Computational Plasma Physics

The Computational Plasma Physics Group (CPPG) at the Princeton Plasma Physics Laboratory (PPPL), develops and applies advanced computational techniques in the areas of production running of analysis codes, scientific visualization, parallel scaling, and algorithmic development. Some of the highlights of the FY07 year are given below.

TRANSP Run Production

In its fifth year of operation, utilization of PPPL's FusionGrid TRANSP computational service continued its rapid growth. TRANSP runs are used for time-dependent analysis and simulation of tokamak experiments — the software has been under continuous development at PPPL and elsewhere for more than 30 years. Fiscal year 2007 saw the introduction of parallelized neutral-beam particle simulations into TRANSP code run production. The PTRANSP (predictive TRANSP) project was resumed in FY07; numerical stability enhancements in temperature prediction enabled the code to be used for a major ITER study. Figure 1 shows the growth in usage in TRANSP over the last three years and the distribution of this set of runs by tokamak device.

Documentation, scripts, and utility code to help prepare and submit runs to the service are provided at http://w3.pppl.gov/transp. Users of the service monitor the progress of their runs at http://w3.pppl.gov/transp/transpgrid_monitor, which gives links to log files and browser enabled graphics of the input data and namelist selected runtime data. Any runs which experience problems during execution are examined by a PPPL expert and the result of this analysis is communicated back to the user through the web interface or by email. Security for the service and the user's data is provided through the use of FusionGrid Certificates (standard internet protocols) as described at the web site. About 100 research users are presently registered with the TRANSP compute service. Access is granted to new users as a routine matter (as a first step, users apply for a PPPL computer account and agree to PPPL Cyber-security policies).

Free-boundary Equilibrium Solver in TRANSP/PTRANSP

TRANSP is being enhanced with new predictive capabilities as part of the PTRANSP project. In support of this effort the reconstruction of the MHD equilibrium with a free-boundary solver has been added to TRANSP. The free-boundary solution improves the fidelity of the magnetic field outside the plasma for the radio-frequency and neutral-beam models and provides a framework for the future coupling of the magnetic field diffusion with the coil and circuit equations. Longer term, it will enable coupling of PTRANSP to a scrape-off layer plasma model. The

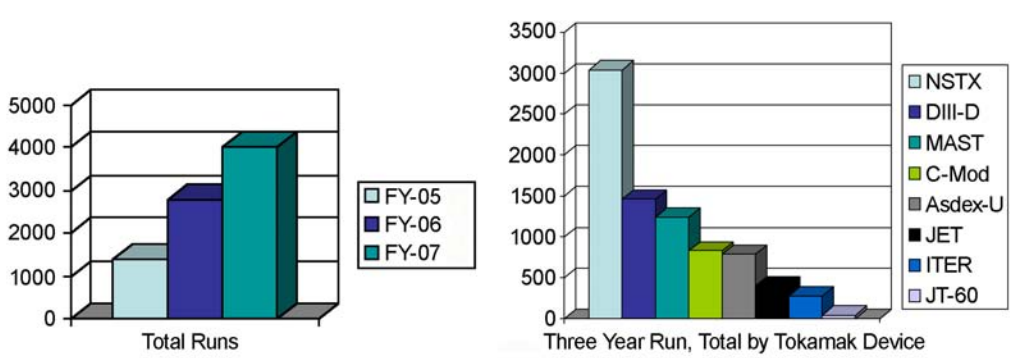

Figure 1. TRANSP run production FY05–07. There were 8,188 runs produced in the last three years, with an increase of about 300% in production in FY07 compared to FY05. More than 60% of the runs were for experiments not sited at PPPL.

free-boundary solver within the TEQ code from the Lawrence Livermore National Laboratory was integrated into TRANSP. Preliminary testing was done on PTRANSP code runs. Figure 2, shows the magnetic flux surfaces in an ITER simulation.

Web Services for Fusion Codes

Based on previous experience in programming methods, the CPPG has begun deploying efficient web-based graphical interfaces that run from a browser and provide secure Internet access to site-specific data and computational physics applications. Graphical user interfaces, running in web browsers for portability and convenient access, communicate with fusion codes running on compute servers maintained at PPPL (Figure 3). The web service approach facilitates collaboration, enforces security, eliminates software installation, and provides broad access to scientific applications. This approach adds visualization to applications with minimal changes to the simulation codes. The client software is written in Java so it is portable across users' computer platforms. The clients send requests to the web server where they are managed by PPPL's Visualization Servlet program. Fusion programs such as RPLOT and GTC-S run on the compute servers and return results to the client for interactive visualization.

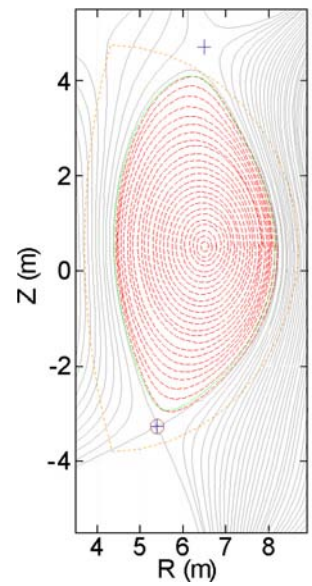

Figure 2. Plot of a poloidal magnetic flux surface reconstruction from the TEQ free-boundary solver during a PTRANSP code run for an ITER plasma.

New Performance Milestones Achieved by the GTC-S Code

The GTC-S code, which was developed based on a generalized gyrokinetic particle model using the

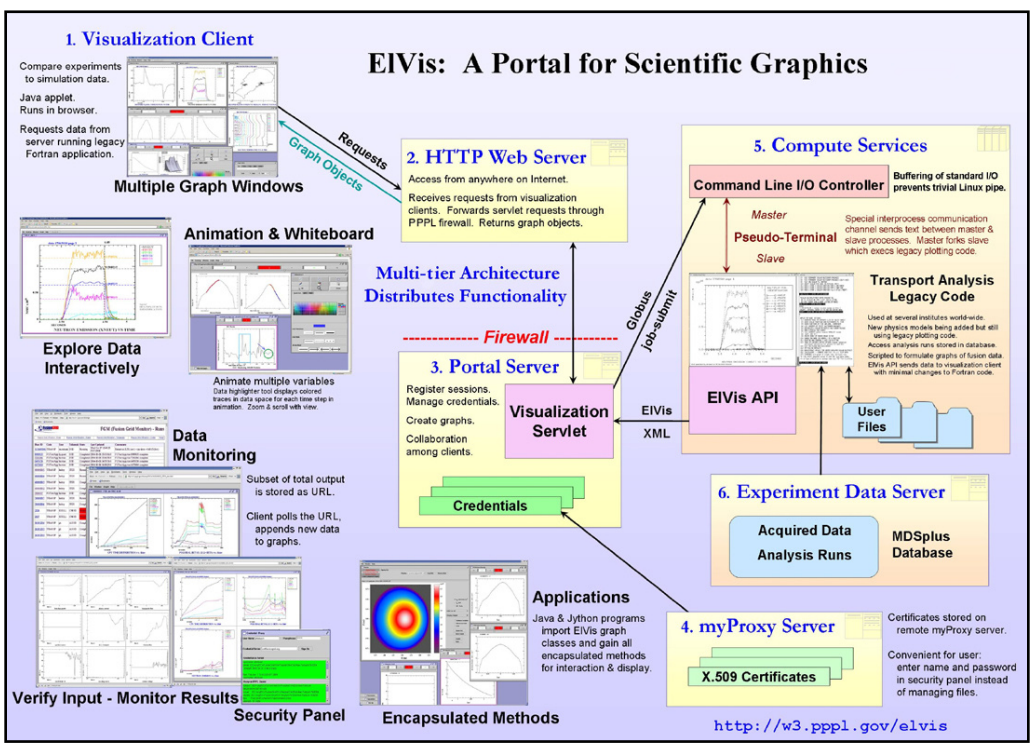

Figure 3. This diagram illustrates a multi-tier architecture for implementing web services to fusion codes.

original Gyrokinetic Toroidal Code (GTC) architecture, simulates plasma microturbulence and associated transport in toroidal (tokamak) experiments. GTC-S was chosen by the Office of Advanced Scientific Computing Research (OASCR) as one of three large-scale applications to fulfill the FY07 JOULE Milestones set by the Office of Management and Budget (OMB). Every year for the past few years, the OMB has set milestones for the most intensive large-scale scientific applications to be run on the OASCR-funded Cray XT4 system, Jaguar, located at the Oak Ridge National Laboratory (ORNL). The FY07 milestones called for a doubling in overall performance of the applications.

Unlike the original GTC, the GTC-S code can treat globally consistent, shaped cross-section tokamak plasmas by directly importing experimental profiles of plasma temperature, density, and rotation, along with the related numerical MHD equilibrium. It can also include equilibrium mean electric field and Coulomb collisions. Those important effects strongly influence the dynamics of the fusion plasmas and are crucial for true validation with experiments and for achieving predictive capability. Although most of the work on GTC-S has been focused on the physics upgrade, the JOULE exercise stimulated significant effort to improve its performance and concurrency through the involvement of CPPG.

Having been specifically developed on the IBM SP at the National Energy Research Supercomputer Center (NERSC), the GTC-S code was first ported and optimized on the Cray XT4 at ORNL. All dependencies on third-party libraries were removed and replaced by Open Source alternatives, which required thorough testing for accuracy and performance. A new level of parallelism was then added to the original toroidal domain decomposition and OpenMP loop-level multithreading by implementing a particle distribution algorithm. This resulted in a considerable increase of concurrency, from a few hundred processors originally on the Cray XT4 to several thousands with the new algorithm.

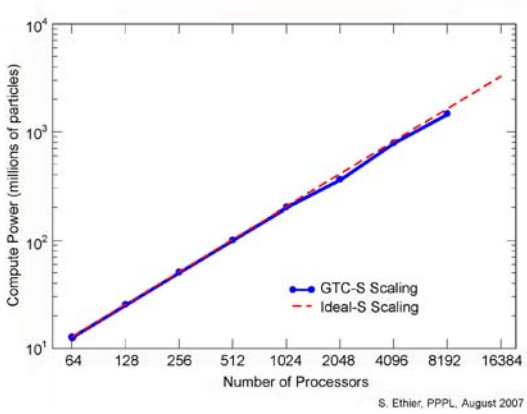

Figure 4. Weak scaling study of the newly improved GTC-S code on the Cray XT4 system (Jaguar) at ORNL. The blue curve shows the number of particles (in millions) moved one step in one second by the GTC code, and the red curve shows the corresponding theoretical maximum assuming ideal scaling.

All of the improvements implemented in the GTC-S code allowed it to easily exceed the FY07 JOULE milestones set by the OMB. Scalability was demonstrated from 64 cores, the original limitation, to 8192 cores, the maximum number of cores available on Jaguar at the time. Figure 4 shows the weak scaling performance of the improved GTC-S on the ORNL Cray XT4 at ORNL.

Strong Scaling of 3-D MHD Codes to Large Numbers of Processors

The M3D code is used for simulating global instabilities in magnetic fusion devices. It is one of the major workhorse codes in use by several of the U.S. Department of Energy (DOE) Office of Science's Scientific Discovery through Advanced Computing Program projects: the Center for Extended MHD Modeling (CEMM, see http://w3.pppl.gov/CEMM), the Center for Simulation of RF Wave Interactions with Magnetohydrodynamics (SWIM, see http://cswim.org/), and the Center for Plasma Edge Simulation (CPES, see http://www.cims.nyu.edu/cpes). It is a high priority for each of these centers to have the M3D code run efficiently on the present and the next generation of DOE computers. The multi-scale multi-physics nature of the M3D code necessitates the use of partially implicit algorithms, which are known to be a challenge to implement efficiently on massively parallel computers.

Previous scaling studies with M3D involved what is known as "weak scaling." In these studies, one begins with a small problem run on a small number of processors. As the problem size (number of mesh points) is repeatedly doubled, the number of processors is simultaneously increased to keep ratio of mesh points to number of processors fixed. Perfect weak scaling would imply that the wall clock time to problem completion stays fixed as you continue to do this. This is by far the most common type of scaling study.

However, what is more relevant for the needs of the above Centers is "strong scaling." In this, one fixes the number of mesh points required for adequate spatial

resolution and time the run using more and more processors for that given mesh. In perfect strong scaling, the running time would decrease (or the run would speed up) proportional to the number of processors that is added. Since the demonstration of strong scaling is essential in order to study tokamaks with ITER parameters, this was the focus in FY07.

Building on previous accomplishments during the previous year's weak scaling campaigns, considerable progress was made in improving the strong parallel scaling properties of the M3D code This was accomplished primarily by: tuning the HYPRE multigrid Poisson solver, further optimizing the algorithm for distributing the mesh points over the processors, eliminating some inefficiency in the inter-processor communications that set in when operating at more than 1,000 processors, and working with the Center consultants to track down bugs in the operating system that only occur for the largest problems. As seen in Figure 5, favorable strong scaling has now been demonstrated up to 5,000 processors on the ORNL Jaguar computer with a parallel efficiency of more than 70%.

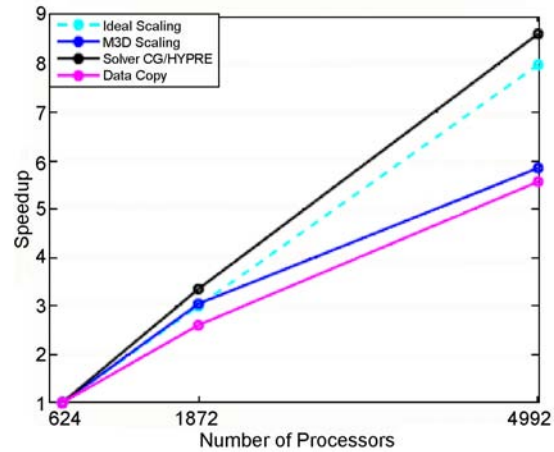

Figure 5. Speedup relative to 624 processors for strong scaling of the M3D code on the massively parallel ORNL computer, Jaguar. The blue curve reflects the overall running time. Future emphasis will be in improving the parallel scaling properties of the "data copy" segment of the code.

A Fully Implicit Method for Resistive MHD

It is widely recognized that temporal stiffness induced by the ideal-MHD characteristics generally make it impractical to simulate resistive MHD phenomena using an explicit time-stepping method. In a close collaboration between the APDEC (Applied Partial Differential Equations Center), TOPS (Towards Optimal Petascale Simulations), and CEMM SciDAC Centers, a fully nonlinearly implicit Newton-Krylov (NK) method was developed for resistive MHD. The equations are written in a fashion such that all the unknown quantities (and their derivatives) are written as a nonlinear function, which is solved using the multivariable Newton method. Each step in the Newton method requires the inversion of a rather large linear system.

The key to achieving a scalable implicit method is to solve these linear systems using a Krylov technique (such as GMRES), which does not require explicit storage or computation of the Jacobian at each Newton step. Hence these methods are often dubbed "Jacobian-Free Newton-Krylov (JFNK)." Furthermore, at the heart of efficient JFNK methods often lies an effective preconditioner, which approximates the Jacobian, resulting in a fast solution during the Krylov step. An operator-based preconditioner applicable to any general system of hyperbolic conservation laws was developed and applied to the resistive MHD system.

The idea behind this preconditioning approach is that because the stiffness of the ideal-MHD system results from the fast compressive and Alfvén MHD waves, an approximate decomposition of the system into its component waves can be developed and preconditioning is done only on the stiffness-inducing parts. This approximate decomposition is based on a second-order time accurate operator splitting of the MHD system into its directional components. Each component then forms a system of eight coupled, one-dimensional advection equations, which are projected into a set of decoupled characteristic equations. The result of this decoupling is a linear block-tridiagonal system of equations which can be easily solved. Within this strategy each characteristic equation is solved independently, enabling solution of only those components inducing stiffness to the fully implicit system, while leaving the slower components alone.

Upon solution of these decoupled equations, the preconditioned solutions are projected back into their original conserved variables, resulting in the approximate solution of the original linear Jacobian system. This approach gives one the freedom to choose to precondition the fast compressive wave, or both the fast compressive and Alfvén waves, or all the waves in the MHD system.

Results of small-amplitude wave propagation tests are shown in Figure 6. Other nonlinear tests also

