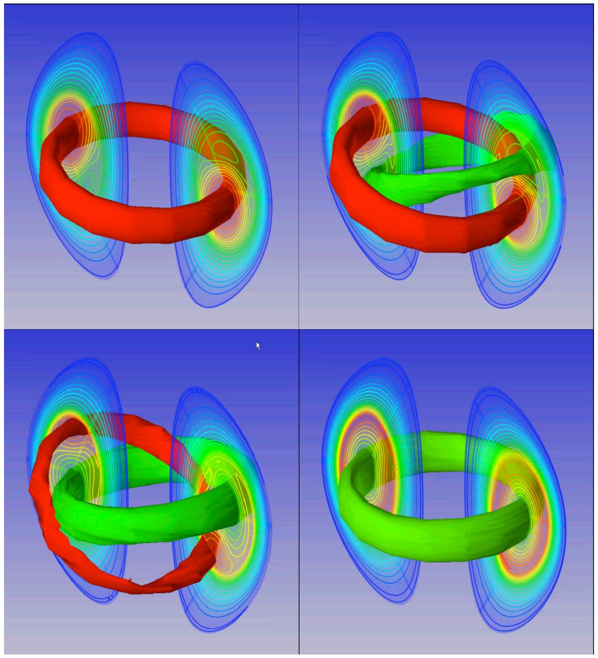


Partnership of DOE OFES and OASCR under the SciDAC

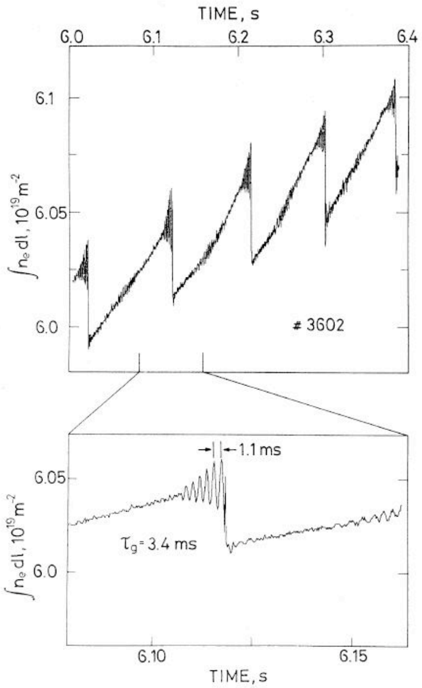
Why couple these particular two disciplines?

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

Sawtooth oscillations can *limit* plasma performance and represent one of the greatest uncertainties for ITER



M3D simulation for CDX-U of internal reconnection event

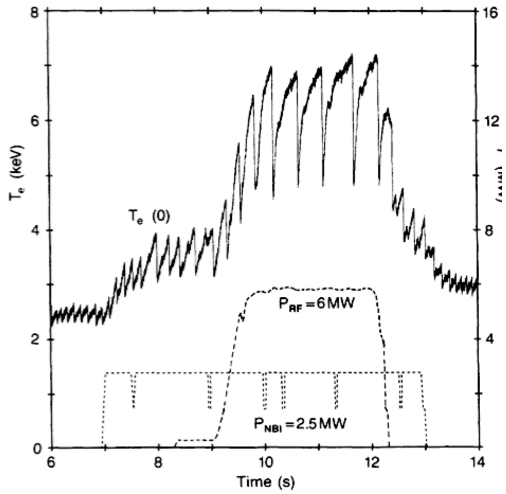


Crash phase is several orders of magnitude faster than growth

- **Sawteeth can directly limit plasma performance – particularly when energetic particles (e.g. fusion alphas) produce large amplitude oscillations**
- **Sawteeth are correlated with other instabilities**
  - **Neoclassical tearing modes – provide seed island**
  - **Disruptions**

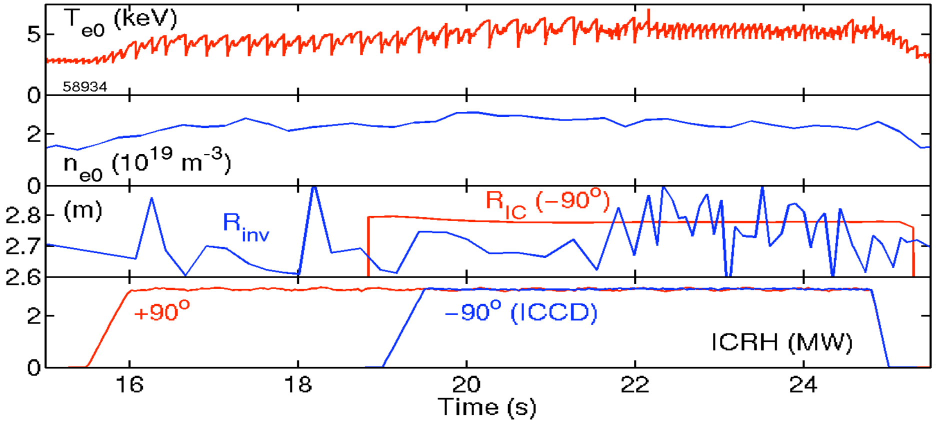
There are several experimentally demonstrated mechanisms by which RF waves can control sawtooth behavior

ICRF stabilization on JET



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

Sawtooth control on JET with Minority Current Drive on JET



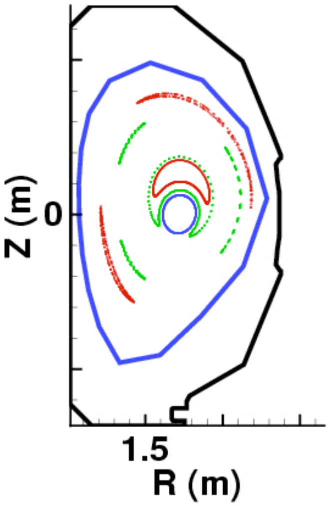
- **ICRF minority current drive can either increase or decrease period and amplitude**
- **Likely stabilization/destabilization mechanism – RF modification of current profile**

- **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**

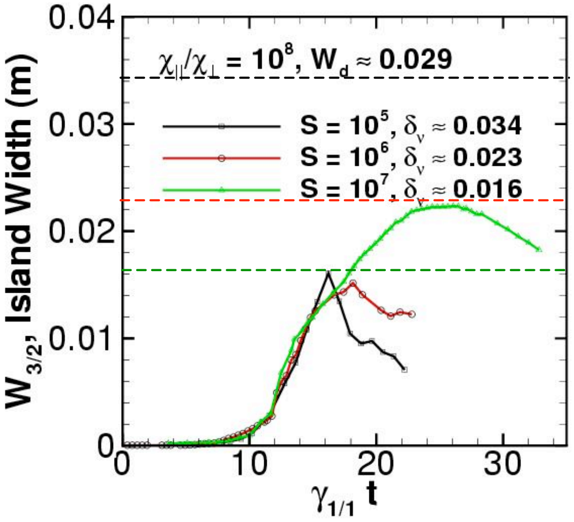
Neoclassical Tearing Modes (NTMs) are a major concern for ITER. They can *limit* plasma pressure in tokamaks and can lead to *disruption*

Simulation with NIMROD

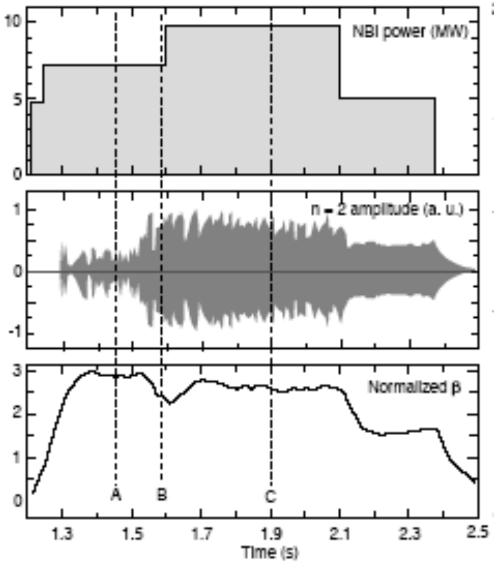
NTM onset and confinement loss measured on ASDEX-U



