Massively Parallel Computing of Plasma Turbulence in Tokamaks

Jeremy Kepler  
Dept. of Astrophysics  
Princeton University  
vkepler@astro.princeton.edu

Scott Parker  
Dept. of Physics  
University of Colorado  
sparker@buteo.colorado.edu

Viktor Decyk  
Dept. of Physics  
UCLA  
vdecyk@pepper.physics.ucla.edu

Abstract
One of the most important problems in magnetically confined fusion research is the turbulent transport of heat observed in present-day Tokamak experiments. Plasma turbulence is very challenging from a theoretical point of view due to the nonlinearity and high dimensionality of the governing equations. Experimental measurements in the core region of a Tokamak are limited by the extremely high temperatures ~10^8 °K. The level of both theoretical and empirical difficulty highlights the importance of numerical simulations. These simulations can potentially play in developing a predictive model for turbulent heat transport. Such a model would dramatically reduce uncertainties in design and may lead to enhanced operating regimes, both of which would reduce the size and expense of a fusion reactor. Simulating this highly non-linear behavior requires solving for the perturbations of the phase space distribution function in five dimensions (via & methods using the gyrokinetic formalism). We use a particle-in-cell approach to solve the equations. The code has been parallelized for a variety of architectures using a 1-D domain decomposition along the toroidal axis. When the simulations start with a very small initial perturbation, there are generally two identifiable phases of the run. First is the linear phase, where modes (i.e., standing waves) grow exponentially. Linear modes can also be found using lower-dimensional, time-independent eigenvalue techniques and are fairly well understood theoretically. The second phase is the turbulent, poloidal modes. The mode is very elongated in the direction following the magnetic field lines. The ions within the plasma move rapidly around the torus, gyrating tightly along the magnetic field lines, like rings on a wire. The radius of gyration p is set by the velocity, mass and charge of the particle. The essential scale of the system is set by a/ρ. Typically p ~ 270 cm, a ~ 85 cm, and ρ ~ 0.15 cm.

Implementation
The ions within the plasma move rapidly around the torus, gyrating tightly along the magnetic field lines, like rings on a wire. The radius of gyration p is set by the velocity, mass and charge of the particle. The essential scale of the system is set by a/ρ. Typically p ~ 270 cm, a ~ 85 cm, and ρ ~ 0.15 cm.

The simplest arrangement of field lines is obtained by wrapping current carrying wires tightly around the minor axis of the torus creating straight magnetic field lines aligned with the torus. Unfortunately, the magnetic field exerts a greater force on the inside of the torus, which causes the ions in the plasma to drift across the field lines. This problem can be partially alleviated by twisting the field lines into a helical shape so that the drifts approximately cancel. A byproduct of this twisting is that the trajectories of particles are now far more complex and are susceptible to a wide range of instabilities, which tend to grow along the toroidal modes of the Tokamak. Figure 2 shows a simplified example of a typical toroidal mode. This could be, for example, the linear growth phase of an ion density perturbation. Generally, the sources of a perturbation follow the helical shape of the magnetic field lines.

Results
When the simulations start with a very small initial perturbation, there are generally two identifiable phases of the run. First is the linear phase, where modes (i.e., standing waves) grow exponentially. Linear modes can also be found using lower-dimensional, time-independent eigenvalue techniques and are fairly well understood theoretically. The second phase is the turbulent, stationary state, during which the growth of the dominant linear modes saturates and the system settles down to a statistical steady state. The ion density in the linear mode is shown on surfaces throughout the Tokamak in Figure 3. In our implementation, a 1D domain decomposition along the toroidal axis is used, which is significantly easier to program than a full 3D decomposition. In addition, along this axis the number of particles per processor remains relatively constant, which minimizes load imbalance. The most recent efforts have been in porting the code to the Cray T3D using Fortran77 and the PVM message passing library. Production runs on the T3D show a performance of approximately 14.4 MFlops per processor (10% of the theoretical peak of 150 MFlops, which is typical of these types applications). To test the scalability of the code, 12 runs were timed with different numbers of processors and problem sizes. To first order, the problem size is given by the total number of particles. The total amount of computer resource consumed is the time per step multiplied by the number of processors. Although the simulation is scalable, the resource consumption should be proportional to the problem size, which is shown by the straight line in Figure 4.