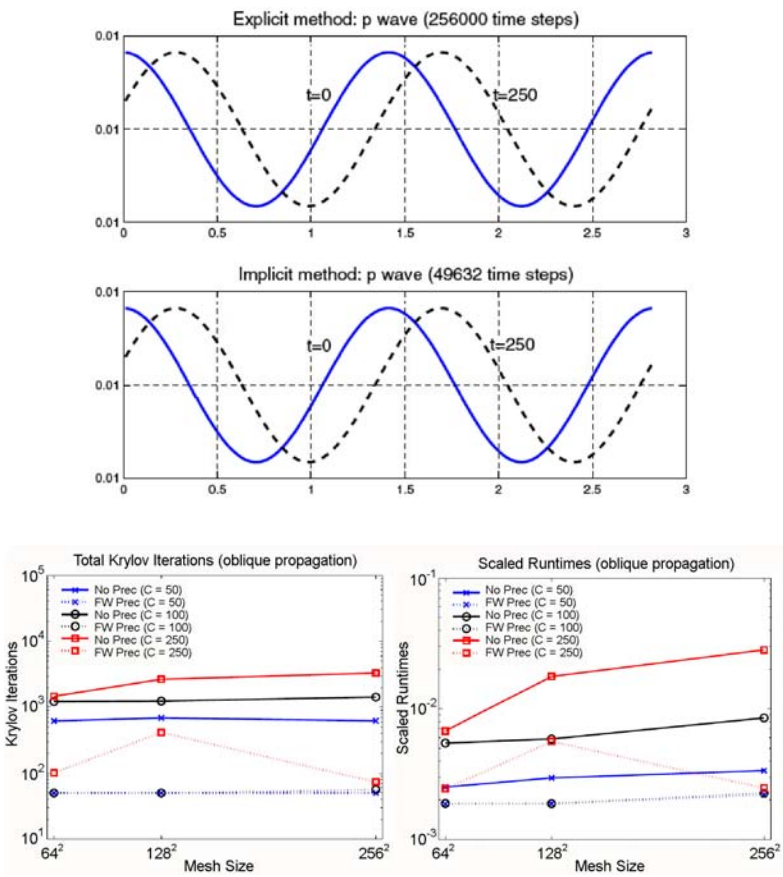

Figure 6. Wave propagation test: The top two panels show a comparison of the implicit solution versus an explicit solution method after 88 transit periods. The bottom panel shows that the number of Krylov iterations decreases as a result of preconditioning (left) and that the scaled CPU time (CPU time per mesh point so that a horizontal line implies perfect scaling) shows good scalability (right). Note: C is the ratio of the time step taken in the implicit method to the Courant-Fredrichs-Lewy-constrained time step of an explicit method.

showed improvements of the preconditioned JFNK over explicit and un-preconditioned methods. Other salient features of this method are: it is conservative and preserves the solenoidal property of the magnetic field.

Development of the M3D-C^1 code for 3-D Two-fluid MHD Studies

Work continued on the development of a new form of the M3D code using a strongly implicit algorithm that is made possible by the use of finite elements with continuous first derivatives (C^1). Significant progress was made in a number of areas.

A new implicit time advance was implemented that exhibits improved numerical stability, especially at the highest spatial resolution, and also improved convergence in the time step Δt, especially for the steady-state solutions. This has allowed use of this code to calculate toroidal equilibrium with flow as shown in Figure 7.

The geometry of the 2-D code was generalized to now be toroidal, with fully unstructured triangular elements. This now uses routines from the Rensselaer Polytechnic Institute (RPI) Scientific Computational Research Center (SCOREC) to hide parallelism and provide mesh functions that allow adaptivity. The code was also generalized to use sparse matrix solvers through the PETSc interface. This makes it possible to easily compare different solvers, and gives a natural path for an extension to 3-D.

These improvements have enabled two new physics studies in 2-D geometry: the study of toroidal two-fluid equilibrium with gyroviscosity and flow and the study of 2-D magnetic reconnection in the presence

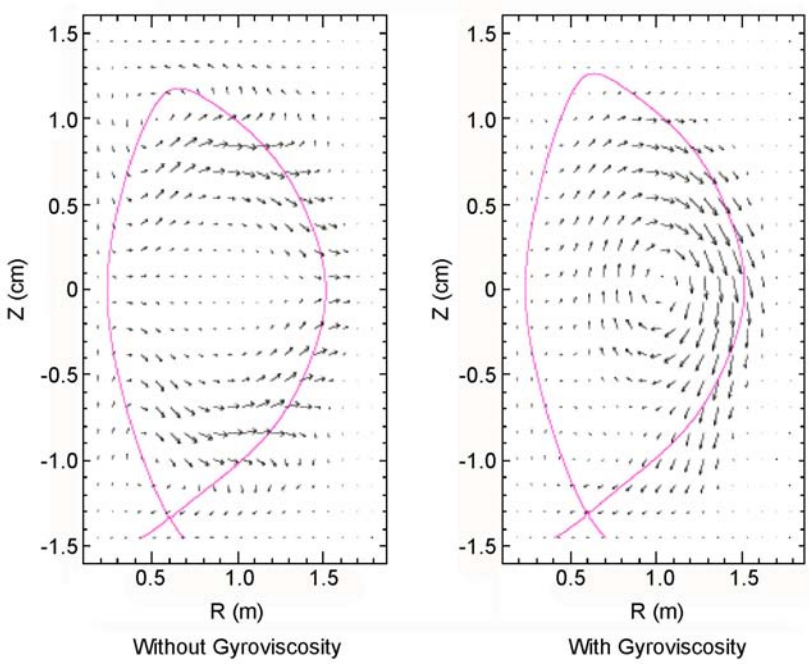

Figure 7. Steady-state toroidal equilibrium on all time scales as calculated as a steady state of the 2-D toroidal M3D-C^1 code. Solution on the left does not include gyroviscosity and does not exhibit rotation. Solution on the right includes gyroviscosity and exhibits spontaneous poloidal rotation.

of a strong guide field. This latter study shows that a guide field can strongly inhibit the onset of fast reconnection. A large parameter regime is being studied to put this result in context, but the fact that the simulations exhibit excellent energy and flux conservation and have been confirmed by an initial comparison with a similar NIMROD code calculation gives confidence in these results.