Sawtooth and secondary islands



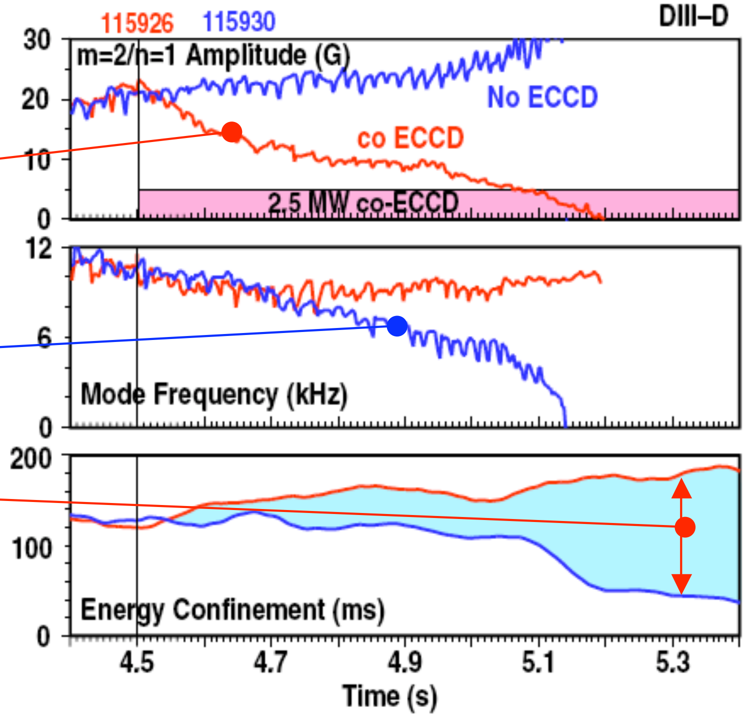
Secondary island growth
Varying S



- **NTMs are slowly growing modes that break up the closed, nested flux surfaces needed for confinement**
- **NTMs cause growth of small, pre-existing magnetic islands – e.g. produced by sawteeth or ELMs**
- **Saturated modes can slow in rotation, *lock*, and lead to disruption**

It has been demonstrated experimentally that suppression of 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**



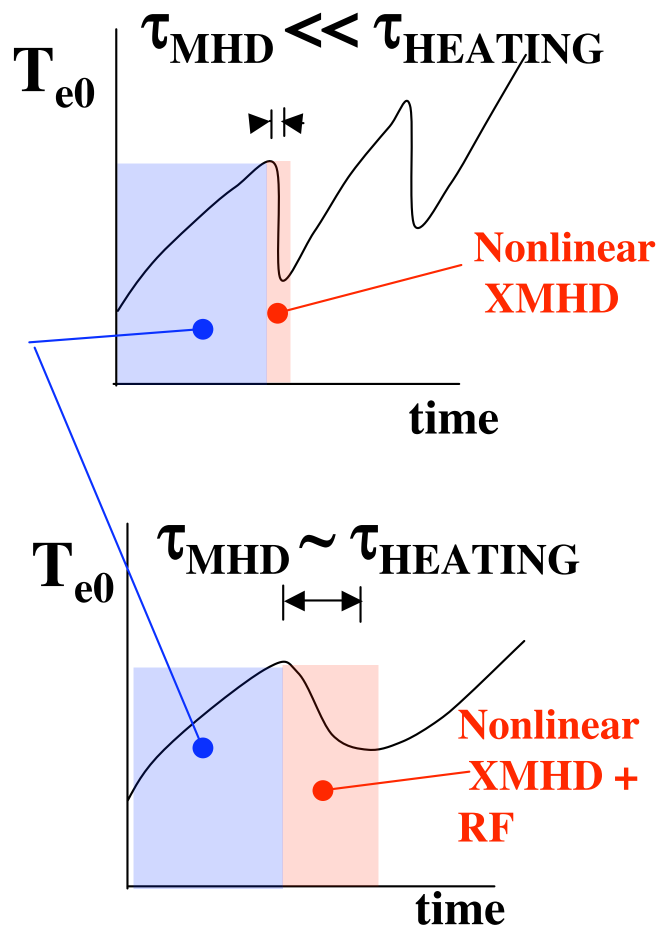
R. Prater APS 2003

- **Empirical scaling of NTM pressure limits in ITER leave no 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**
- **RF (mainly ICRH) drives slow plasma evolution, sets initial conditions for fast MHD event**
- **Example: sawtooth crash**



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**

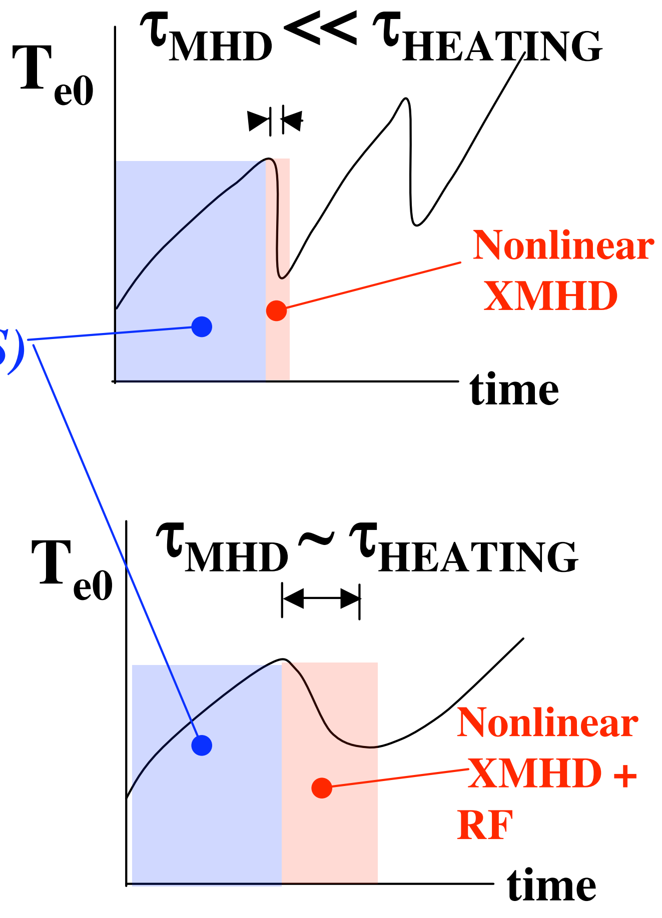
These two regimes are related

- **Fast sawtooth crash can provide seed island for NTM growth**
- **Slow growth of NTM island can lead to fast disruption events**
- **Calculation of slow ramp of sawtooth, with incomplete reconnection or persistent islands, may actually require the same capabilities as NTM evolution**

A major element of the SWIM project is development of an Integrated Plasma Simulator (IPS)

Objectives of Integrated Plasma Simulator

- **Simulate slow time-scale plasma evolution for SWIM – Fast MHD campaign**
- **Provide advanced simulation capabilities for burning plasma research, beyond SWIM project**



Integrated Plasma Simulator (IPS)

- **Serve as test-bed for tighter component coupling required in SWIM – Slow MHD campaign**
- **Gain experience in computer science and mathematics issues to be faced in a comprehensive fusion simulation**
- ***Develop a flexible, extensible computational framework capable of coupling in any fusion code and able to evolve to a complete simulation capability***

Progress on Scientific Goals and Plans for following year

- **Completion of design for first version of Integrated Plasma Simulator and its implementation**
- **Design and implementation of SWIM web portal**
- **Progress on physics studies**

SWIM Software Goals & Requirements

- Develop an Integrated Plasma Simulator (IPS) supporting…
  - Fusion simulation science needs (near term) → two SWIM physics campaigns, device modeling
  - Design study for future *full* integrated simulations (long term)
- Explore interoperability and interchangeability of components in common infrastructure
  - Looking towards a flexible "toolkit" for integrated modeling
  - Useful for V&V
- Maximize (re)use of existing code
- Minimize changes to physics codes for non-physics driven reasons
  - → Avoid bifurcation of physics components – not different SWIM/stand alone versions
  - → Ease of debugging
- Capable of running on high-end systems from the start

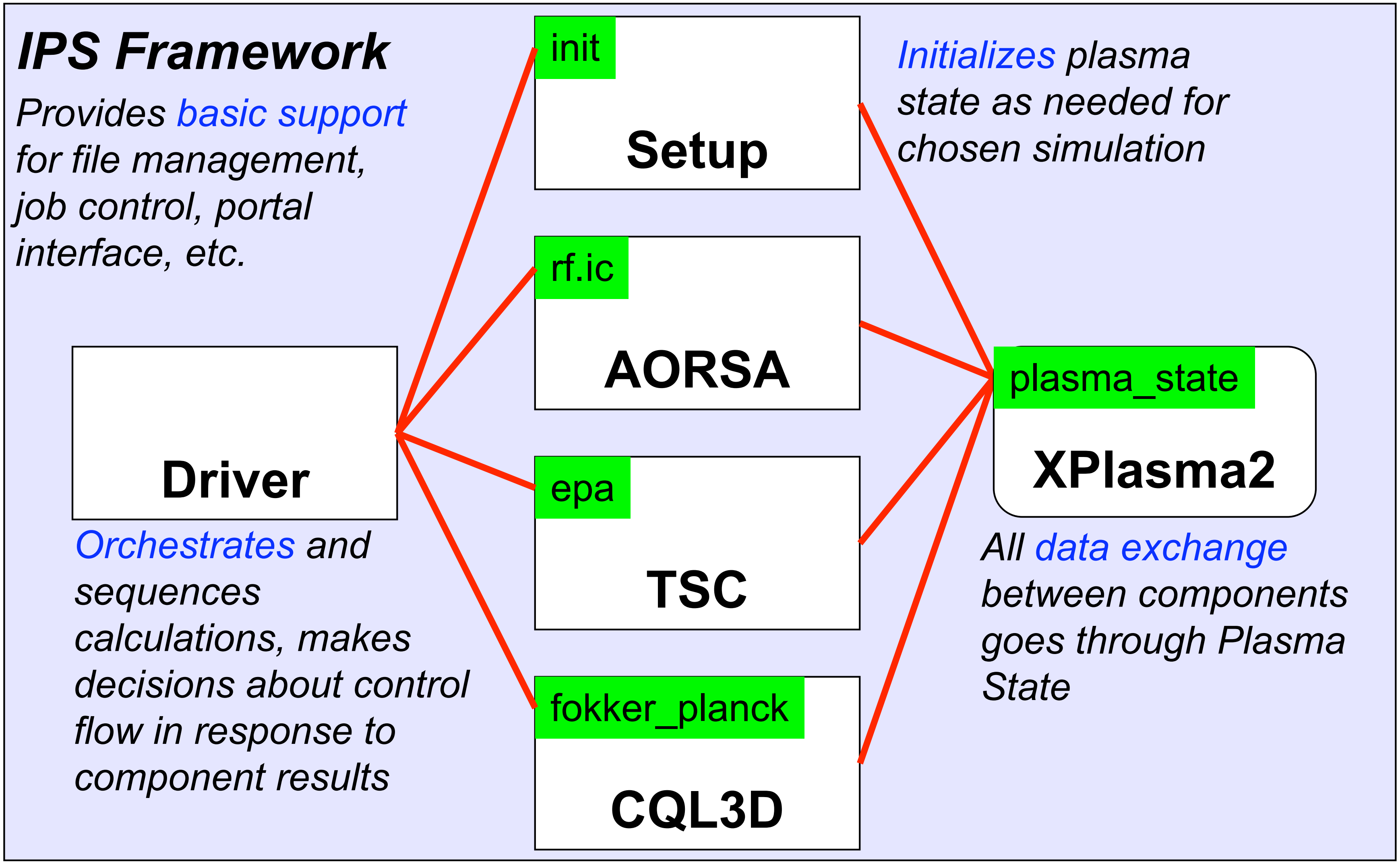
IPS Design Approach

- Framework/component architecture
  - Components initially existing codes wrapped up
  - Framework provides basic utility services
  - “Driver” component orchestrates simulation
  - “Plasma State” component is official data manager
- Emphasize interfaces
  - Carefully defined the mathematical functionality of the components
  - Components providing the same functionality should do so through the same interface
  - Project intentionally includes at least two distinct codes for most classes of functionality
- Start simple, increase sophistication as science needs dictate
  - File-based communication → in-memory data exchange
  - Whole codes wrapped with scripts → finer-grain native-language components
  - Project-specific framework → Common Component Architecture compliant framework

IPS Framework Features

- **Provides environment in which components are instantiated and executed**
  - **Manages association between interfaces and components implementing them**
- **Provides basic services to components**
  - **Configuration (input) management → simulation details specified in configuration file**
    - **Easily extensible to additional components w/o outside changes**
    - **Plan "Tokamak machine configuration file" to standardize machine specific**
  - **File management abstraction**
    - **Manages working directories**
    - **Temporary and permanent files, stored separately**
  - **Job management (parallel execution)**
  - **Interface with web portal**
    - **Without changes to underlying components**
    - **Framework can run without portal**
  - **Currently all part of a single `services` interface**
- **Framework implemented in Python**

The IPS is a system for composing fusion simulation applications → Schematic of an IPS Application



IPS Framework

Provides basic support for file management, job control, portal interface, etc.

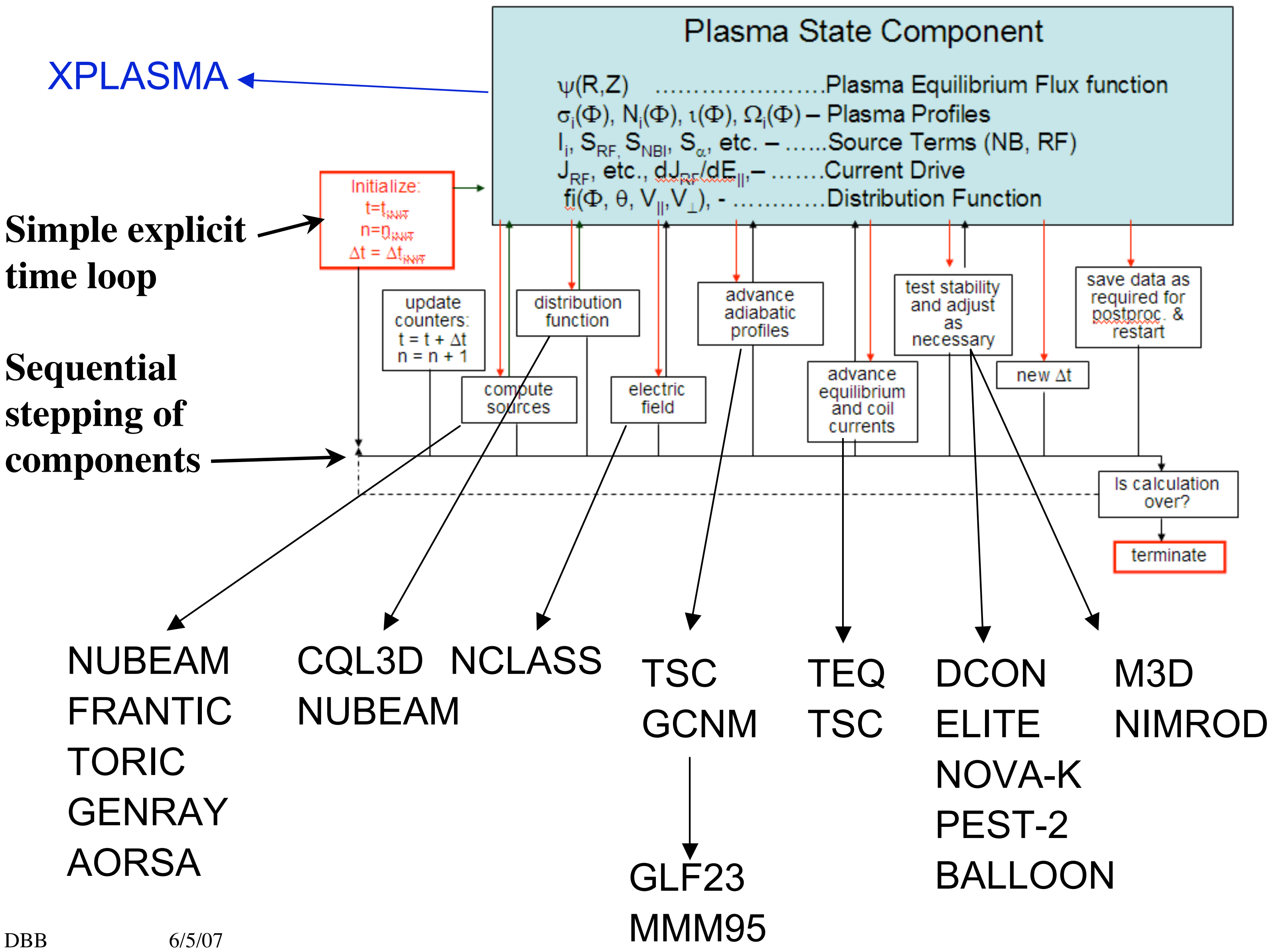
Initializes plasma state as needed for chosen simulation

