10
1-16-91 JS①

UCRL-ID-105650

Comprehensive Numerical Modelling of Tokamaks

R.H. Cohen
B.I. Cohen
P.F. Dubois

January 3, 1991

Lawrence
Livermore
National
Laboratory

This is an informal report intended primarily for internal or limited external distribution. The opinions and conclusions stated are those of the author and may or may not be those of the Laboratory.

Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (615) 576-8401, FTS 626-8401.

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.,
Springfield, VA 22161

<u>Price Code</u>	<u>Page Range</u>
A01	Microfiche
<u>Papercopy Prices</u>	
A02	1- 10
A03	11- 50
A04	51- 75
A05	76-100
A06	101-125
A07	126-150
A08	151-175
A09	176-200
A10	201-225
A11	226-250
A12	251-275
A13	276-300
A14	301-325
A15	326-350
A16	351-375
A17	376-400
A18	401-425
A19	426-450
A20	451-475
A21	476-500
A22	501-525
A23	526-550
A24	551-575
A25	576-600
A99	601 & UP

COMPREHENSIVE NUMERICAL MODELLING OF TOKAMAKS

R. H. Cohen, B. I. Cohen and P. F. Dubois
Lawrence Livermore National Laboratory

UCRL-ID--105650

(Draft, 1/3/91)

DE91 006219

ABSTRACT

We outline a plan for the development of a comprehensive numerical model of tokamaks. The model would consist of a suite of independent, communicating packages describing the various aspects of tokamak performance (core and edge transport coefficients and profiles, heating, fueling, magnetic configuration, etc.) as well as extensive diagnostics. These codes, which may run on different computers, would be flexibly linked by a user-friendly shell which would allow run-time specification of packages and generation of pre-and post-processing functions, including workstation-based visualization of output. One package in particular, the calculation of core transport coefficients via gyrokinetic particle simulation, will become practical on the scale required for comprehensive modelling only with the advent of teraFLOP computers. Incremental effort at LLNL would be focused on gyrokinetic simulation and development of the shell.

I. BACKGROUND

With tokamaks in the U.S., Europe and Japan on the threshold of achieving breakeven conditions, the tokamak is currently the most advanced concept for making fusion power. However, there is no comprehensive modelling tool that can be used to design and project the performance of future machines or new running conditions on present machines. Rather, tokamak predictions and analyses are based on a patchwork of independent physics codes which model, with varying accuracy, selected aspects of tokamak operation, empirical scaling laws, and systems codes which do not include detailed physics models. In particular, future-generation machines such as the Compact Ignition Tokamak (CIT) and the International Thermonuclear Experimental Reactor (ITER), designed on this basis, represent a significant extrapolation from present experimental parameters, and the projections are in directions in parameter space which might be far from optimal were the true scalings known. Largely, this situation is driven by the absence of a viable calculation of plasma transport coefficients; other significant factors include the lack of a robust, sufficiently comprehensive calculation of flows in the edge region and the present-day difficulty of performing three-dimensional resistive magnetohydrodynamics (MHD) calculations to accurately characterize transient effects (disruptions and sawteeth) on the magnetic configuration.

The likely advent of teraFLOP massively parallel computers in the next several years opens the prospect of changing this outlook. Gyrokinetic particle-simulation techniques, under development at LLNL and elsewhere, offer a promising avenue for calculation of tokamak transport. Gyrokinetics, in which the rapid gyration of charged particles in strong magnetic fields is analytically averaged and finite-orbit effects are included via cross-field drifts, is well-suited for simulation of the low-frequency (relative to the gyration frequency) turbulent phenomena generally thought to be responsible for tokamak transport. Present-day gyrokinetic codes, running with simplified physics and geometry and stretched parameters, tax the limits of present-generation CRAY computers to calculate turbulence in a slice of a tokamak. However, a three-order-of-magnitude increase in computing capability would greatly facilitate gyrokinetic simulation, enabling inclusion of much more complete physics. TeraFLOP computers would also facilitate calculation of resistive MHD phenomena via either gyrokinetic or fluid-based simulation. More generally, the prospect of advances in both physics modelling and computational capability could precipitate a

quantum leap in our ability to understand current experiments, and to predict and optimize the performance of future devices.

With such computing capability on the way, it is time for a community-wide effort to prepare for its arrival. The LLNL MFE Program proposes to contribute by (a) aggressively developing gyrokinetic simulation techniques, in order that the weakest link in the chain of tokamak physics computations be addressed as soon as possible, and (b) by developing a computational shell (framework) within which a suite of validated physics codes could then interconnect to produce a comprehensive tokamak modelling tool, and investigating the interface of appropriate, validated, standalone physics codes through such a shell. We note that the LLNL Computations Department is proposing (to the D.O.E. High-Performance Computing Program) a multi-institutional effort to develop a "3D Computational Physics Laboratory", which is to be a framework for doing high-performance distributed computing. This project (which will be discussed in a bit more detail below) may provide many of the computer science advances needed for (b). Gyrokinetic simulation and related theory are presently supported from programmatic sources at about the 3 1/2 FTE level. An additional 2 FTE effort is funded from internal Laboratory sources. Shell development is presently not supported. There is a reasonable likelihood of some additional internal (Director's Initiative) funding of both (a) and (b), particularly if there is a prospect of new external funding. Thus we offer, at present and possibly more in the future, internal leveraging of DOE funds.

II. Status, Prospects and Plans for Gyrokinetic Simulation

Gyrokinetic PIC simulations are being undertaken at Princeton, UCLA, LLNL, University of Texas, and Nagoya, Japan. There are 2½ and 3D electrostatic and electromagnetic codes in slab and cylindrical geometry. There is an electromagnetic toroidal formulation of gyrokinetics, valid to leading order in the ratio of the minor to major radius of the torus (inverse aspect ratio), but no code implementation as yet. The gyrokinetic codes are expensive to use on the CRAY 2; useful simulations of ion temperature gradient modes and trapped electron modes by Sydora at UCLA require ~100 hours. Partially linearized simulations (δf method; follows departure δf of distribution function from a prescribed background) greatly reduce statistical requirements, and useful 3D simulations with adiabatic electrons have been performed in less than an hour, although the partial linearization and the requirement of small $\delta f/f$ limit the physics. (The most important limitation for purposes of the following discussion is that δf techniques are inappropriate for simulating evolution of profiles on the transport timescale.) Statistical noise has been a severe problem in finite-beta electromagnetic simulations with fully kinetic electrons, but using a partially adiabatic or partially linearized electron response has resolved this difficulty. Simulations of ion temperature gradient modes and trapped electron modes have obtained saturated field amplitudes, estimates of the thermal transport, and credible frequency spectra. Because of the limited particle statistics or grid resolution, these simulations are crude at best. A 2½D electromagnetic code with partially adiabatic electrons has been ported to a Connection Machine CM-2 by the Princeton group, and the LLNL group in collaboration with Sydora at UCLA has also ported a 3D electrostatic code to a CM-2. LLNL is also working on porting gyrokinetics codes to the BBN-2000 parallel computer.

The "useful" simulations discussed in the preceding paragraph have a limited number of Fourier modes (mesh points) and (except for the δf runs) artificially large gradients. Assuming adequate statistics and separation of the fluctuating and equilibrium length scales, transport coefficients can be scaled analytically to realistic gradients (this scaling is assumed in the algorithm for the δf codes). A thousand-fold increase in computing speed would obviously facilitate

gyrokinetic simulations with realistic gradients, better statistics, and adequate number of modes. The run-time requirements for simulations of ion-temperature-gradient-driven and drift-wave turbulence with realistic parameters on present-day and teraFLOP computers is examined in some detail in the Appendix. The essential results are as follows. For explicit codes with fully kinetic electrons, there are three constraints that determine the cost of the simulation: the requirement that the thermal fluctuation amplitude per mode be less than 10% of the saturated amplitude for the most important modes sets a constraint on the total number of particles; the condition that the spectrum-integrated field energy in the noise be less than 10% of the saturated field energy sets a constraint on the number of particles per cell; and requiring that the explicit scheme be numerically stable sets a timestep constraint. The number of cells is set by requiring that the simulation resolve inverse wavenumbers equal to a gyroradius in the radial and poloidal directions, that in the direction parallel to the magnetic field the code resolve wavenumbers such that ion phase velocity is of order of the thermal speed (set by ion Landau damping), and that the entire core of the tokamak be simulated. Combining these requirements, along with a cost per timestep scaled from the best obtained on existing codes, leads to an estimate of 3×10^8 hr. on a teraFLOP machine to simulate an energy confinement time for a machine the size of the proposed Compact Ignition Tokamak (CIT)! A significant improvement can be obtained with implicit techniques, which should allow the code to run on the ion timescale; even then simulation of an energy confinement time requires about 20,000 hours, which is prohibitively long. However, simulation of the turbulence timescale for purposes of determining transport coefficients is more feasible; an implicit code with fully nonlinear electrons and ions could simulate the entire core of CIT in about 20 hrs. Furthermore, combining implicit and δf techniques could reduce this time by another two orders of magnitude, and simulating a toroidal annulus to determine local transport coefficients would gain at least another order of magnitude. Thus, gyrokinetics is potentially quite useful for determining transport coefficients locally or even in an entire machine.

III. The Comprehensive Model

This section outlines what we see as the ultimate scope of the physics and computational aspects of the comprehensive tokamak model. These are long-term goals, and will require a community-wide effort. In the next section we present some plausible near- and intermediate-term goals for the Livermore MFE Program's contribution.

(a) Physics Content

The physics ingredients required in a comprehensive tokamak model include all relevant transport processes in the core and the edge and the resultant self-consistent profiles; the magnetic configuration; r.f. deposition and the resulting heating and current drive; neutral-species transport and interaction with the plasma, for energetic (from neutral beams) as well as low-energy (from wall recycling, gas puffing and pellet fueling) neutrals; fusion nuclear physics; and equilibration of energetic ions (alphas or those resulting from neutral-beam injection). These will be discussed in more detail in the following paragraphs. In addition, the model should include extensive diagnostics, including the ability to simulate any conceivable experimental diagnostic. Finally, the model must be packaged in such a way that it is not only easy to run, but reasonably easy to modify.

As mentioned in Sec. II, barring major advances in gyrokinetic simulation techniques, running-time issues preclude the use of gyrokinetic codes to simulate a large (CIT- or ITER-class) tokamak for times long enough ($\sim$ energy confinement time) to evolve energy and density profiles, even on a teraFLOP computer. It would be similarly unfeasible to use particle-in-cell (PIC) simulation for

the scrapeoff-layer region, running for the time required to establish core profiles. On the other hand, gyrokinetic simulation for the purpose of determining transport coefficients will be quite reasonable; with δf techniques, one can even envision calculating transport coefficients or fluxes for the entire tokamak. Because of these considerations, we advocate, as the primary transport workhorse tools, a 1D transport code for the core, coupled to a 2D fluid simulation of the edge/scrapeoff-layer region; these codes would periodically fetch transport coefficients generated by 3D gyrokinetic simulations for the core, and 3D gyrokinetic, full ion-dynamics PIC, fluid, or hybrid fluid-particle codes in the edge. Also to be run iteratively with the transport codes are packages describing the other physics ingredients summarized in the preceding paragraph.

We note that the domain of the edge/scrapeoff-layer code extends from somewhat inside the separatrix out to the wall. Hence this code has the burden of calculating such diverse phenomena as power flow onto the divertor, the transition between L and H mode (assuming current qualitative notions linking the transition with edge phenomena are correct), and the effect of edge-localized modes (ELM's).

We propose two possible modes of interaction between the 1D and 2D transport codes and the transport-coefficient-generating codes (which, for the present discussion, we label as gyrokinetic, even for the edge): (1) the gyrokinetic code is used to generate an interpolatable database of transport coefficients (or fluxes); the database includes descriptions of the domain of validity. (For example, if transport coefficients were generated assuming the zero-order distributions are Maxwellian, the database includes a notation of that limitation.) The transport code queries the database, and logical tests are made to determine if the database is useful for the locale in question. If so, the transport coefficients (fluxes) are obtained; if not, the gyrokinetic code is run (in a toroidal annulus) to generate valid transport coefficients (fluxes); these are appended to the database. Note that the database has many parameters, and will ultimately contain many entries; maintenance and access to it will likely benefit from massively parallel computation. (2) the transport code is run iteratively with a gyrokinetic simulation of the entire machine (or gyrokinetics for the core, and whatever else is appropriate for the edge). The gyrokinetic simulation is, presumably, some version of a δf simulation, and is run only long enough to establish transport coefficients. It is rerun only when the profiles from the transport codes have changed significantly. Mode (2) does not require maintenance of a database, though the transport coefficients could be added to a database for future running under mode (1). Mode (2) would be expected to be more expensive (in terms of running time) than (1), but is likely to be able to handle a larger universe of phenomena: for example, if transport turns out to be significantly nonlocal, it would be more easily calculated by mode (2). (An illustration of this example is provided by experimental observations suggesting that, under fixed local conditions, core transport coefficients depend on whether the tokamak is in L or H mode, which in turn is thought to be governed by conditions in the edge.) Another (probably related) circumstance which would favor mode (2) is if the number of parameters or qualifiers required to specify a database entry is inordinately large.

In the near term, the principal benefit of gyrokinetic simulation of tokamak turbulence is likely to be increased understanding of the underlying physics, hopefully pointing the way to better analytic theories. This application will continue and become of even more value as more computational power becomes available. In the best of all worlds, the end result would be verified analytic expressions for transport coefficients valid in all conceivable regimes of tokamak operation. A more likely outcome is partial analytic success: transport coefficients valid over a limited regime of parameters and/or a reduction in the number of significant independent parameters. Such a result would still be of tremendous value in facilitating mode (1) of the preceding paragraph.

Both modes require advances in more than just computation speed: at present, no one uses particle codes in a production mode. A considerable amount of experimentation must typically be

done with time steps, mesh size, simulation domain, particle number and weights, etc., to obtain trustworthy answers. Production running of gyrokinetic codes as part of a larger suite would require development of bullet-proof criteria for these parameters, probably in conjunction with some artificial-intelligence procedures for assessing the trustworthiness of simulation results. Another area that may well not be completely straightforward is the interface of the core and edge transport codes.

An alternative to the iterative running of transport codes and particle codes is 3D fluid simulations running for the entire energy-confinement timescale. Compared to straight gyrokinetic simulation, one gains a significant speed advantage since the number of fluid quantities followed is typically small compared to the number particles per cell in the gyrokinetic code. A drawback of this approach is that it does not intrinsically include kinetic effects; these must be added in somewhat ad-hoc ways to the fluid equations. Kinetic effects are important in a number of the leading contenders for core transport; hence this restriction is significant. There are also potential difficulties associated with the inevitable truncation of the hierarchy of fluid moments. Hybrid fluid-particle codes represent one possible means of removing these restrictions, and so merit consideration; we plan to explore such codes as an alternative to pure gyrokinetic simulation for calculating transport coefficients.

Present-day "1 1/2 D" transport codes interface a 1D transport code (with, typically, extensive empirical transport models) to a 2D MHD equilibrium code. Typically, these codes have fixed-boundary equilibrium models and are thus unable to follow toroidally symmetric dynamical motion of the plasma. The TSC code is an exception; this code carefully models axisymmetric MHD motion on the *Alfvén* timescale, but the transport physics is quite limited. The comprehensive model will also require an interface to an MHD equilibrium calculation. We propose that the equilibrium be modelled quasi-statically, but with the ability to couple electrically to the poloidal-field (PF) coils to model the complete slow evolution of the plasma and thereby determine engineering constraints on the PF coils, control systems and other electrically connected systems, and to couple electrically to the plasma-facing components and control systems so as to model position control and the effects of off-normal performance (e.g., disruptions) on the structure. LLNL has developed an equilibrium code, TEQ, with much of the above capability; the remaining capabilities will be added.

In addition to the steady-state equilibrium, transient effects on the magnetic configuration, such as sawteeth and disruptions, must be included. This would be done with a 3D resistive MHD package, which could be a fluid-based code or possibly the gyrokinetic code; the equilibrium configuration and profiles would periodically be fed to the resistive MHD package, which would return a range of flux surfaces mixed by sawteeth and information about the occurrence and extent of any disruption.

Neutrals packages must be included to model gas, pellet and beam fueling and recycling of neutrals from the walls. The packages must describe penetration into the plasma (via either Monte-Carlo calculations or analytic models), as well as the various atomic physics processes involved with interaction with the plasma: ionization, excitation, and charge exchange. These processes lead to sources and sinks for the transport code and also affect the zero-order distribution functions used in the gyrokinetic transport-coefficient calculations.

Fusion reactions would be included either as subroutines of the transport codes or separate packages; alpha particles would be carried as a species (as would be various impurity ions) in the transport codes. Energy exchange between alphas and/or energetic ions from neutral-beam injection with the bulk plasma would be calculated in the transport code.

There are many different schemes for radio-frequency heating and current drive; hence the model must be provided with a menu of r.f. packages from which to choose. Electron-cyclotron

resonance heating (ECRH) and lower-hybrid packages would consist of ray-tracing codes with incorporated calculations of local heating and current drive. An ion-cyclotron heating package presumably requires a full-wave calculation to determine the power deposition. Packages for these and other heating and current-drive techniques would be added as needed.

While the comprehensive model would probably begin as a physics model, we expect that, as it matures, it will become of increasing interest to engineers, and hence would expect to see development and incorporation of engineering packages. