Center for Simulation of Wave Interactions with MHD (SWIM)

PASCI PAC meeting, May, 2007

E. D'Azevedo[1], D. B. Batchelor[1], G. Bateman[2], D. E. Bernholdt[1], L. A. Berry[1], P. T. Bonoli[3], R. Bramley[4], J. Breslau[5], M. Chance[5], J. Chen, M[5]. Choi[6], W. Elwasif[1], G-Y Fu[5], R.W. Harvey[7], W. A. Houlberg[1], E.F. Jaeger[1], S. C. Jardin[5], D. Keyes[8], S. Klasky[1], S. Kruger[9], L.P. Ku[5], D. McCune[5], J. Ramos[3], D. P. Schissel[6], D. Schnack[10], J.C Wright[3],

[1] ORNL, [2] Lehigh, [3]U, PSFC-MIT, [4]Indiana U, [5]PPPL, [6]General Atomics, [7]CompX, [8]Columbia U, [9]Tech-X, [10]U. Wisconsin

The broad scientific objectives of the SWIM project are: to improve our understanding of interactions that both RF wave and particle sources have on extended-MHD phenomena thereby improving our capability for predicting and optimizing the performance of burning plasmas in devices such as ITER: and to develop an integrated computational system for treating multi-physics phenomena with the required flexibility and extensibility to serve as a prototype for the FSP. The SWIM Center consists of three elements:

(1) **Development of a computational platform referred to as the Integrated Plasma Simulator (IPS)** that will allow efficient coupling of the full range of required fusion codes or modules.

(2) **A physics campaign addressing long timescale discharge evolution in the presence of sporadic fast MHD events**. This involves interfacing the IPS to both linear and 3D non-linear extended MHD codes and carrying out a program of research related to use of RF and other driving sources to study and control fast time-scale MHD phenomena such as optimizing burning plasma scenarios and improving the understanding of how RF can be employed to achieve long-time MHD stable discharges and control sawtooth events.

(3) **A physics campaign for modeling the direct interaction of RF and extended MHD for slowly growing modes.** This requires development of new approaches to closure for the fluid equations and the interfacing of RF modules directly with the extended MHD codes and with code modules that implement the fluid closures. The primary physics focus of this campaign is to improve the understanding of how RF can be employed to control neoclassical tearing modes.

The activities of the past year have primarily been completion of the design for the IPS, implementation of the IPS framework and components, code porting, initial testing, and physics analysis.

The IPS design is based on a component architecture in which the various required physics functionalities have been abstracted at a high level and formal interfaces defined such that the components can be implemented by any code that provides the required functionality, so that multiple code implementations can be used. The framework is written in Python for flexibility and portability. Its main functions are to assemble and instantiate the components required for the specific simulation and to provide a set of services to the components. The services include mechanisms for managing data, job launch, resources and events. The data manager does all of the data staging and archiving of the plasma state and other input and output files for the components. The job launch manager launches the component codes on a set of computer resources and monitors their status. The resource manager keeps track of what resources are available and determines what resources are given to a particular job. The event handler receives and publishes events from the codes, and eventually a fault tolerant back plane will allow modification of the workflow accordingly.

All components support a basic interface of ‘initialize’, ‘step’, and ‘finalize’ functions. and are implemented as Python scripts, most of which wrap large granularity physics codes in fortran. There is a distinguished Driver component that uses framework services, to set up the working directory structure, initialize the other components, start the physics time stepping loop, archive the plasma state and code output files at the end of each successful time step, and calls the component finalize functions at the end of the simulation. The time stepping loop of the driver is intended as a ‘physicist accessible’ layer that will allow great flexibility in how the user constructs the workflow of the simulation. Presently it consists of a simple explicit series of time-steps advancing the components sequentially, but a more elaborate time stepping scheme has been designed.