Orchestrates and sequences calculations, makes decisions about control flow in response to component results

All data exchange between components goes through Plasma State

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

Planned components and implementations *(from 2006 PSACI)*



XPLASMA

Simple explicit time loop

Sequential stepping of components

Plasma State Component

$\psi(R,Z)$ ……………………Plasma Equilibrium Flux function

$\sigma_i(\Phi)$, $N_i(\Phi)$, $\iota(\Phi)$, $\Omega_i(\Phi)$ – Plasma Profiles

I_i, S_{RF}, S_{NBI}, S_α, etc. – ……Source Terms (NB, RF)

J_{RF}, etc., $dJ_{RF}/dE_{||}$, – ……Current Drive

$fi(\Phi, \theta, V_{||}, V_\perp)$, - …………Distribution Function

Initialize:
$t=t_{INIT}$
$n=n_{INIT}$
$\Delta t = \Delta t_{INIT}$

update counters:
$t = t + \Delta t$
$n = n + 1$

distribution function

compute sources

electric field

advance adiabatic profiles

advance equilibrium and coil currents

test stability and adjust as necessary

new Δt

save data as required for postproc. & restart

Is calculation over?

terminate

NUBEAM
FRANTIC
TORIC
GENRAY
AORSA

CQL3D
NUBEAM

NCLASS

TSC
GCNM

GLF23
MMM95

TEQ
TSC

DCON
ELITE
NOVA-K
PEST-2
BALLOON

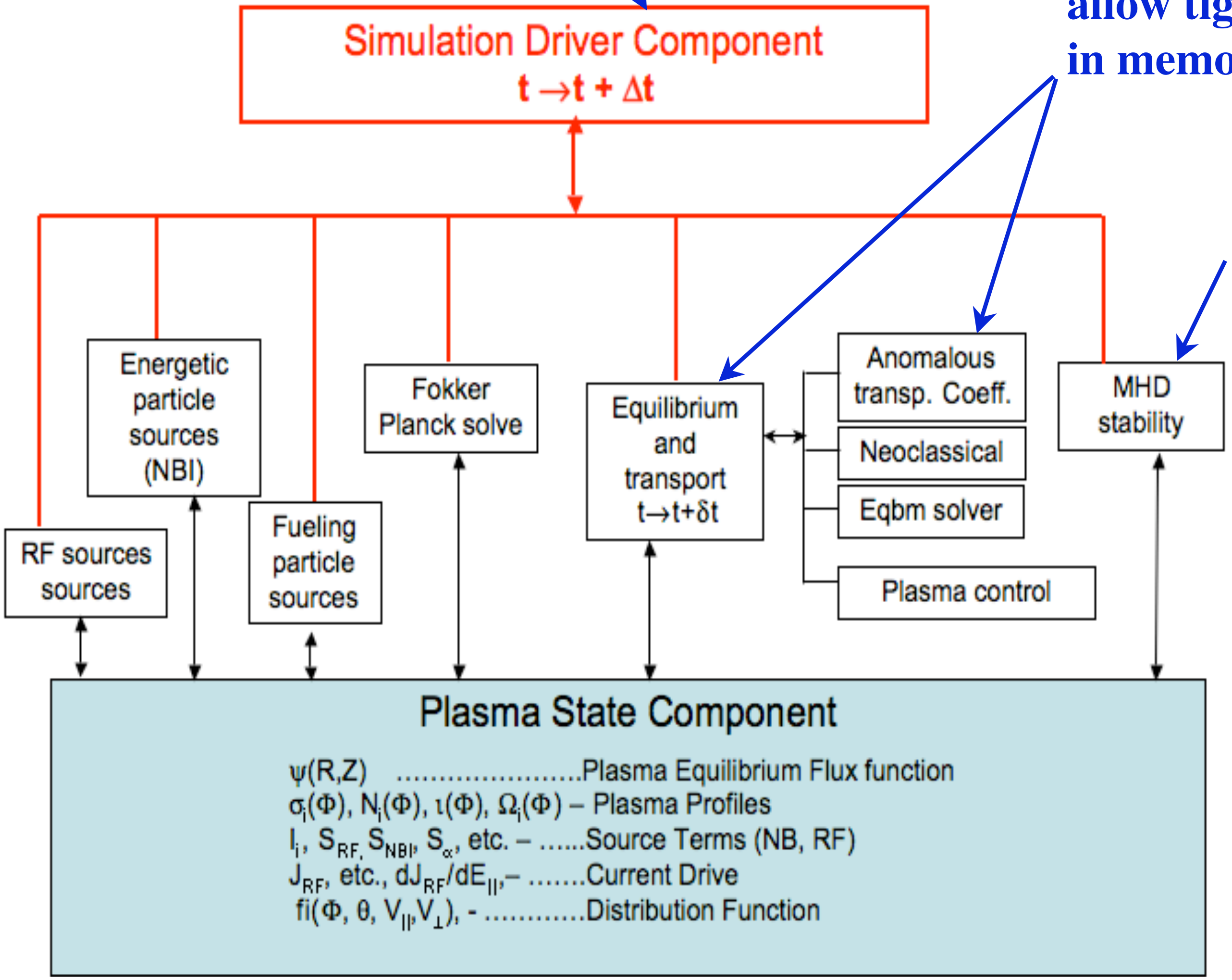
M3D
NIMROD

New design allows for increased flexibility of simulation control and linkage

Driver physics layer allows extensibility, flexibility in controlling simulation

Composite components: allow tighter coupling, in memory communication

Same interface for linear/nonlinear MHD



Simulation Driver Component

$t \rightarrow t + \Delta t$

Energetic particle sources (NBI)

Fokker Planck solve

Equilibrium and transport $t \rightarrow t + \delta t$

Anomalous transp. Coeff.

Neoclassical

Eqbm solver

Plasma control

MHD stability

RF sources sources

Fueling particle sources

Plasma State Component

$\psi(R,Z)$ ....Plasma Equilibrium Flux function

$\sigma_i(\Phi)$, $N_i(\Phi)$, $\iota(\Phi)$, $\Omega_i(\Phi)$ – Plasma Profiles

I_i, S_{RF}, S_{NBI}, S_α, etc. – ......Source Terms (NB, RF)

J_{RF}, etc., $dJ_{RF}/dE_{||}$,– ....Current Drive

$fi(\Phi, \theta, V_{||}, V_\perp)$, - ....Distribution Function

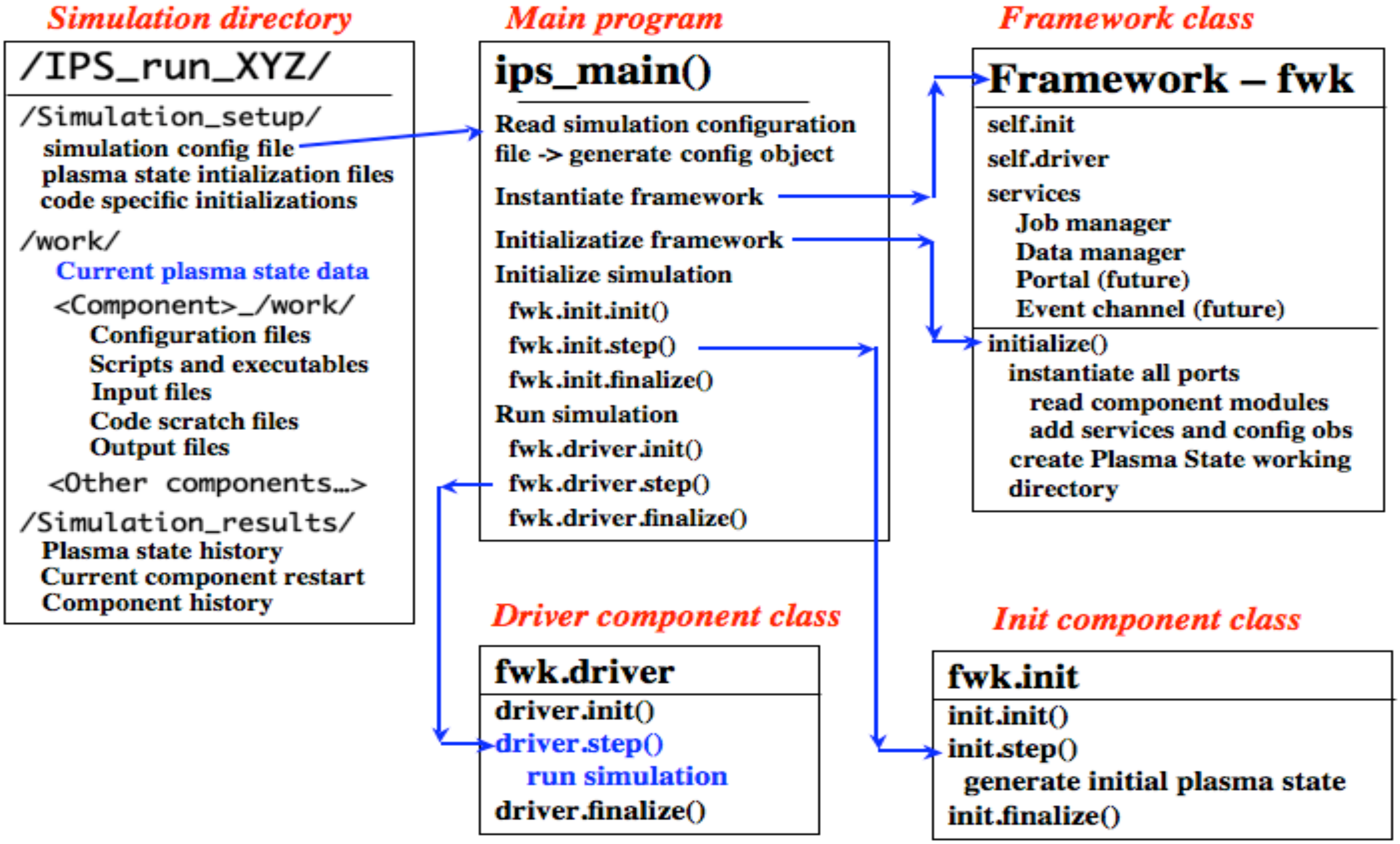
The Basic Component Interface

→ Observation Most coarse-grain (application-level) components in a time-stepped simulation can be expressed with just a few basic operations

- `Init(ialize)`
  - **Prepare to run component for a series of time steps**
- `Step`
  - **Do whatever computation is appropriate to the current time step**
- `Finalize`
  - **Clean up at end of run**

In IPS almost everything (including the framework, simulation initialization and driver) is an instance of abstract class 'Component' and supports this interface.

IPS framework



Simulation directory

/IPS_run_XYZ/

/Simulation_setup/
  simulation config file
  plasma state intialization files
  code specific initializations

/work/
  Current plasma state data
  <Component>_/work/
    Configuration files
    Scripts and executables
    Input files
    Code scratch files
    Output files
  <Other components...>

/Simulation_results/
  Plasma state history
  Current component restart
  Component history

Main program

ips_main()

Read simulation configuration file -> generate config object

Instantiate framework

Initializatize framework

Initialize simulation
  fwk.init.init()
  fwk.init.step()
  fwk.init.finalize()

Run simulation
  fwk.driver.init()
  fwk.driver.step()
  fwk.driver.finalize()

Framework class

Framework – fwk

self.init
self.driver
services
  Job manager
  Data manager
  Portal (future)
  Event channel (future)

initialize()
  instantiate all ports
    read component modules
    add services and config obs
  create Plasma State working directory

Driver component class

fwk.driver

driver.init()
driver.step()
  run simulation
driver.finalize()

Init component class

fwk.init

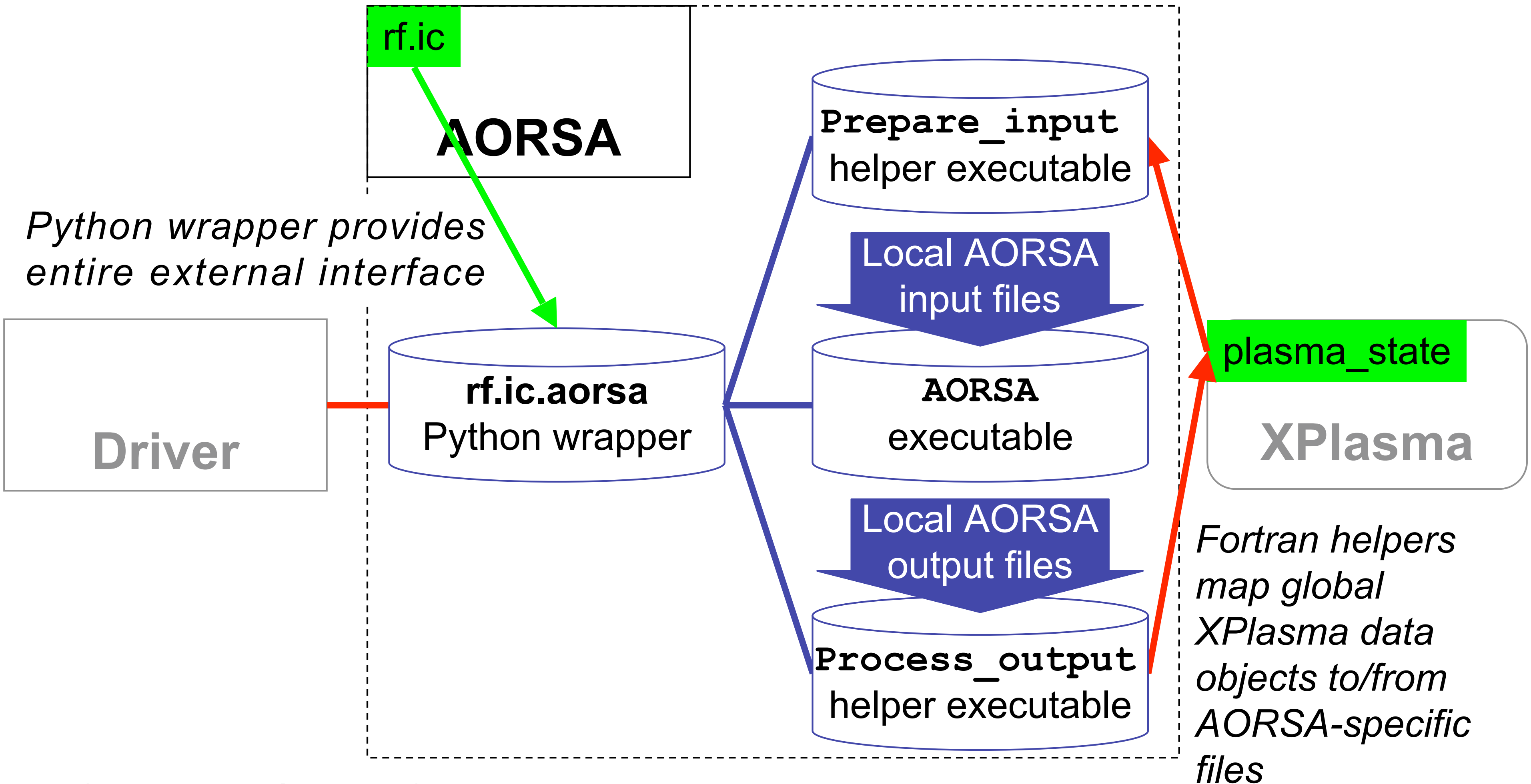
init.init()
init.step()
  generate initial plasma state
init.finalize()

Two distinguished components are called by ips_main: init, driver

Sketch of Basic IPS Driver

```
Read in simulation configuration
Setup initial plasma state
Foreach component c (in appropriate order)
    Call c.init
For $t = t_i$ to $t_f$
    Foreach component c (in appropriate order)
        Call c.step(t)
    Commit this time step to Plasma state
    Stage output files
Foreach component c (in appropriate order)
    Call c.finalize
```

Schematic typical component internals – rf_ic implemented by AORSA



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

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

- Fortran 90 Module – built on PPPL's XPLASMA2 library
  - Distributed by NTCC, used in TRANSP
  - netCDF for backend storage
- Formally a library at present, not a component
- Very simple user interface → functions: get, store, commit
- Other powerful XPLASMA functions available, but not required → e.g grid interpolation
- 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 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

Our arsenal of developed components includes

- **RF Solve – ion cyclotron two implementations**
  - **AORSA2D**
  - ***TORIC – testing***
- **Fokker Planck Solve**
  - **CQL3D**
  - **NUBEAM not yet a wrapped component but car run as TSC/TRANSP coupling**
- **Equilibrium and profile advance (EPA)**
  - **TSC implementation**
  - **TSC replay implementation – reads state file from previously generated simulation directory tree – useful for testing and time-slice analysis by other components**

Coming soon

- **Linear MHD component – multiple implementations: BALLOON, DCON, PEST-I and PEST-II, NOVA-K, Nonlinear MHD M3D/NIMROD**
- **MONITOR– accumulates state data into time series for plotting/monitoring with Elvis**
- **RF EC and LH components – implemented by GENRAY**
- **Fokker Planck solves – implemented by ORBIT-RF, NUBEAM**

Plan for coming year – IPS

- Continued implementation and testing of physics components
- Test harness for components and plasma state
- More extensive use of NLCF computers
- Implementation of more elaborate time stepping algorithms
  - Only the driver has the *big picture* of the simulation, not the components
  - Responding to un-planned events in the middle of a time step that require action by driver or other components – e.g. sawtooth events, plasma control actions, runtime faults
  - Development of an information rich data object returned to driver from components
  - Requires extending the basic interface used so far – interfaces no longer generic across components
- More use of in-memory data exchange → may require rethinking Plasma State
- Managing parallelism → some components serial, others highly parallel
- Analyze migration to standard component architecture (e.g. CCA)
  - Will facilitate leveraging outside components – I/O, math libraries, etc
  - Current design is intentionally close to CCA

Progress on Scientific Goals and Plans for following year

- **Completion of design for first release of Integrated Plasma Simulator and its implementation**
- **Design and implementation of SWIM web portal**
- **Progress on physics studies**

A web portal is being developed for IPS → to ease access security, submission, job monitoring, meta-data management

- Security (transparent to the user)
  - Authentication : User signs on securely with FusionGrid credentials- an id and password, operated by ESnet
  - Authorization : Component service provider checks to see if authenticated user is authorized to access its service
- Component Submission and Monitoring
  - Platform to securely initialize and instantiate simulation runs on remote systems at PPPL (mhd, viz) and ORNL (Jaguar) through the IPS Framework
  - User ability to monitor the run launched remotely viw events notifying the user on the status of the component run
  - Browser visualizations / displays of run results
- Meta-Data Management
  - Automated generation and storage of metadata resulting from the runs to facilitate users to quickly retrieve information on runs made
  - Monitor run directories and store/extract information such as location, size, timestamp, owner as external meta-data and physics quantities of interest to users as internal meta-data

Prototype is operational

Security Model

The resource you requested requires you to authenticate.

User ID

Password

Login gives you 8-hour access without repeat login to protected Web resources.

WARNING: Protect your privacy! Prevent unauthorized use!
Completely exit your Web browser when you are finished.

Copyright © 2007 Test Pubcookie Login Server hosted on star11

SWIM Web Portal

Launch and Monitor Current Simulation Runs:

- Demo Code (PPPL)

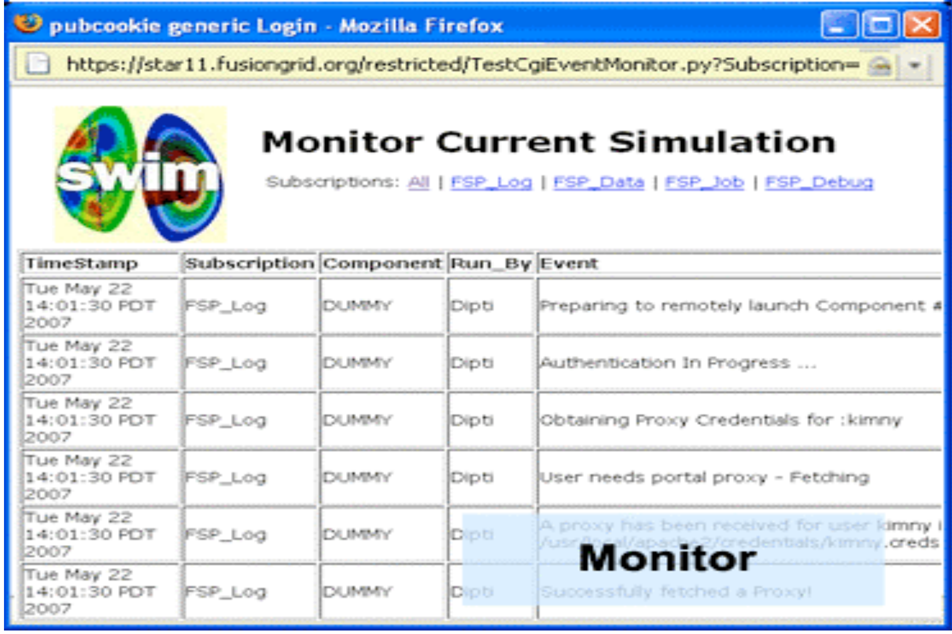
Query for Events Monitored from Previous Simulations Runs:

- Event Monitor

Query for Meta-Data from Simulations Runs made:

- Summary from Simulation Runs

Portal Services

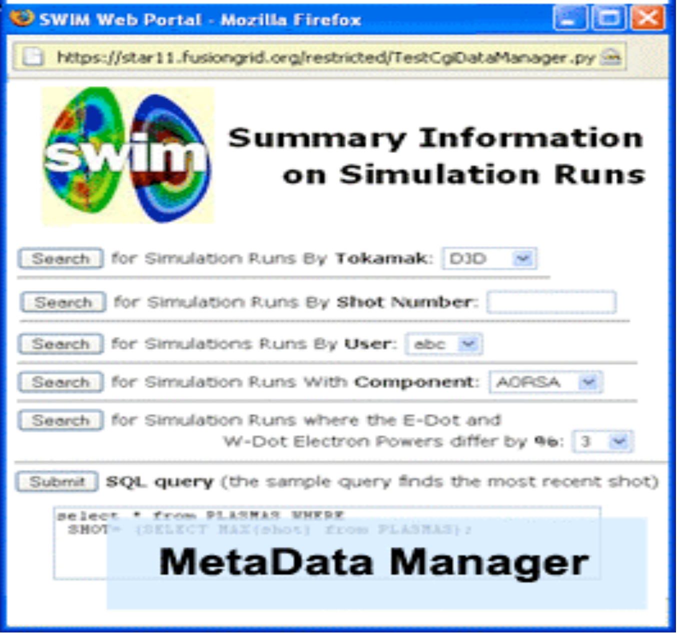


Monitor Current Simulation

Subscriptions: All | FSP_Log | FSP_Data | FSP_Job | FSP_Debug

TimeStamp	Subscription	Component	Run_By	Event
Tue May 22 14:01:30 PDT 2007	FSP_Log	DUMMY	Dipti	Preparing to remotely launch Component #
Tue May 22 14:01:30 PDT 2007	FSP_Log	DUMMY	Dipti	Authentication In Progress ...
Tue May 22 14:01:30 PDT 2007	FSP_Log	DUMMY	Dipti	Obtaining Proxy Credentials for :kimny
Tue May 22 14:01:30 PDT 2007	FSP_Log	DUMMY	Dipti	User needs portal proxy - Fetching
Tue May 22 14:01:30 PDT 2007	FSP_Log	DUMMY	Dipti	A proxy has been received for user kimny i /usr/local/apache2/credentials/kimny.creds
Tue May 22 14:01:30 PDT 2007	FSP_Log	DUMMY	Dipti	Successfully fetched a Proxy!

Monitor



Summary Information on Simulation Runs

Search for Simulation Runs By **Tokamak**: D3D

Search for Simulation Runs By **Shot Number**:

Search for Simulations Runs By **User**: abc

Search for Simulation Runs With **Component**: AORSA

Search for Simulation Runs where the E-Dot and W-Dot Electron Powers differ by **%**: 3

Submit **SQL query** (the sample query finds the most recent shot)

```
select * from PLASMAS WHERE
 SHOT= (SELECT MAX(shot) from PLASMAS);
```

MetaData Manager

Plan for coming year – Portal

- **Proposed security model works on PPPL systems**
- **Meta-data management requires further organization**
  - **Scheme for generating unique run identifiers**
  - **Support internal meta-data queries as simulation components become available**
- **Provide browser-based graphical visualizations of run results with Elviz**
- **Release Portal as a production system that can accept new components as they become available**

Progress on Scientific Goals and Plans for following year

- **Completion of design for first release of Integrated Plasma Simulator and its implementation**
- **Design and implementation of SWIM web portal**
- **Progress on physics studies**

The largest physics analysis task for SWIM is developing an MHD closure that includes RF effects → closures working group

RF effects produce additional contributions to the fluid equations and modify the fluid closure moments

Basic formulation starts from general kinetic equation

$$\frac{df}{dt} = C(f) + Q(f),$$

$$\frac{d}{dt} = \frac{\partial}{\partial t} + \vec{v}\cdot\nabla + \frac{q_s}{m_s}(\vec{E} + \vec{v}\times\vec{B})\frac{\partial}{\partial\vec{v}}$$

$$Q(f) = \frac{\partial}{\partial\vec{v}}\cdot\vec{\vec{D}}\cdot\frac{\partial f}{\partial\vec{v}}$$

RF contributions enter via quasi-linear diffusion operators as functions in 5-D

Taking moments yields

$$\frac{\partial n_s}{\partial t} + \nabla\cdot(n_s\vec{v}_s) = 0,$$

$$m_s n_s\left(\frac{\partial\vec{v}_s}{\partial t} + \vec{v}_s\cdot\nabla\vec{v}_s\right) = n_s q_s(\vec{E} + \vec{v}\times\vec{B}) - \nabla p_s - \nabla\cdot\vec{\vec{\pi}}_s + \vec{R}_s + \vec{F}^{rf}_{s0},$$

$$\frac{3}{2}n_s\left(\frac{\partial T_s}{\partial t} + \vec{v}_s\cdot\nabla T_s\right) + n_s T_s\nabla\cdot\vec{v}_s = -\nabla\cdot\vec{q}_s - \vec{\vec{\pi}}_s : \nabla\vec{v}_s + Q_s + S^{rf}_{s0}$$

Additional terms due to RF

$$\vec{F}^{rf}_{s0} = \int d^3\vec{v}\; m_s\vec{v}Q(f_s),$$

$$S^{rf}_{s0} = \int d^3\vec{v}\;\frac{1}{2}m_s v'^2 Q(f_s)$$

While the fluid equations are exact, a treatment of the closure problem is needed

A procedure to produce the needed closure information is under development - Hegna and Callen, Sherwood '07

- **Assuming RF produces a small kinetic distortion**

$$f_s = f_{Ms} + F_s, \quad F_s << f_{Ms}$$

- **With $f_s = f_{Ms}$, the RF contributions to the fluid equations are determined**

$$\vec{F}_{s0}^{rf} = \int d^3\vec{v}\; m_s \vec{v} Q(f_s) \cong \int d^3\vec{v}\; m_s \vec{v} Q(f_{Ms}) = \vec{F}_{s0}^{rf}(n_s, \vec{v}_s, T_s, \vec{E}_{rf}, \ldots),$$

$$S_{s0}^{rf} = \int d^3\vec{v}\; \frac{1}{2} m_s v'^2 Q(f_s) \cong \int d^3\vec{v}\; \frac{1}{2} m_s v'^2 Q(f_{Ms}) = S_{s0}^{rf}(n_s, \vec{v}_s, T_s, \vec{E}_{rf}, \ldots),$$

- **However, we're not done yet --- the RF terms modify the closures**

Using a Chapman-Enskog-like ansatz, an equation for the kinetic distortion can be derived

Using a Chapman-Enskog-like approach, an equation for the kinetic distortion is derived with RF source terms

- **The kinetic equation is given by**

$$\frac{dF_s}{dt} - C(F_s) = Q(f_{Ms}) + \vec{v}'\cdot[\nabla\cdot\vec{\vec{\pi}}_s - \vec{R}_s - \vec{F}^{rf}_{s0}]\frac{f_{Ms}}{n_s T_s} + (\frac{m_s v'^2}{3T_s} - 1)[\vec{\vec{\pi}}_s : \nabla\vec{v}_s + \nabla\cdot\vec{q}_s - Q_s - S^{rf}_{s0}]\frac{f_{Ms}}{n_s T_s}$$

$$-(\frac{m_s v'^2}{2T_s} - \frac{5}{2})\vec{v}'\cdot\nabla T_s \frac{f_{Ms}}{T_s} + \frac{m_s}{T_s}[\vec{v}'\vec{v}' - \frac{v'^2}{3}\vec{\vec{I}}]:\nabla\vec{v}_s f_{Ms}$$

Additional source terms for the kinetic distortion due to RF

- **Simple applications have been worked out**

  e. g., the Spitzer problem with RF

$$R_{\|} = \frac{m_e \nu_e}{n_e e^2 \Lambda_{00}} J_{\|} + \sum_{k=1} \frac{\Lambda_{0k}}{\Lambda_{00}} \frac{F^{rf}_{ke}}{n_e e}, \quad F^{rf}_{je} = \int d^3\vec{v}\ \vec{v}' L^{3/2}_j(x) Q(f_{Me})$$

- **For applications to magnetic islands, the kinetic equation takes on a form similar to that used by Held et al to calculate closures**
  - **Same operator on kinetic distortion F$_s$, the RF bits enter as additional sources**

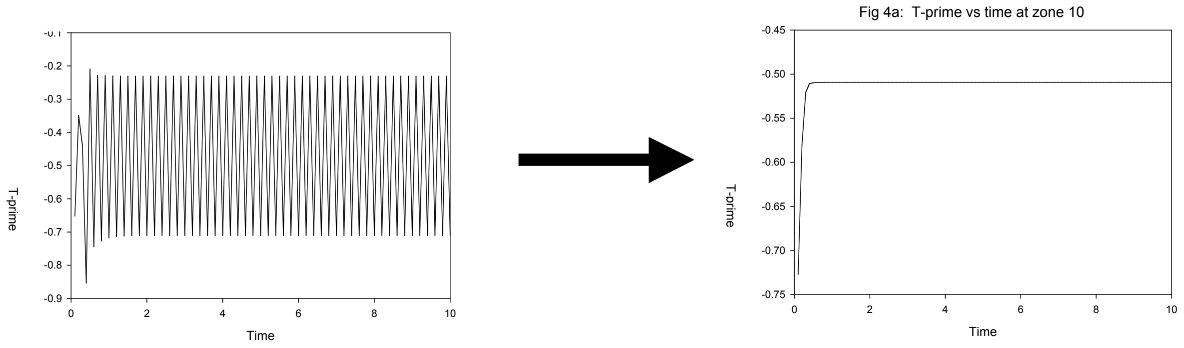
$$v_{\|}\nabla_{\|}F - C(F_s) \cong Q(f_{Ms}) - \vec{v}'\cdot[\vec{R}_s + \vec{F}^{rf}_{s0}]\frac{f_{Ms}}{n_s T_s} + \ldots$$

- **Procedures for inverting this operator have been developed, Held *et al* '03**

Numerous improvements to SWIM physics models and codes in collaboration with other projects and base theory

PTRANSP project

- New Newton method to deal with numerically stiff transport model such as GLF23 and MMM95



- Porcelli module for triggering sawtooth crashes
- Porcelli model for partial magnetic reconnection

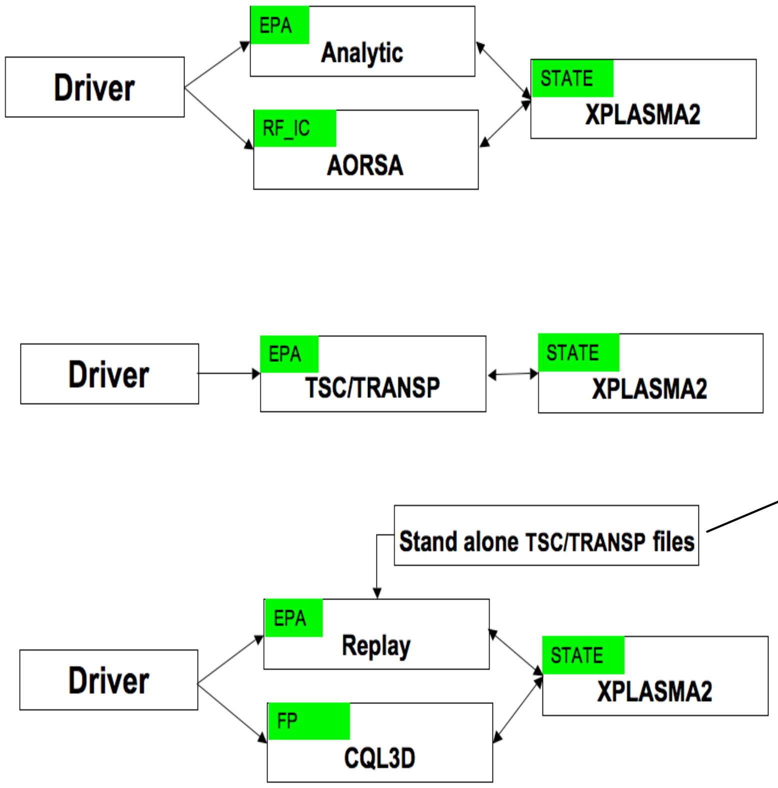
SciDAC CEMM

- Shared development of M3D and NIMROD as required for sawtooth and NTM calculations

