

A Coupled ECCD/MHD model using NIMROD, GENRAY, and the Integrated Plasma Simulator



THE UNIVERSITY
of
WISCONSIN
MADISON

Thomas G. Jenkins, *University of Wisconsin-Madison*
in collaboration with

Dalton Schnack, Carl Sovinec, Chris Hegna, Jim Callen
University of Wisconsin-Madison

Scott Kruger
Tech-X Corporation

Eric Held
Utah State University

Wael Elwasif
ORNL

Bob Harvey, Alexander Smirnov
CompX

The SWIM project team
<http://cswim.org>



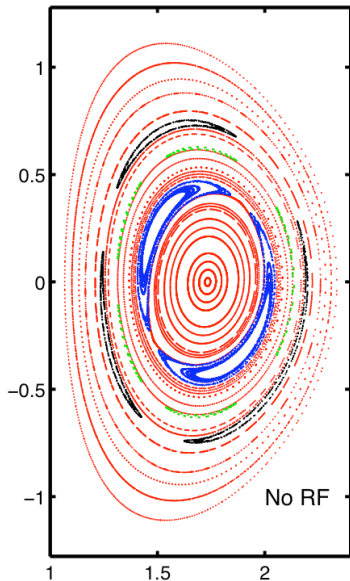
CEMM meeting
1 November 2009 Atlanta, GA

Outline

- One-page summary - the coupled RF/MHD problem of interest
- The Integrated Plasma Simulator – what it is and how it works
 - The NIMROD-IPS interface – how to plug a code into the IPS
 - Some issues with fully coupled RF/MHD simulations, and a status report
- Recent results using ad hoc model for toroidally localized RF

The problem: mitigation and control of tearing modes (magnetic islands) by electron cyclotron current drive (ECCD)

Magnetic islands, t = 0.12815 s



- Neoclassical (or resistive) tearing modes generate magnetic islands in tokamaks. Experimentally, RF waves resonant with electron cyclotron motion can drive currents that alter the island structure. We want to model this process.

- RF effects enter the kinetic equation as a quasilinear operator

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{v} \cdot \nabla f_\alpha + \frac{q_\alpha}{m_\alpha} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \frac{\partial f_\alpha}{\partial \mathbf{v}} = C(f_\alpha) + Q(f_\alpha)$$

- Velocity moments of $Q(f_\alpha)$ modify the MHD equations, and appear in closure calculations for \mathbf{q} and Π . We want to get $Q(f_\alpha)$ from an RF code (not from NIMROD).

$$\nabla \cdot \mathbf{B} = 0$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

$$\mathbf{E} + \mathbf{u} \times \mathbf{B} = \eta \mathbf{J} + \frac{\mathbf{F}_{e0}^{rf}}{n|q_e|}$$

$$S_{\alpha 0}^{rf} \equiv \int \frac{1}{2} m_\alpha v^2 Q(f_\alpha) d\mathbf{v}$$

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \mathbf{J} \times \mathbf{B} - \nabla \cdot \Pi + \sum_\alpha \mathbf{F}_{\alpha 0}^{rf}$$

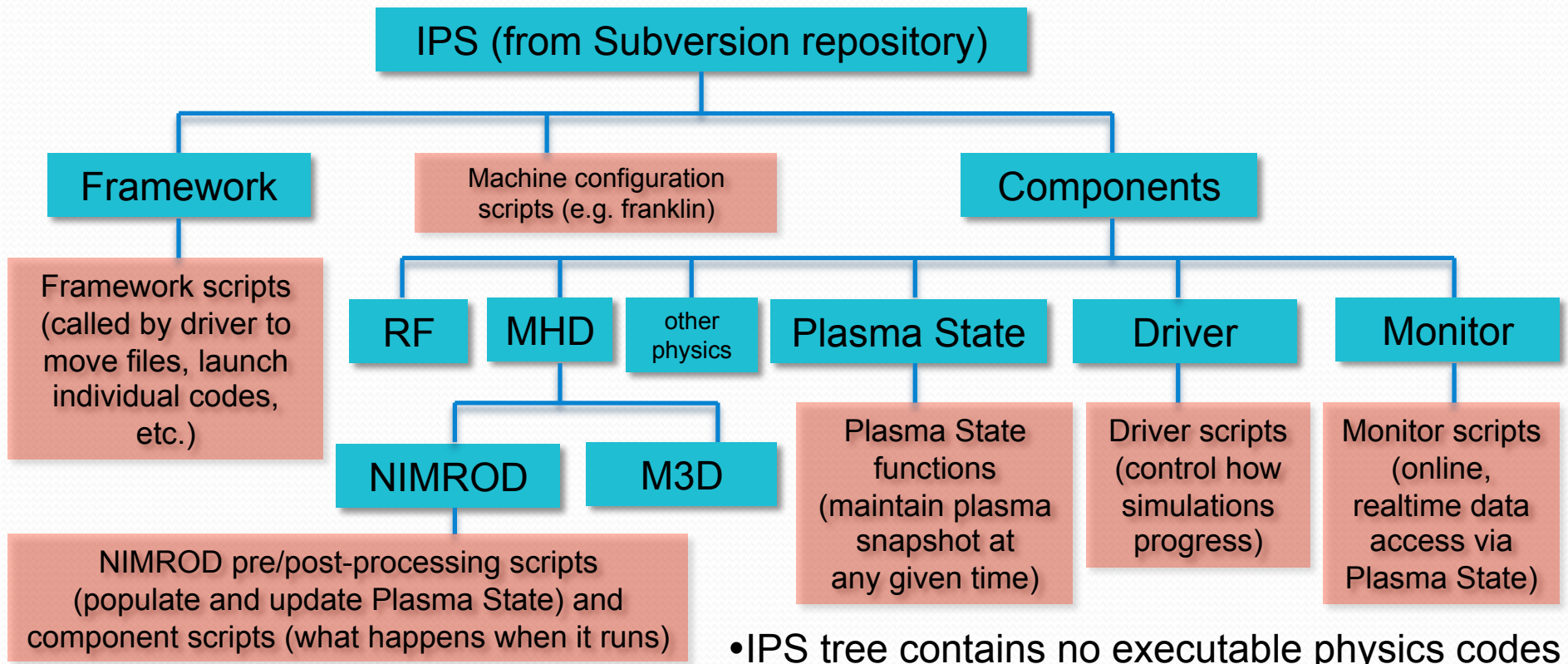
$$\mathbf{F}_{\alpha 0}^{rf} \equiv \int m_\alpha \mathbf{v} Q(f_\alpha) d\mathbf{v}$$

$$\frac{3}{2} n \left(\frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T \right) + p \nabla \cdot \mathbf{u} = -\nabla \cdot \mathbf{q} - \Pi : \nabla \mathbf{u} + Q + \sum_\alpha S_{\alpha 0}^{rf}$$

- Use the Integrated Plasma Simulator (IPS) to do this.

What is the Integrated Plasma Simulator (IPS)?

- "Management" software – a way to run multiple physics codes concurrently or sequentially, using consistent input parameters
- A way to couple different codes that model various physical processes, so we can study complicated plasma behavior without extensive expansion of any one code
- At some level, just a collection of Python scripts that interact in interesting ways

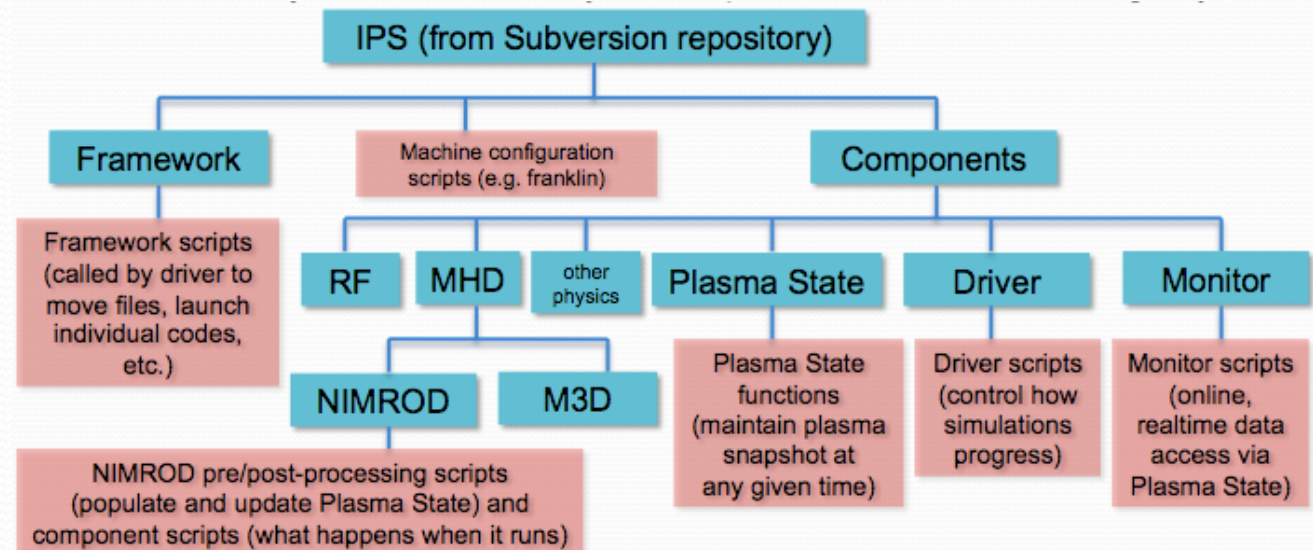


The IPS moves data in and out of the “Plasma State”

- Plasma State - a collection of (exhaustively specified)
 - ◆ data about the plasma and plasma controls, e.g.
 - Equilibrium pressure profile $p(\psi)$ (1D arrays)
 - Spatial location/orientation of the ECCD launcher (scalars)
 - Magnetic fields (multidimensional arrays)
 - (alternatively) files describing magnetic fields (formatted files)
 - ◆ metadata about which IPS components generated the data, how it is described, and which components may update it
 - Kind of like a code repository – file [object] X was last altered by user [component] Y at time [simulation time] T...
 - ◆ metadata about itself (version number and contents)
 - As different physical variables are needed for different physics problems, the Plasma State can be expanded or altered.
- Detailed descriptions of Plasma State data can be found at
http://w3.pppl.gov/~dmccune/SWIM/plasma_state_spec.dat

What does an IPS user have to do?

- Pick a physics problem – e.g. finding the self-consistent response of resistive tearing modes to electron cyclotron current drive (ECCD)
- Decide which components are necessary to model the problem (MHD, RF, driver, monitor). Choose physics codes for the physics components (NIMROD=MHD, GENRAY=RF – other choices are possible) and get executables for them.
- Configure the component scripts, including driver and monitor components, to control how the physics codes will interact and what they'll show you as they do it.
- Set up scripts for pre- and post-processing of data from physics codes (to interface with the Plasma State). Goal is *minimal IPS-related modification* of actual physics codes (not always practical, though; see next page).
- Run the simulation and watch the output.



For ECCD/MHD problem, NIMROD/IPS interactions are a bit different

- Initial IPS philosophy – **don't make users "change" their codes**. Instead,
 - get input data for a given physics code from the Plasma State (preprocessing it if necessary) via a script,
 - run the code as-is,
 - put output data from the code back into the Plasma State (postprocessing it if necessary) via another script.

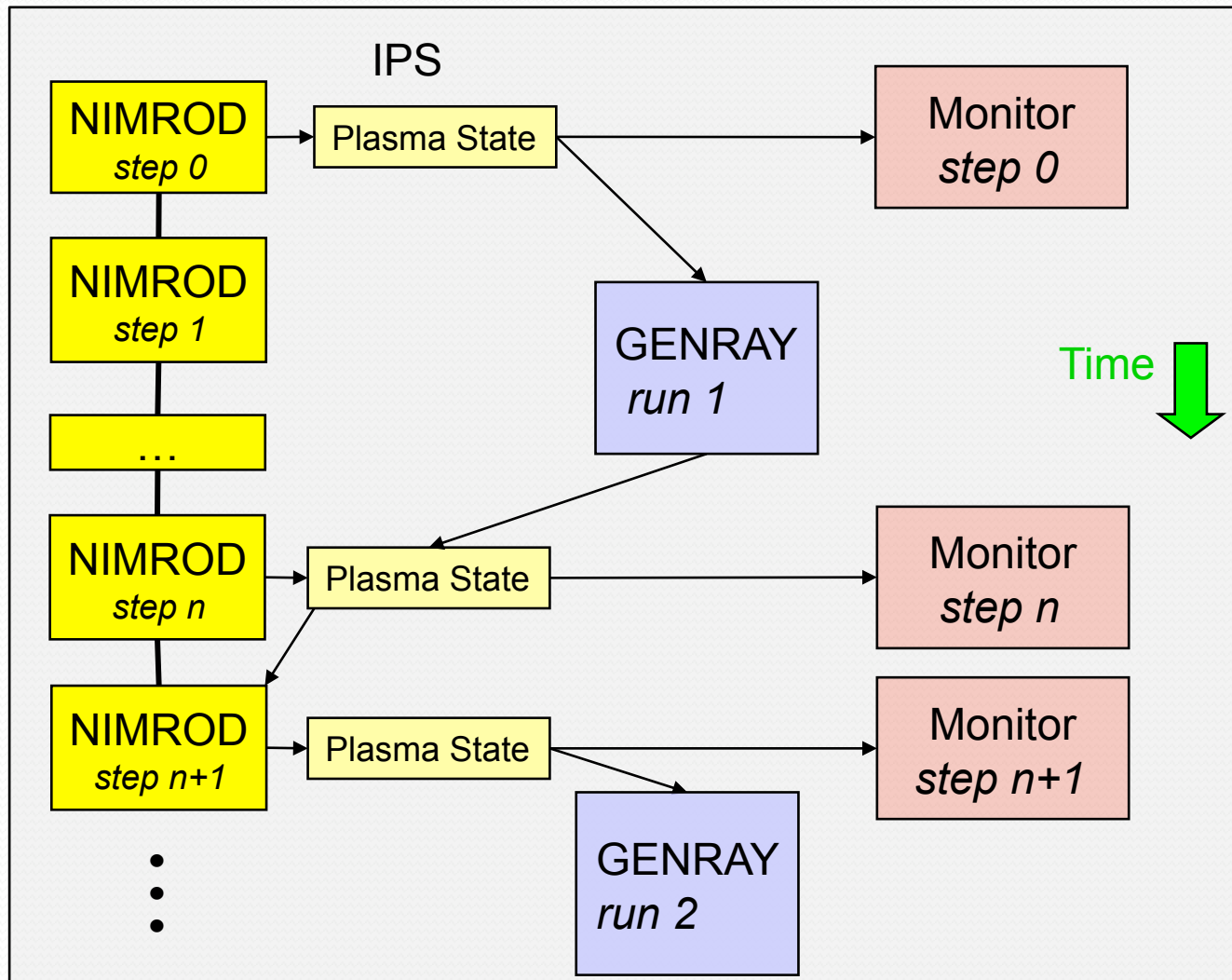
• Not ideal for the RF/MHD problem; we don't want/need to start and stop NIMROD repeatedly (RF changes very slowly in time). Interfacing NIMROD with the Plasma State directly allows us to run NIMROD constantly and get updates from GENRAY periodically.

• Likewise, GENRAY data is postprocessed inside NIMROD, (the relevant output is a netCDF file).

• IFDEF logic enables the Plasma State interface when needed.

```
SUBROUTINE make_eqdsk
  USE bicube
  USE struct_mod
#ifdef HAVE_PS
  USE plasma_state_mod
#endif
  IMPLICIT NONE
  TYPE(bicube_type) :: bcrz
  TYPE(spline_type) :: intgd
  TYPE(spline_type) :: profile
  INTEGER(i4) :: pmx, pmy, nrgrid, nzgrid, ir, iz, nw, nh, npsi
  INTEGER(i4) :: ix, iy, ip, it, mthetaft, mpx, mpy, npf, iter, i, j
```

Schematic NIMROD/GENRAY coupling under IPS



- Island geometry only weakly modifies magnitude of $|\mathbf{B}|$
- Resonance condition of GENRAY changes weakly
- If greater sensitivity discovered, can move to tighter coupling
- What we need for this to work:
 - NIMROD/GENRAY component scripts
 - NIMROD/GENRAY scripts for Plasma State interface (prev. page)
 - Driver/monitor scripts

The IPS maintains “event channels” to enable interaction between components

- Every physics component has an “event channel”, i.e. a place where it can broadcast a record of what it’s doing to other components
- Components can subscribe to “event channels” of other components in their component scripts; e.g.
 subscribe(NIMROD, PHYS_GENRAY_EVENTS)
tells NIMROD to periodically look at what GENRAY is broadcasting about itself.
- The component scripts can use this information to determine the workflow, e.g. the NIMROD component script can do something like this:

```
IF <PHYS_GENRAY_EVENTS = “new GENRAY output”> THEN
  Read the new GENRAY output
  Use it to calculate the new local current drive in NIMROD
  Write a dumpfile and postprocess it so GENRAY can read it
  Publish “new NIMROD output” to the PHYS_NIMROD_EVENTS channel
ELSE
  Keep the code running while we wait for GENRAY output
END IF
```

Physics code component scripts – straightforward in principle

- Component scripts use only three operations: INIT, STEP, FINALIZE.

NIMROD (mhd_nimrod.py)

GENRAY (genray_xmhd.py)

<ul style="list-style-type: none">•Finds input files, and runs nimset.•Postprocesses dumpfile (nimset –regrid). Sends file & profile quantities to Plasma State. Publishes to NIMROD channel.•Subscribes to GENRAY channel.	<p>INIT</p> <ul style="list-style-type: none">•Subscribes to NIMROD channel.
<ul style="list-style-type: none">•Runs NIMROD in usual way.•When GENRAY channel says a new RF source file is available:<ul style="list-style-type: none">-Calculates new RF source terms-Writes and postprocesses new dumpfiles for GENRAY, and sends to Plasma State.-Publishes to NIMROD channel.	<p>STEP</p> <ul style="list-style-type: none">•When NIMROD channel says a new dumpfile from NIMROD is available:<ul style="list-style-type: none">-Reads Plasma State profiles to generate GENRAY input-Runs GENRAY & postprocesses-Sends new RF source file to Plasma State-Publishes to GENRAY channel.
<ul style="list-style-type: none">•Does nothing (when NIMROD stops running, we're done)	<p>FINALIZE</p> <ul style="list-style-type: none">•Does nothing (NIMROD won't call it again)

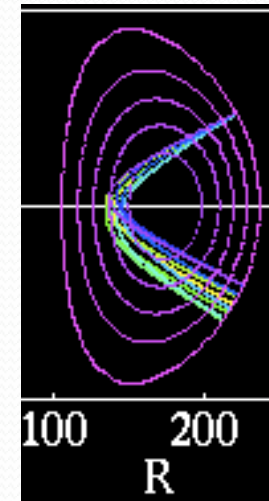
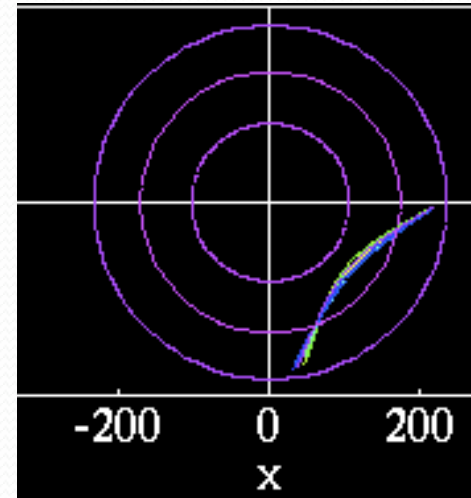
Driver script for the ECCD/MHD problem is easy

- Driver script – because we don't ever stop NIMROD, and there are only two physics components, it's very simple:

- call NIMROD INIT function
- call GENRAY INIT function
- call NIMROD STEP function (runs entire MHD simulation while perpetually checking for GENRAY output)
- call GENRAY STEP function (perpetually waits for NIMROD output, runs, returns data)

Issues for coupled simulations

-Wave accessibility (once we have NIMROD data going to GENRAY, need to find RF parameters that yield RF deposition at the desired point – using GENRAY diagnostic capabilities to figure this out)



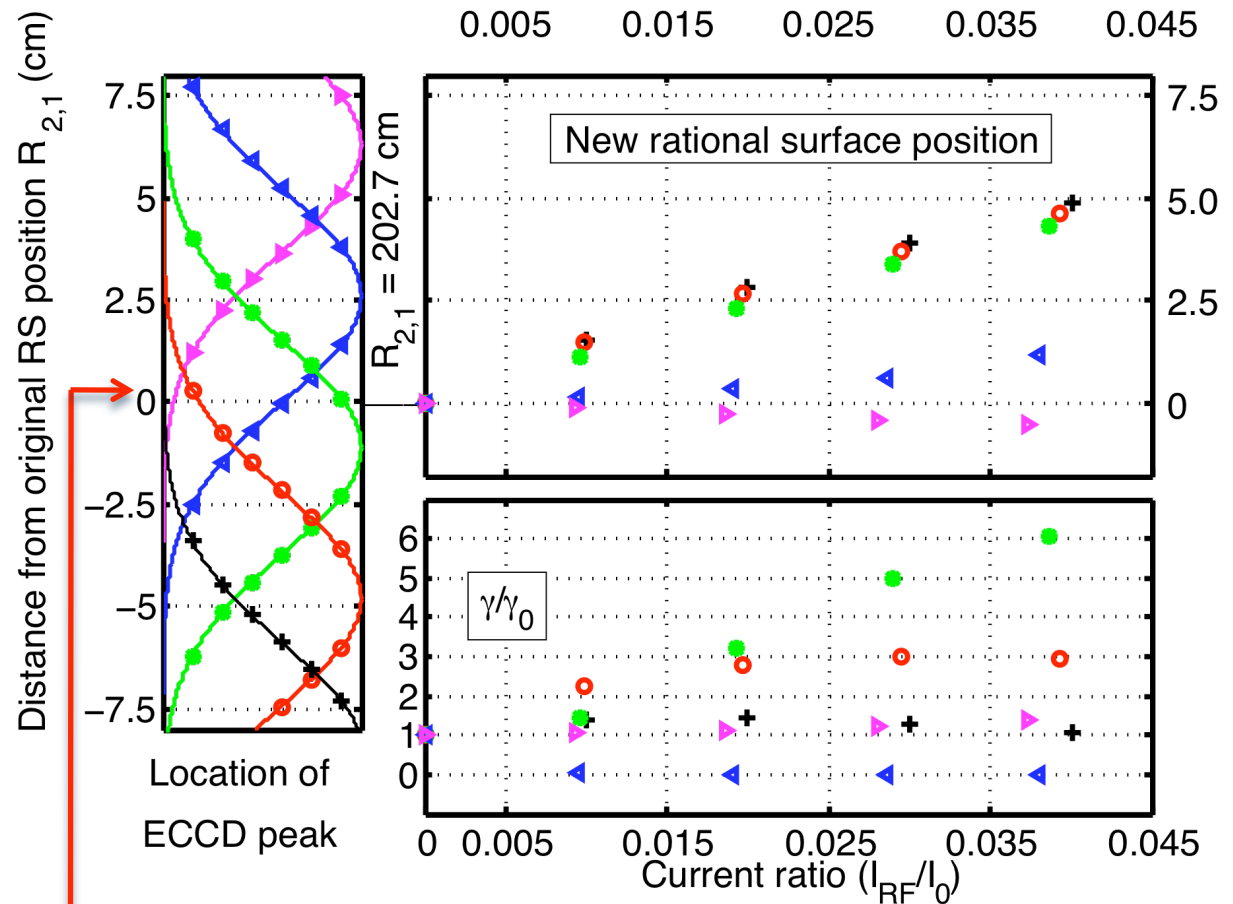
-Time dependence of RF – how do we turn it on; how do we optimally phase it to control island behavior?

-Self-consistency – so far, only have leading-order interaction term (Ohm's law) working. Correction terms in momentum and energy equation coming soon (from transfer of GENRAY's quasilinear diffusion tensor to NIMROD via Plasma State). Closures also need to be calculated self-consistently.

-Neoclassical (as opposed to resistive) tearing modes – need to get plasma equilibria that are sufficiently near marginal stability and have high β .

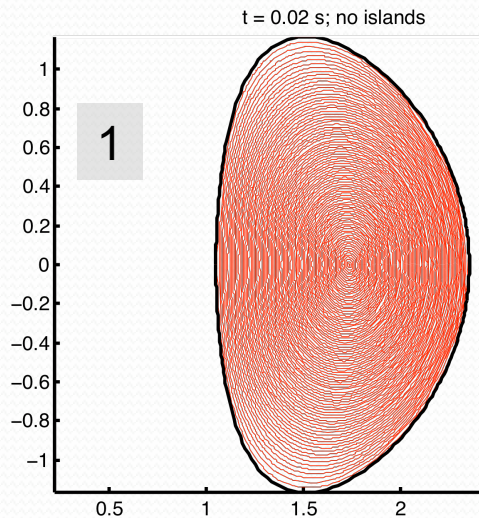
Results from a toroidally localized RF model

- RF deposition model (Gaussian in poloidal plane, and toroidally symmetric) yields RF-induced rational surface motion and alters tearing mode growth rates.
- Improve model – use toroidally localized RF. Deposit near existing islands (can't find the RF-modified equilibria first, since tearing mode growth is also going on at the same time).

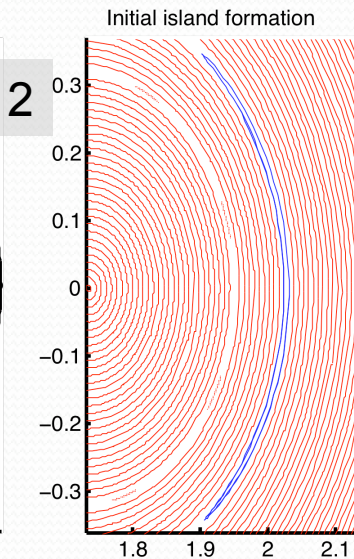
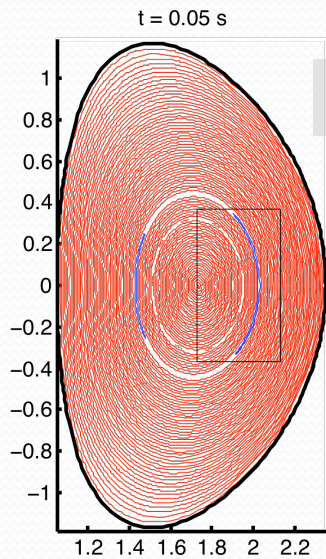
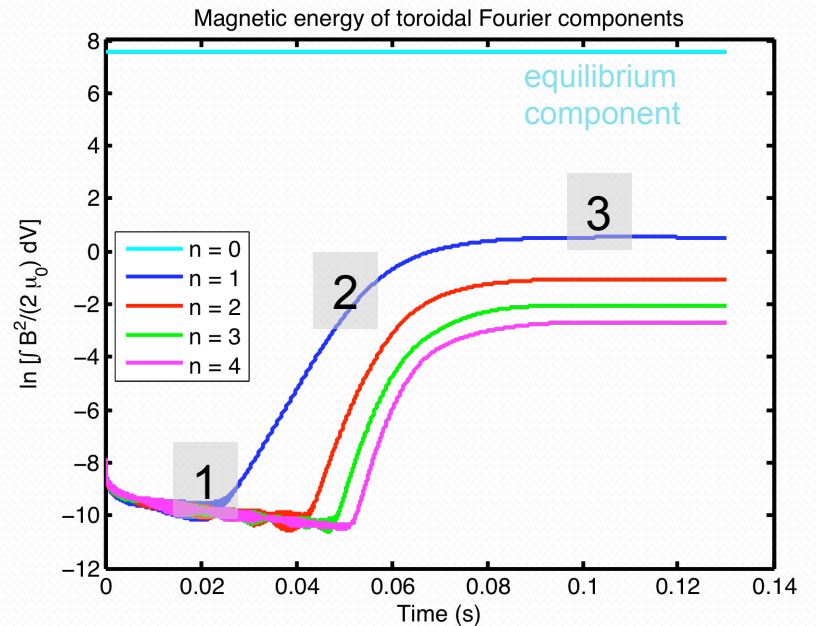


“RF deposition profiles on outboard midplane” relative to “initial rational surface position on outboard midplane” (the zero point).

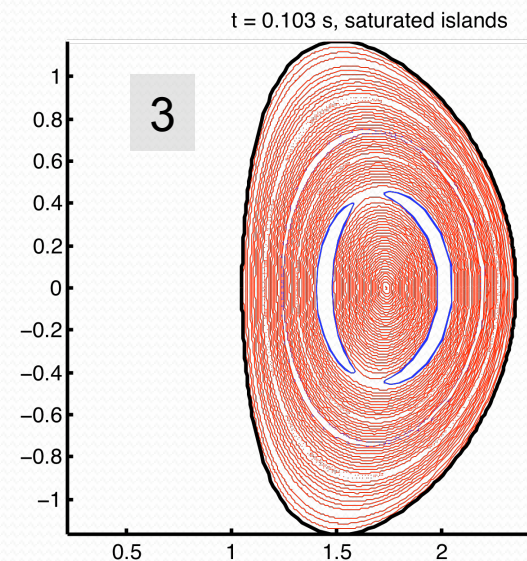
What happens without RF? (Where is the island?)



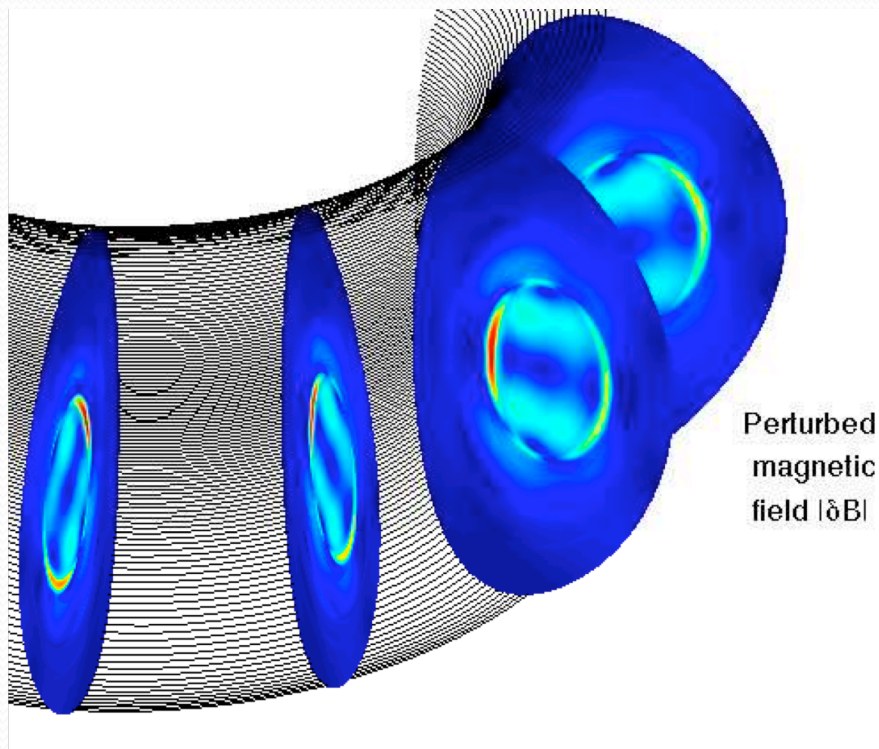
• In NIMROD's toroidal Fourier decomposition, components of the magnetic energy can be associated with mode growth.