An important feature of the IPS is the exchange of simulation data between components using an intermediate Plasma State service that has a simple user interface and a standardized data format. It also provides services to map between different grids when the grid required by the components differs from that on which it is stored in the state. It is implemented as a fortran 90 module layered over the XPLASMA2 software, thus permitting multiple time-step instances to be in memory at once. The Plasma State code is automatically generated from a plasma state specification file allowing for ease and accuracy of modification and extensibility.

The framework and Plasma State service are fully implemented and tested, and several Driver components have been developed. The SWIM Plasma State service has been adopted by TRANSP and TSC as their communication mechanism for distributed computing, and is in routine use in TSC/TRANSP free boundary hybrid simulations. The TRANSP/TSC hybrid was developed as a prototype in the predictive TRANSP (PTRANSP) project. At present there are 6 types of physics components that are complete or nearly complete – Equilibrium and Profile Advance (EPA), RF solver, Fokker-Planck solver, MHD Stability, and Visualization. The main EPA component is based on the TSC code although other simplified EPA components have been written for testing purposes. Two interchangeable RF Ion Cyclotron components have been developed based on the AORSA2D and TORIC codes. A Fokker Planck solver component based on the CQL3D code has been developed and a Monte Carlo RF Fokker Planck solver based on ORBIT–RF is under development. It is planned to develop a Fokker Planck/neutral beam component based on NUBEAM although NUBEAM can already be used in SWIM simulations through the direct coupling between TSC and TRANSP. The linear MHD component contains the capabilities of equilibrium refinement, flux surface mapping, ballooning stability evaluation using BALLOON, and low-n stability evaluation using DCON, PEST-1 or PEST-2. It will be extended in the near future to include the energetic particle code NOVA-K and the nonlinear codes – M3D and NIMROD. Ultimately the MHD component will be extended to include nonlinear codes – M3D and NIMROD. The Visualization component extracts data from the simulation data structure and presents it in a form for runtime monitoring using ELVIZ.

A SWIM Web Portal is being developed to provide access to simulation data and for live monitoring of SWIM runs. While the Portal is intended for convenience to physicists, it is not a required single entry point for making simulation runs. Its goal is to provide a browser-enabled interface for secure accesses, launching and monitoring of simulation runs. Currently the data management back end (Obsidian) has been developed and runs with interfaces in C, Perl, and general web services. A preliminary schema has been created for file managment, and we are iterating on a schema for the physics metadata associated with the simulations. The current operational prototype of the portal also provides secure access, can monitor simulations via an event channel.

There have been two significant code-porting efforts. Since it was considered essential for development and testing purposes to have a stable computing environment under our control, an additional 16 processors for the PPPL Altix 350 cluster were purchased with SWIM funds and all relevant codes ported to that platform. Also to prepare for production runs requiring leadership scale computing, the major MPP codes implementing SWIM components were ported to the Jaguar computer at NCCS and optimization studies were performed. The AORSA2D code has scaled very well up to 22,500 processors with high processor efficiency using High Performance LAPAC for the dense linear solves. M3D has been ported to both the Phoenix and Jaguar machines at NCCS and scaling studies up to 5,120 nodes on Jaguar were performed with up to 80% efficiency using the Hypre algebraic multigrid solver within PETSc. NIMROD has also been ported to Jaguar and production runs are beginning. The CQL3D and ORBIT–RF Fokker Planck solvers were also ported to Jaguar.

The primary effort in physics analysis has been to address how RF waves modify the MHD equations and how to include RF effects in the fluid closure as required by the Slow MHD campaign. A multi-level approach for has been developed for simulating the interaction of an RF source with magnetic islands in toroidal plasmas. This involves three levels of sophistication that can be pursued somewhat in parallel. The first approach is a computational effort to model the interaction RF with magnetic island evolution by inserting an analytically chosen form for a source term in the Ohm's law. In the second approach, a phenomenological evolution equation will be used to describe the temporal and spatial structure of the source term. In the third approach, a more rigorous analytic problem will be solved where the inclusion of RF effects are treated as closure problems. The modified equations can then be implemented in numerical simulations. In the second and third approaches direct interfaces with the RF codes are required.