

Selected highlights from the SWIM project May 2010

Predictive Modeling of a Non-Resonant Internal Mode in a Beam Heated NSTX Discharge

We have used the SWIM framework to perform detailed predictive modeling of the onset and saturation of a non-resonant internal mode in NSTX. The SWIM framework provides a convenient method to bring together state-of-the-art tokamak physics codes and code packages. In these simulations, we make use of actual coil currents (for example for NSTX shot #124379) in the free-boundary Tokamak Simulation Code (TSC) to simulate the discharge from 0.1 to 0.70 sec. The Monte Carlo neutral beam code NUBEAM is used to calculate neutral beam heating and current drive. We model the ramp-up and flat-top phases of the discharge using a theory-based transport model that closely reproduces the measured temperatures and densities in the experiment. The evolution of the current profile in the presence of inductive drive and sources due to both neutral beams and bootstrap current is calculated. As a figure of merit, we calculate the root-mean-square of the difference between the experimental and simulation time-dependent poloidal flux values at a series of 40 flux loops around the device. The safety factor (q -profile) is well above one at the start of the simulation (early in the current ramp-up) but monotonically decreases during the simulation flat-top phase, in agreement with the experiment. As the central value of the q -profile, q_0 , approaches unity, we begin analyzing the discharge stability at frequent time slices using the M3D- C' code and find that we can closely predict the onset time of the observed $n=1$ mode that develops in the experiment.

The equilibrium found to be linearly unstable is then analyzed by the M3D code to determine saturation amplitude. In an illustrative calculation, we follow the nonlinear evolution of a linearly unstable $n=1$ mode in NSTX. The Poincare plots in Figure 1 show the central part of the discharge deforming in a quasi-helical manner and saturating, in qualitative agreement with the experimental results. In Figure 2, we show the plasma temperature contours at a later phase of the saturation, showing that the core confinement can be greatly degraded.

In continuing studies, we are improving the realism of our models by developing the ability to import density and/or temperature profiles directly from the experimental data so that our ability to predict these profiles can be better tested. We are also including plasma rotation, and the ability to import the high-energy neutral beam distribution function from NUBEAM to the stability codes.

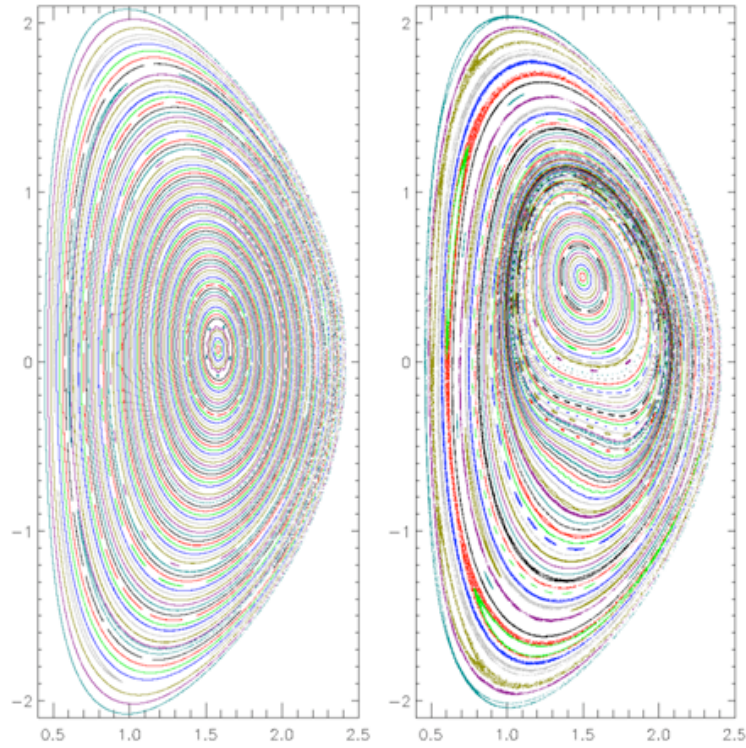


Figure 1: Poincaré plots before and during when internal mode develops and saturates.