• Find the toroidal angle where the O-points of saturated islands cross the outboard midplane; plot islands.

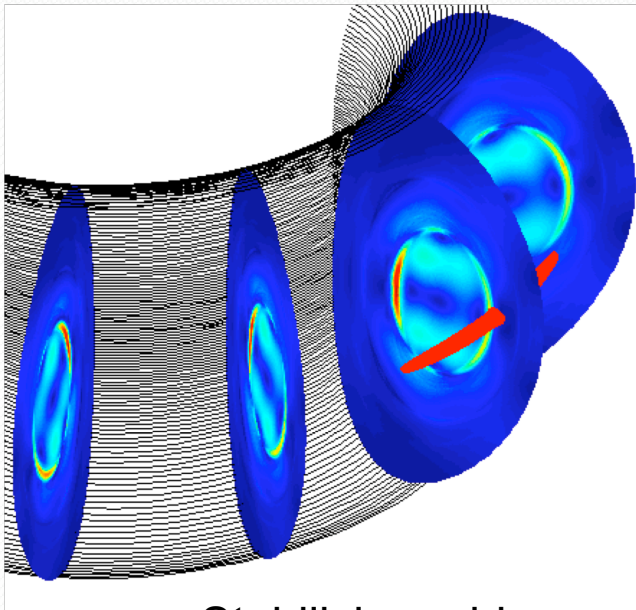


Where should we put the RF?



- In the absence of RF, we know where the island is as a function of time. (No toroidal rotation in this case.)
- However, we expect RF-induced rational surface motion (as in the toroidally symmetric case) will complicate matters.
- Experimental solution – sweep across the island. Numerically, dynamics associated with a moving source may make it harder to see what's going on. Try several stationary sources instead.

Localized deposition near O-point should be stabilizing



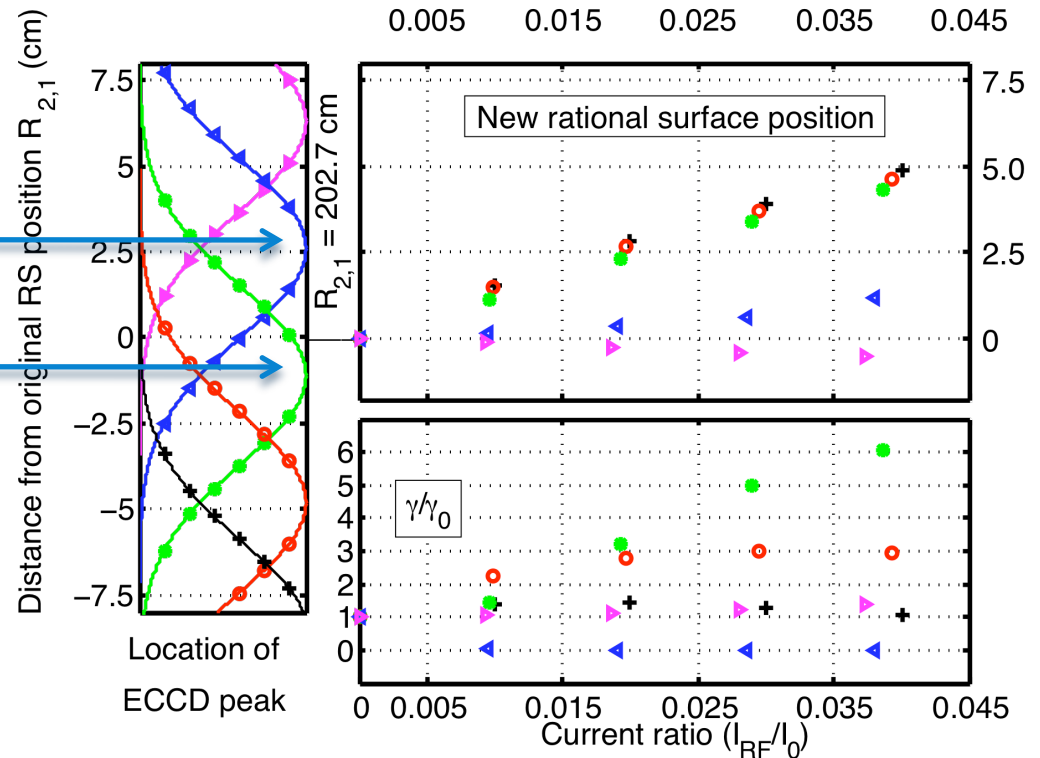
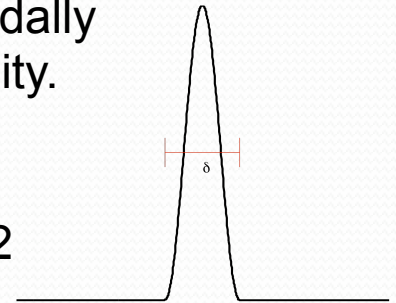
Stabilizing – blue points on lower plot
 Destabilizing – green points on lower plot

•Initially, put RF somewhere in between.

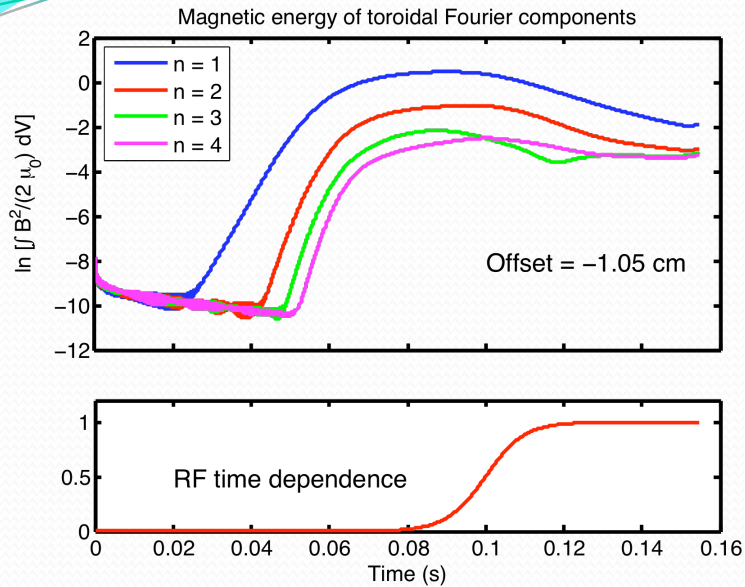
•Toroidal localization yields current in island O-point, but not X-point (as in toroidally symmetric case) – increased stability.

•Toroidal form of RF:

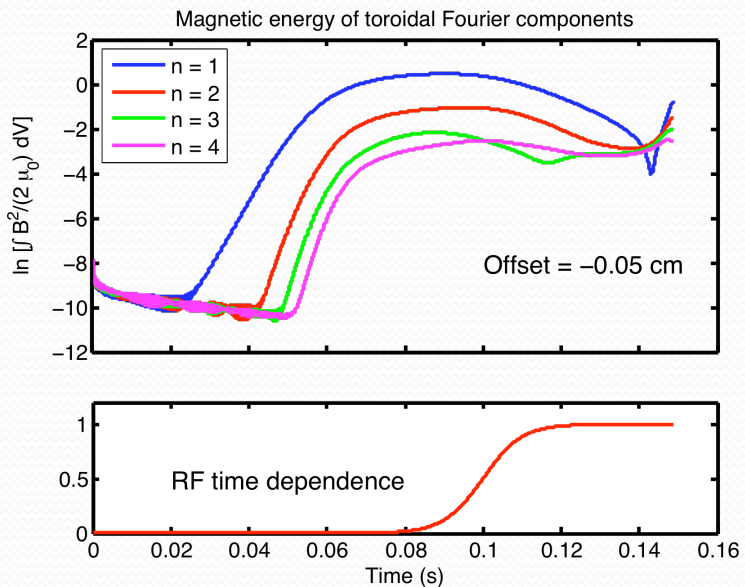
$$f(\zeta) = \cos^2[(\zeta - \zeta_0)\pi/\delta] \text{ if } |\zeta - \zeta_0| < \delta/2 \text{ and zero otherwise}$$