SciDAC CSWPI

- Improvements to AORSA, TORIC, CQL3D, ORBIT-RF driven by or supported by SWIM

Physics runs with IPS are beginning



Driver

EPA Analytic

RF_IC AORSA

STATE XPLASMA2

Driver

EPA TSC/TRANSP

STATE XPLASMA2

Stand alone TSC/TRANSP files

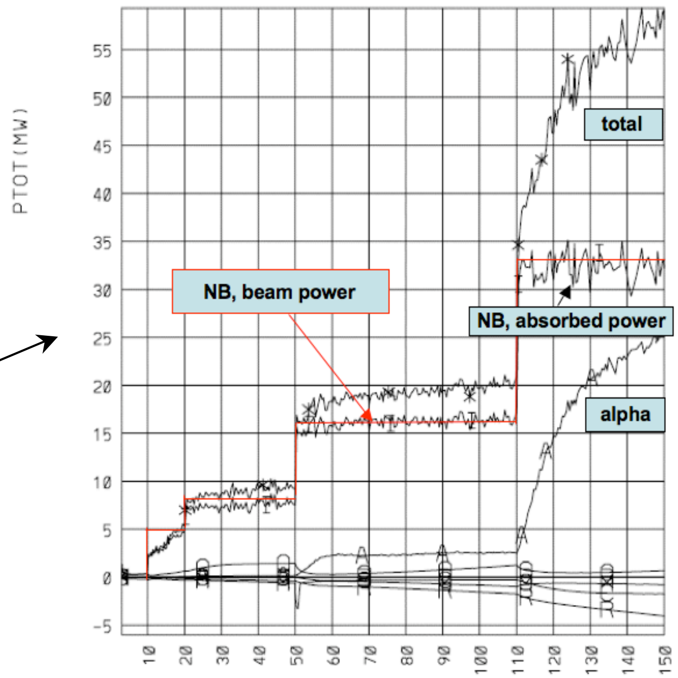
Driver

EPA Replay

FP CQL3D

STATE XPLASMA2

Integrated power deposition in an off-axis, beam heated ITER discharge simulation of the first 150 s.



- Analysis of time slices using AORSA
- Time slice analysis of fast electron generation using CQL3D
- AORSA/TORIC benchmarking

Plans for coming year – physics

Applications of IPS

- **Validation studies**
  - **AORSA/TORIC + CQL3D minority heating and rate of tail formation, C-mod comparisons**
- **Time slice simulations**
  - **AORSA comparisons for ITER scenario studies run with TORIC**
  - **Stability analysis of ITER scenario runs**
  - **Toroidal Alfven Eigenmode studies on C-mod using M3D/NOVA-K**
- **Dynamic simulations**
  - **RF effects on sawtooth oscillations – DIII-D or JET**
  - **Runaway electron production in ITER startup/shutdown – have begun this study**

Physics analysis and development

- **Slow MHD strategy (new post doc coming on board – U. Wisc.)**
  - **Restart computational tearing mode work and consider RF effects on classical tearing modes**
  - **Develop a phenomenological evolution equation for Jrf in MHD codes, using numerical RF sources**
  - **Derive a more rigorous closure model including RF**

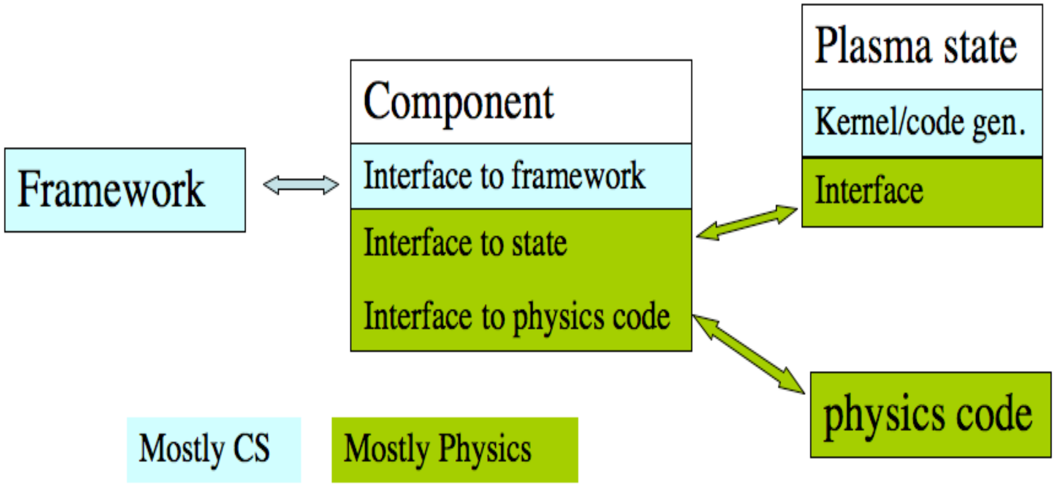
Role of collaborations

- **SciDAC – SWIM is built on collaborations with other projects**
  - Center for Extended MHD modeling (CEMM) ↔ Linear and non-linear MHD
  - Center for Simulation of Wave-Plasma Interaction (CSWPI) ↔ RF and Fokker Planck
  - Predictive TRANSP (PTRANSP) ↔ Plasma State, TRANSP
  - Look forward to working with new SciDAC projects ↔ energetic particle, turbulent transport
- **OASCR Centers for Enabling Technology (CET)**
  - Toward Optimal Petascale Simulations (TOPS) ↔ advanced solvers, application of PETc to NIMROD
  - Center for Technology for Advanced Scientific Component Software (TASCS) ↔ component architcture, CCA
  - Center for Interoperable Technologies for Advanced Petascale Simulations (ITAPS) ↔ mesh transfer
  - Visualization and Analytics Center for Enabling Technologies (VACET) ↔ visualization, analysis of magnetic islands
- **Students – Columbia, Princeton Univ, IU, U. Wisc ↔ ORNL, TechX, LLNL, PPPL**
- **International collaborations**
  - International Tokamak Physics Activitiy (ITPA) ↔ SWIM physics/ CS at Integrated Modeling a Global Effort (IMAGE, ITPA subgroup)
  - US Japan Workshop on Integrated Modeling of Fusion Plasmas ↔ SWIM and US CS/Math participation

Internal Collaborations

We spent a lot of time defining functionality and specifying interfaces

We have 4 groups of developers – framework, component, plasma state, physics code



Allows people to focus on what they do best

- **Project user/advisory committee – M. Greenwald (MIT), C. Kessel (PPPL), M. Murakami (ORNL/DIII-D), A. Siegel (ANL), A. Sussman (U. MD)**
- **Communication – Project meetings, web site, conference calls, SVN repository**

Role of leadership class computing

INCITE project Simulation of Wave Interactions and MHD

- Early years mostly CEMM and CSWPI → SWIM later years
- Renewed in 2007
- On track to use our 2007 allocation

Major codes ported to Jaguar

- M3D, NIMROD, AORSA, CQL3D, ORBIT-RF, Plasma State/ XPLASMA2
- Have had issues with availability, stability, utilities (like the compiler)

Porting to PPPL SGI cluster

- A gateway for development and testing SWIM invested in 16 SGI processors on PPPL cluster
- All major codes also ported to PPPL cluster

Directed deliverables for next year

- **Public release of Integrated Plasma Simulator**
- **Transition of IPS to Jaguar**
- **Slow MHD – effect of RF on classical tearing mode studied with NIMROD**
- **Initial sawtooth simulation with IPS – Demonstrate transition from 2D (axisymmetric) equilibrium to 3D nonlinear MHD code (NIMROD and/or MHD), compute sawtooth event, and transition back to axisymmetric equilibrium when mode re-symmetrizes**