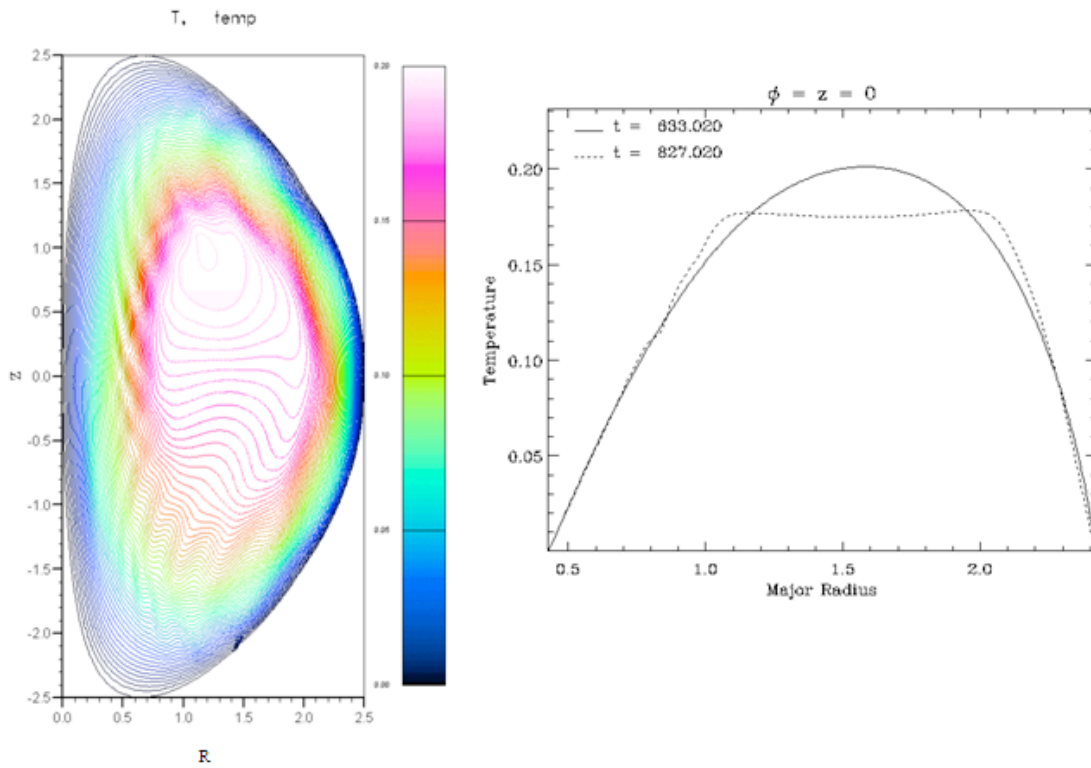


Figure 2: Temperature contours (left) and midplane profile showing flattening due to saturated internal mode.

Studies of tearing mode stabilization by electron cyclotron current drive

Experimentally, the use of radio frequency (RF)-generated electron cyclotron current drive (ECCD) to suppress tearing mode-induced magnetic islands in tokamaks has been highly successful. The exploration of the detailed physics of this ECCD/MHD interaction (through numerical simulation) is the goal of the SWIM Slow MHD campaign. For this it is necessary to introduce new terms into the MHD equations that model the effects of RF. This effect is a diffusion in velocity-space somewhat similar to the effect of collisions but tending to drive the plasma away from thermal equilibrium rather than toward it. Recent efforts have focused on determining the correct numerical implementation of this so called RF quasilinear diffusion in NIMROD (a nonlinear extended MHD code) and verifying its accuracy. The quasilinear operator is constructed using data from GENRAY, a ray tracing code that calculates the propagation and power deposition of RF waves using NIMROD's fields and plasma profiles.

At lowest order, the RF appears as an effective current in the MHD Ohm's law. Simulations with NIMROD including this current, but not the velocity-space diffusion, have demonstrated that resulting motion of rational surfaces due to RF induced currents has quantifiable effects on the stability properties of tearing modes [T. G. Jenkins et al., Phys. Plasmas 17, 012502 (2010)]. We have also demonstrated the complete stabilization of low-beta tearing modes via ECCD injection, and verified NIMROD's capability to model temperature equilibration around existing island structures [see Figure 3]. Our ongoing work toward the implementation of the quasilinear operator will enable us to explore how these effects, as well as the dynamics of the tearing mode itself, are influenced by the transfer of momentum and energy from RF waves to the plasma. We anticipate that additional ongoing work towards the determination of suitable RF-modified closures for the coupled RF/MHD system will permit the study of neoclassical tearing mode suppression in experimentally relevant, high-beta plasmas.

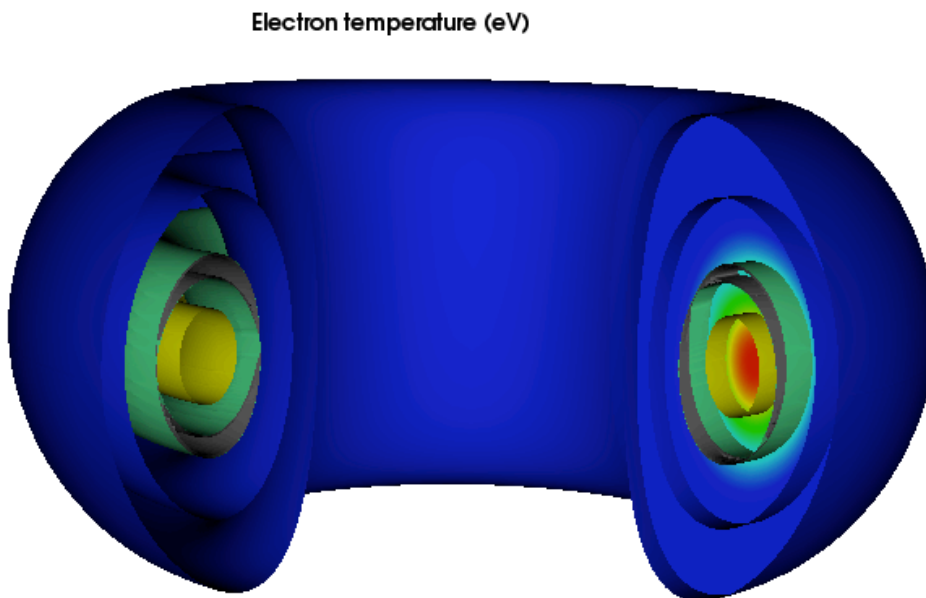


Figure 3: NIMROD simulations with a magnetic island. Color indicates electron temperature. Note equilibration of temperature on island structure, in green.

Enhancements to SWIM IPS framework for better concurrency and resilience

SWIM's Integrated Plasma Simulator (IPS) framework has had many recent enhancements focusing on both functionality and resilience. The previously-developed concurrent multitasking capability of the IPS has been extended to support pools of parallel tasks. This allows the simulation driver to effectively launch many more parallel tasks than there are resources to execute simultaneously, leaving it to the IPS to schedule their execution. This capability provides a very general mechanism for IPS-based applications to express high levels of concurrency, and the possibility to interact with the system's resource manager to grow and shrink the pool of allocated resources according to the needs of the job.

A challenge for many areas of simulation science, not only fusion, is to make effective use of hundreds of thousands to millions of processors for modeling systems that evolve in time. Such systems are not intrinsically parallelizable in time, since one cannot calculate a given segment of time until all of the previous time has been calculated. However using an algorithm called the Parareal algorithm [1] it is sometimes possible to break this impasse and effectively parallelize over time. In this approach two levels of time evolution solvers are used, a 'coarse solver' which can have low accuracy and use approximate physics models but must run very fast, and a 'fine solver' which has high accuracy and the desired level of physics fidelity and therefore runs much slower. The coarse solver is used to quickly obtain an approximate solution over the full time interval. The full time interval is broken into many short sub-intervals and the fine solver is run in parallel on each of these after being initialized at the beginning of the sub-interval using the previously obtained approximate solution. The Parareal algorithm provides a method to iterate this process and converge to an accurate solution over the full time. The final solution requires more computer cycles than a straight through solution with the fine solver, but takes advantage of the large number of available processors to reach the solution in less time.

The concurrent IPS framework has now been used to implement the Parareal algorithm. We have demonstrated the success of the method by solving the Lorenz system of equations. The Lorenz system consists of 3 nonlinear, coupled ordinary differential equations, first introduced to model atmospheric convection and which can exhibit chaotic behavior. The IPS Parareal simulation is also being applied to drift wave turbulence to understand the efficiency of Parareal in a fusion context. This work is supported as an Oak Ridge Laboratory seed money (LDRD) project.

The resilience of long-running simulations on large-scale HPC systems is an area of growing concern to many application teams. SWIM, in collaboration with the OASCR-funded Coordinated Infrastructure for Fault Tolerant Systems (CIFTS) project, has integrated the CIFTS Fault Tolerance Backplane into the IPS, and demonstrated simulations which use FTB information to run through and adapt to hardware failures. The IPS has also been recently enhanced to provide coordination for checkpointing individual components, one of the practical challenges to resilience in the concurrent multi-tasking environment that the IPS provides.

[1] J.-L. Lions, Y. Maday, and G. Turinici. A parareal in time discretization of pde's. C.R. Acad. Sci. Paris, Serie I, 332:661–668, 2001. URL <http://www.elsevier.nl/geom/10/37/18/47/27/35/article.pdf>.

Use of SWIM concurrent, multi-tasking framework to accelerate ITER scenario studies

The design, implementation, and successful testing of a parallel framework (the Integrated Plasma Simulator) has allowed time dependent simulations of simultaneous neutral beam injection (NBI) heating, ion cyclotron resonance heating (ICRH), and plasma discharge evolution to be carried out for the ITER burning plasma, with a physics fidelity that was never before possible. The simulations employed the massively parallel NUBEAM code to calculate neutral beam injection and fusion alpha particle dynamics using one million particles for each energetic species. This large particle count gave sufficiently smooth and accurate distribution functions as to be useable directly in wave heating codes. Also the massively parallel ion cyclotron RF code TORIC was used with much higher spatial resolution than was previously feasible. Thus we are able to complete fully coupled, high physics fidelity ITER simulations in a small fraction of the time these kinds of simulation usually require.

Now the framework has been extended to flexibly manage the concurrent multi-tasking of massively parallel components. This not only shortens the simulation time by bringing in another level of parallelism but also allows flexibility to improve load balancing and minimize time in essentially serial tasks. Although there are many approaches to using this capability, a simple example is shown in Figure 4. Here the framework is used to run TORIC and NUBEAM concurrently at each time step, then perform several minimally parallel task concurrently to improve load balancing. A similar scheme has been used to run both the NUBEAM and TORIC codes simultaneously and to directly couple to the TSC transport code.

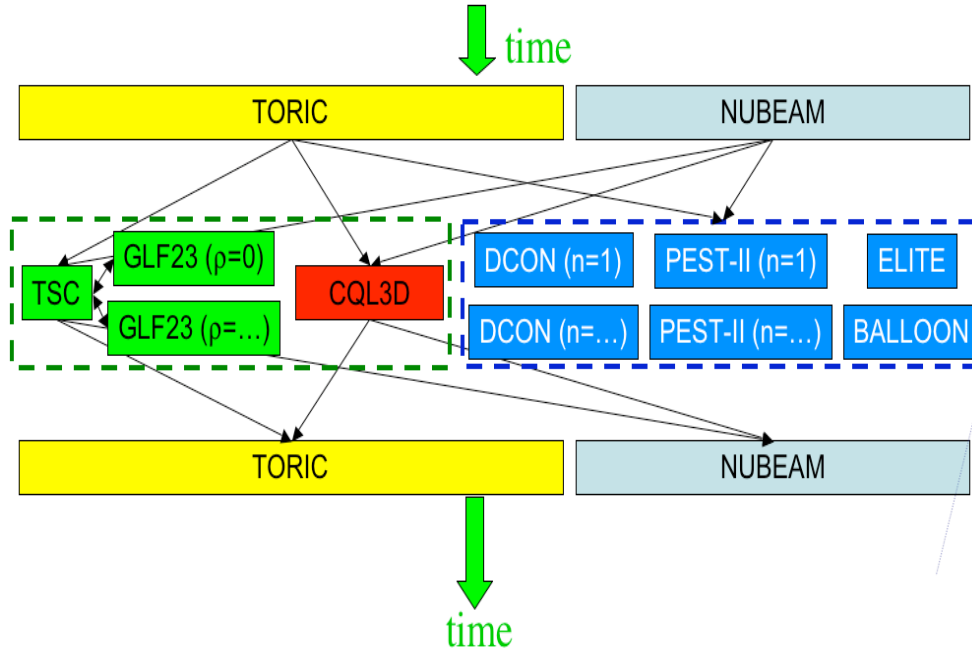


Figure 4: Example simulation flow using concurrent IPS framework to accelerate scenario simulations and improve load balancing


Improvements to SWIM monitoring portal

An improved version of the SWIM monitoring portal software has been deployed. While it is compatible with the earlier version, it has several new capabilities that allow SWIM users to monitor runs and collaborate more effectively. Users can (1) write and share comments on every simulation after the run, (2) rate each simulation to indicate the quality or importance of the simulation, (3) purge unwanted simulation results or restore them if purging was a mistake, and (4) rapidly search and sort runs by multiple fields providing a greatly improved discovery capability.

The new portal software relies on standard open source web technologies and leverages work done to support DIII-D experimental operations. The development was done using the Python programming language and the Django web framework. It relies on MySQL as the database and Apache as the web server. The new monitoring portal is more robust and flexible compared to the older Java-based version. The new portal is more flexible for the end-user since it can be served simultaneously via multiple ports removing the somewhat onerous requirement for some institutions to modify their institutional firewall restrictions. The other benefit of the new portal is its support for agile software development. Since being released, numerous incremental improvements to the SWIM portal have been made without affecting users while it was running in production mode. The portal is now used heavily by the project and finding time to schedule taking the system down is very difficult. Therefore, the ability to make changes in real-time to the production system is a critical requirement.