Localized deposition is only temporarily stabilizing

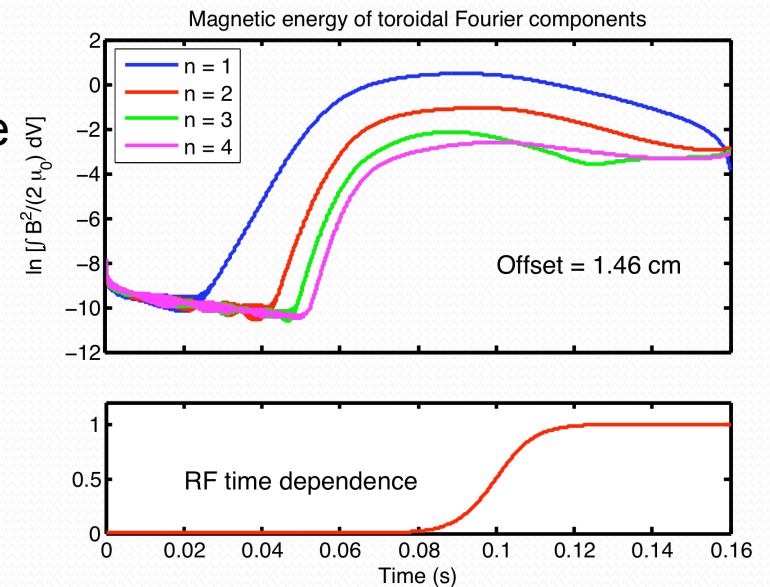
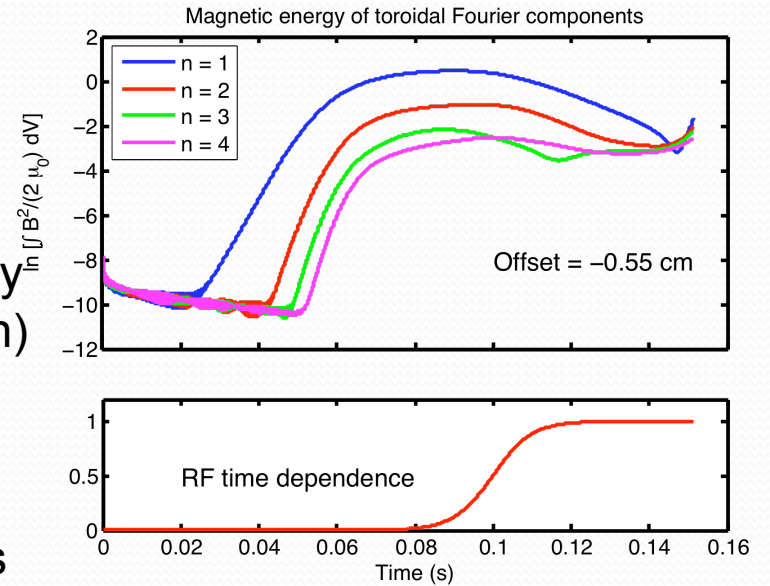


- RF deposition near the island initially shrinks it, but then the magnetic energy (i.e. island width) abruptly goes back up.



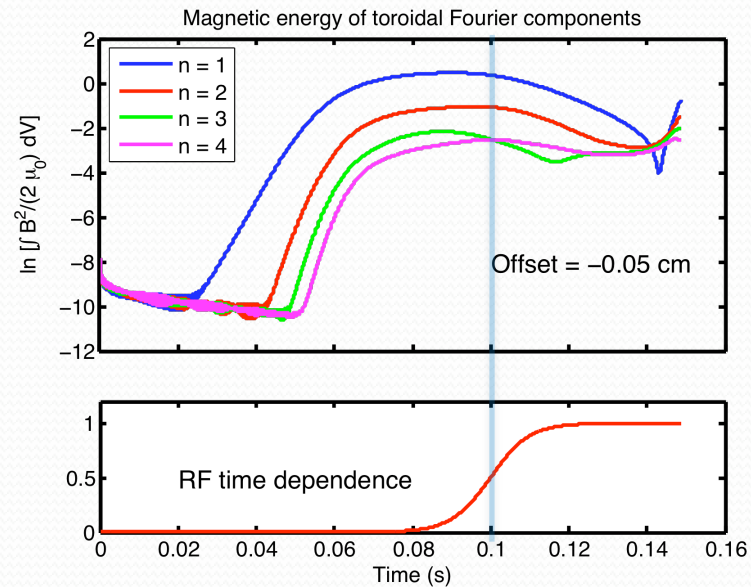
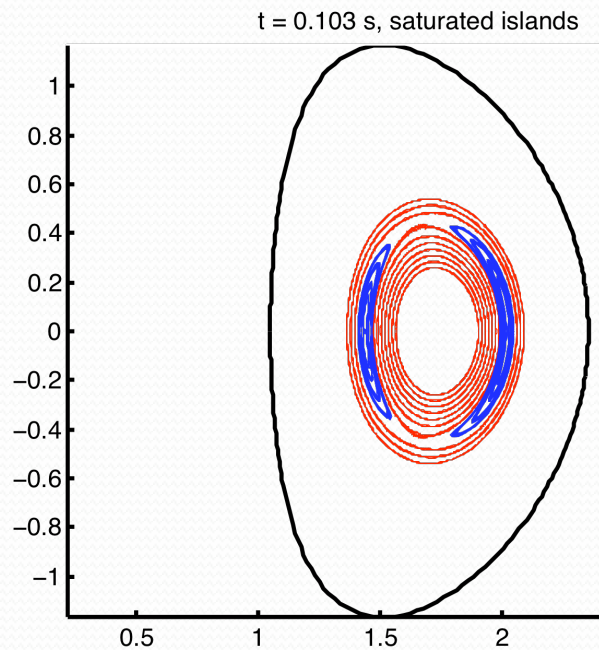
- Smaller offsets from original rational surface position) reduce island width more quickly.

- Look at field line traces to see what's going on...

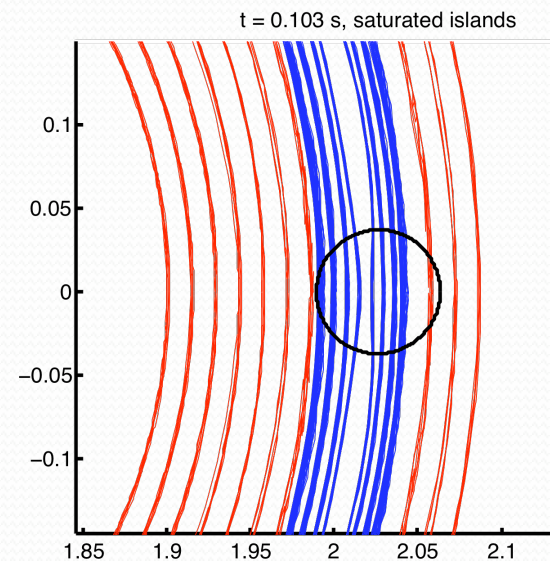


The RF alters the interior island structure

- As RF turns on, field lines inside the island are pinched together.

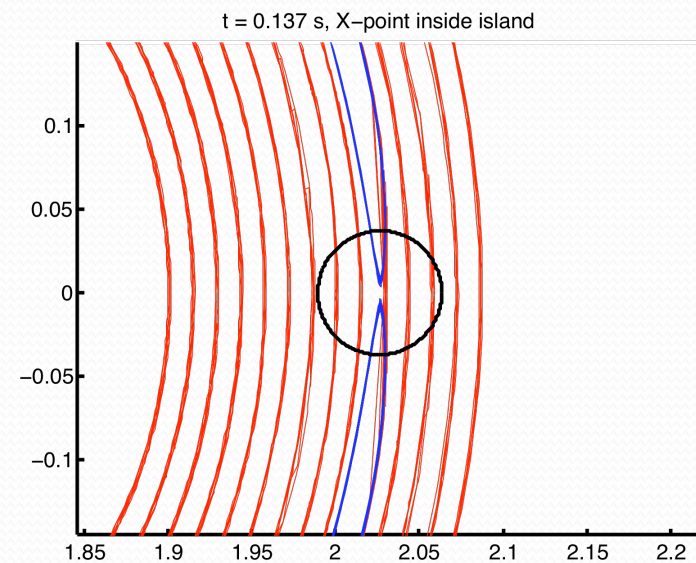
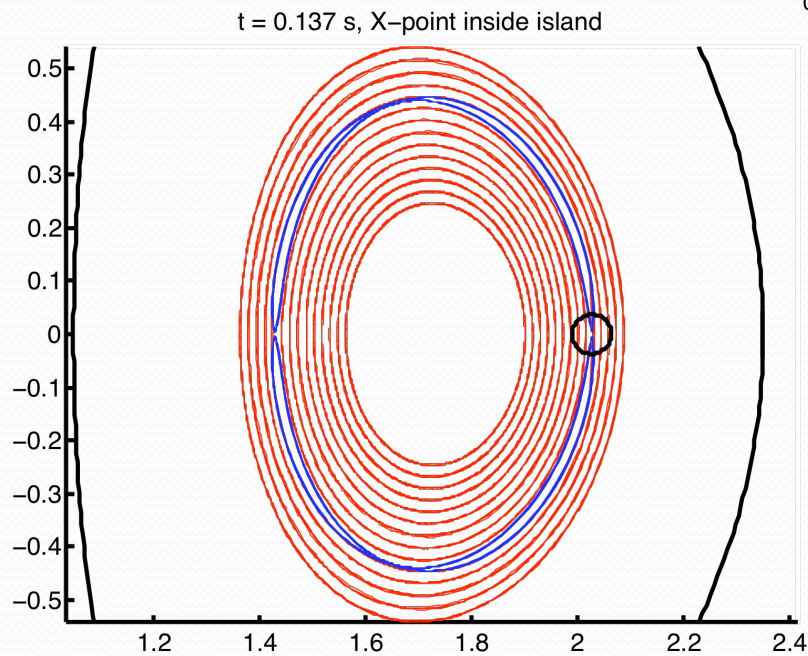
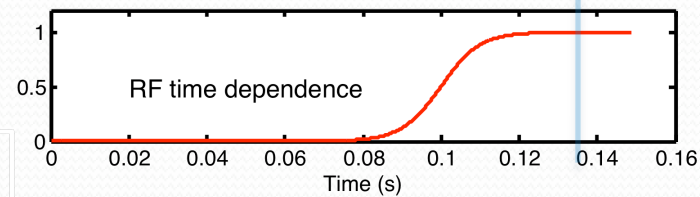
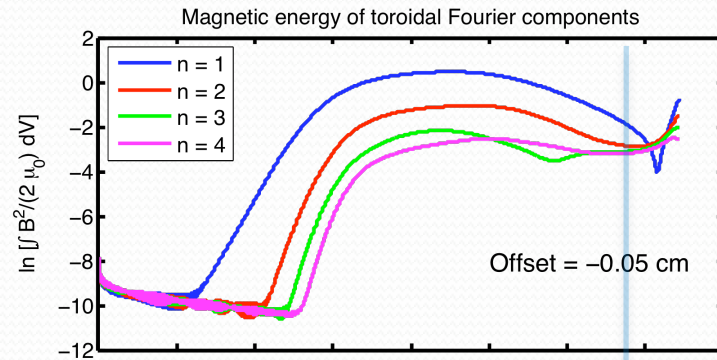


- Take a closer look...



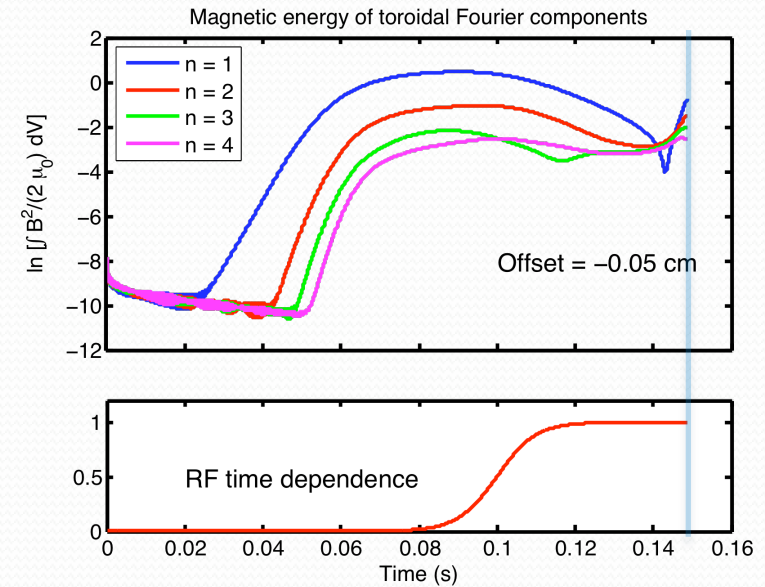
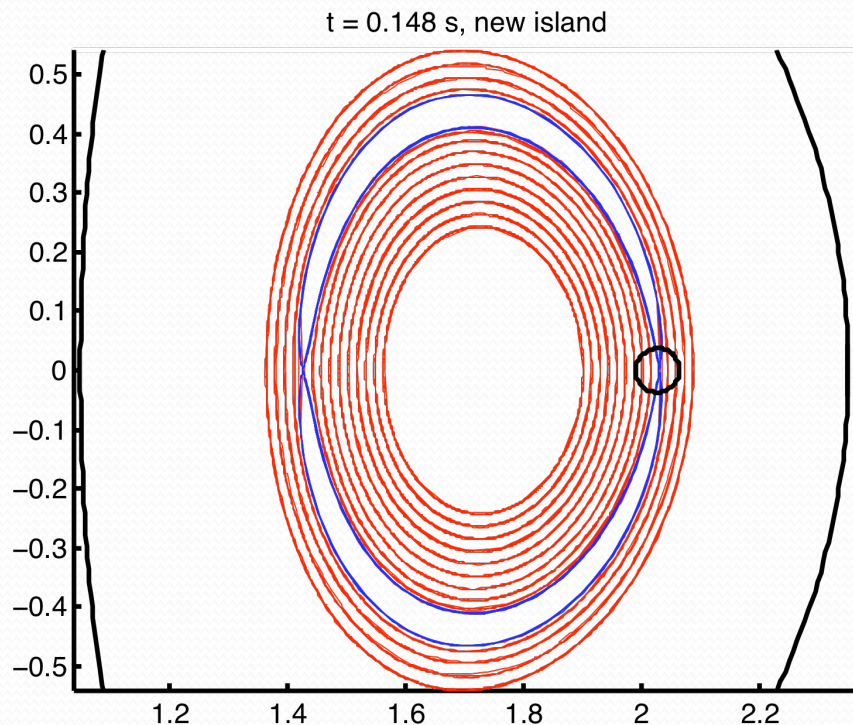
X-points form inside the island in response to RF

- X-point from original island persists as island shrinks in response to RF. (2,1) island becomes (4,2) island (shrinking).



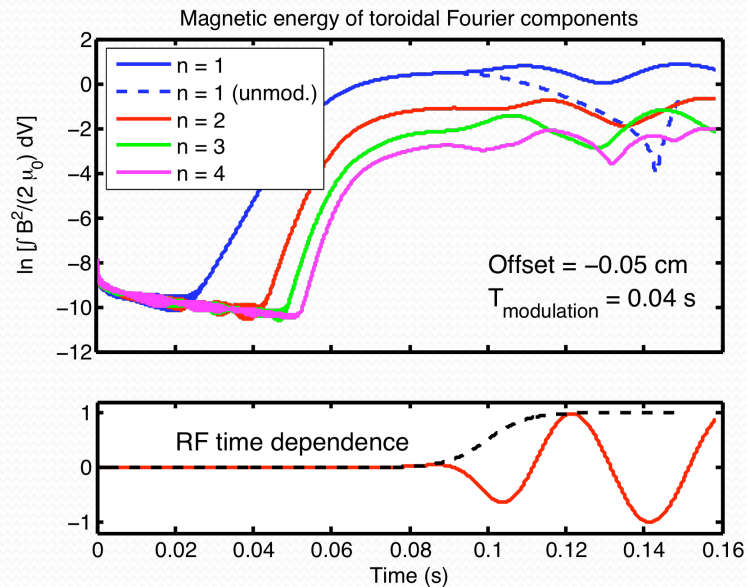
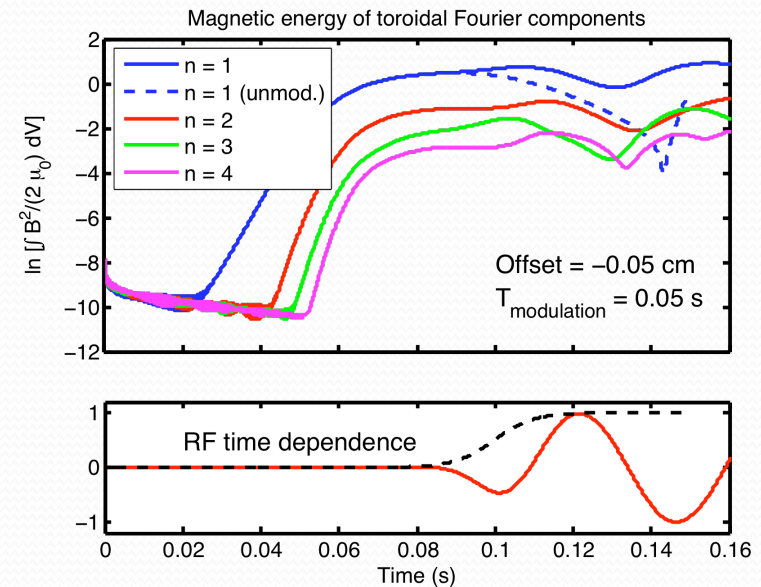
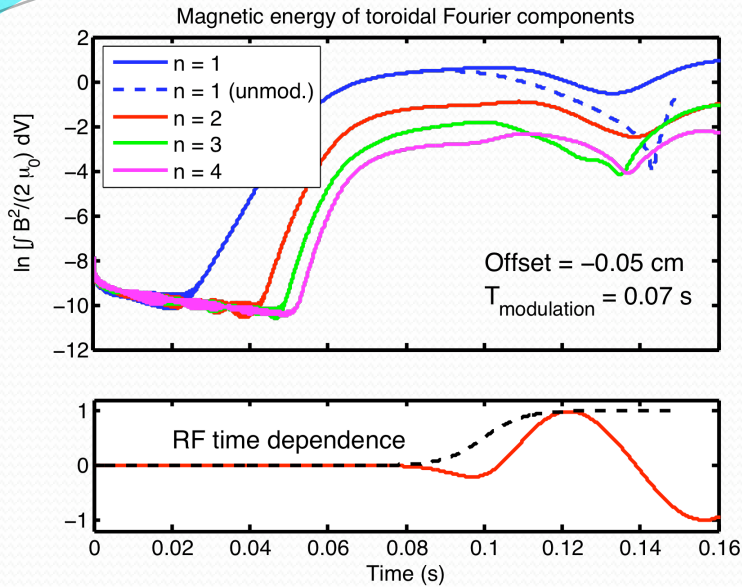
As RF suppresses the original islands, new islands arise

- (2,1) islands grow up again after (4,2) island has been suppressed; new islands are 90° out of phase (poloidally) from the old ones.



- Can we vary the RF amplitude in time to keep island width small?

Initial time-modulated RF shows some promise



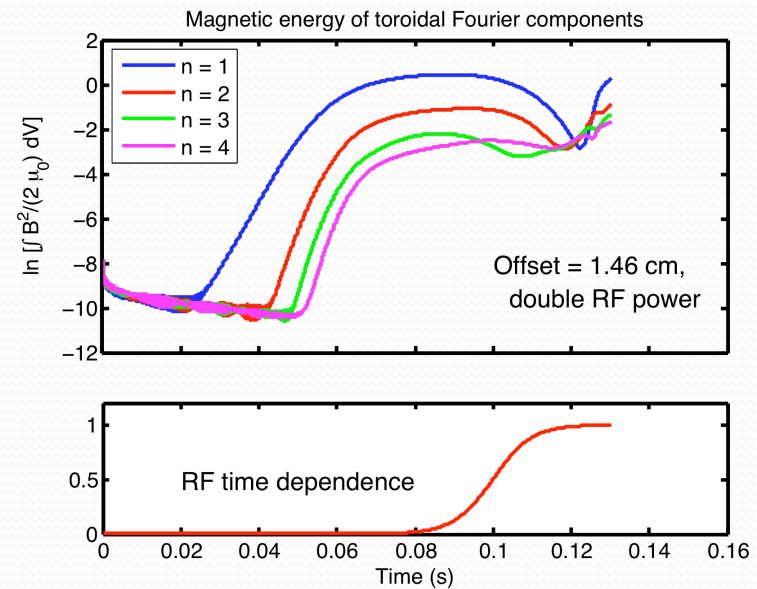
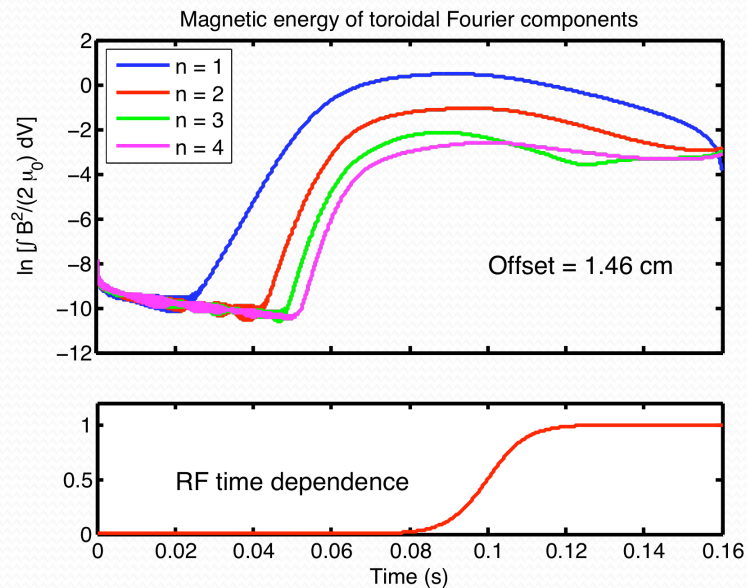
- RF needs more abrupt transitions to push amplitude down – sinusoidal oscillations can vary amplitude, but not reduce it much.

RF time dependence:

$$\sin[(2\pi t/T_{\text{mod}}) + \Phi] \times [\tanh((t-t_0)/t_p) - \tanh(t_0/t_p)]$$

Increases in RF power reduce island width more rapidly

- For otherwise identical runs, doubling the RF amplitude suppresses the mode much faster. (A work in progress...)



Where we are, presently

- Ad hoc model for RF deposition is giving interesting results, and hopefully leading to better intuition about physically self-consistent RF models.
- Sample IPS-NIMROD-GENRAY runs with dummy components work; good progress toward replacement of dummy components with actual physics codes
- Getting NIMROD data into Plasma State is almost completed
- Getting GENRAY data into NIMROD seems to work (compiler issues?) for Ohm's law term. Ready to work on transfer of quasilinear diffusion operator (for momentum/energy equations, and closures).