 Center for Simulation of
 RF Wave Interactions with
 Magnetohydrodynamics
Monitor


[About](#) | [Logout](#)

Search:


[Show My Runs](#) | [Show Purged Runs](#) | Sorted by **RunID** in ascending order

RunID	Rate	Purge	Status	User	Last Update	Code	Time-stamp	Comments
0 16824	★ ★ ★		Running	Batchelor	2010-05-13 15:44:08	epa__tsc	14.000	Target = aprun -n 1 -cc 3-3 -N 1 /project /projectdirs/m876/phys-bin/phys//tsc/bin /tsc_041210 hy040510_001 ITER 2010 001 step, task_id = 42
0 16823	★ ★ ★		Completed	wspear	2010-05-13 02:05:48	Framework	-1	Simulation Ended
0 16822	★ ★ ★		Completed	wspear	2010-05-12 23:07:29	Framework	-1	Simulation Ended
0 16821	★ ★ ★		Running	Batchelor	2010-05-11 19:50:13	epa__tsc	14.000	Target = aprun -n 1 -cc 3-3 -N 1 /project /projectdirs/m876/phys-bin/phys//tsc/bin /tsc_041210 hy040510_001 ITER 2010 001 step, task_id = 42
1 16820	★ ★ ★ ★ ★		Completed	Batchelor	2010-05-10 15:56:47	Framework	100.000	Simulation Ended
1 16819	★ ★ ★		Running	Batchelor	2010-05-07 16:41:31	drivers_dbb_generic_driver	100.000	Target = tsc@4:step()
0 16818	★ ★ ★		Completed	Batchelor	2010-05-06 20:47:20	Framework	16.000	Simulation Ended
0 16817	★ ★ ★		Completed	Batchelor	2010-05-06 01:57:18	Framework	100.000	Simulation Ended
0 16816	★ ★ ★		Completed	Batchelor	2010-05-05 19:24:15	Framework	16.000	Simulation Ended
0 16815	★ ★ ★		Completed	Batchelor	2010-05-05 18:27:36	Framework	16.000	Simulation Execution Error
0 16814	★ ★ ★		Completed	Batchelor	2010-05-05 17:03:13	Framework	16.000	Simulation Execution Error

Page 1 of 17. [next](#)

SWIM Team (2010)

Figure 5: Main page of the SWIM monitoring portal.



Center for Simulation of
RF Wave Interactions with
Magnetohydrodynamics

swim Monitor

About | Logout

Search:

Home

Portal Run ID: **16820**
Batchelor ★★★★

Run Comment:	hy040510_001 restart at 13 sec run to 16 sec TSC and TORIC nmod=127
Tokamak:	ITER
Shot No:	N/A
Sim Name:	hy040510_001
Sim RunId:	hy040510
Output Prefix:	N/A
Tag:	TORIC_scaling
Logfiles:	N/A
Visualization URL:	N/A

+ Add a new comment for 16820

Time	Commented by	Comment content
2010-05-11 01:07:59	abla	Test
2010-05-10 21:15:04	Batchelor	The portal shows that the simulation finished at 25,870 sec whereas the log file says: MONITOR finalize monitor finalize finished MONITOR finalize finished =>> PBS: job killed: walltime 37811 exceeded limit 37800 It looks like it sat there and

Time	Seq Num	Event Type	Code	State	Wall Time	Phys Time-stamp	Comment
2010-05-10 15:56:47	2107	IPS_END	Framework	Completed	25870.27	100.000	Simulation Ended
2010-05-10 15:56:47	2106	IPS_CALL_END	drivers_dbb_generic_driver	Running	25870.24	100.000	Target = monitor@3:finalize()
2010-05-10 15:56:47	2105	IPS_CALL_BEGIN	drivers_dbb_generic_driver	Running	25870.21	100.000	Target = monitor@3:finalize()
2010-05-10 15:56:47	2104	IPS_CALL_END	drivers_dbb_generic_driver	Running	25870.17	100.000	Target = tsc@4:finalize()
2010-05-10 15:56:47	2103	IPS_CALL_BEGIN	drivers_dbb_generic_driver	Running	25870.14	100.000	Target = tsc@4:finalize()
2010-05-10 15:56:47	2102	IPS_CALL_END	drivers_dbb_generic_driver	Running	25870.10	100.000	Target = toric@5:finalize()
2010-05-10 15:56:47	2101	IPS_CALL_BEGIN	drivers_dbb_generic_driver	Running	25870.06	100.000	Target = toric@5:finalize()

Figure 6: Shows the part of the detail of one simulation.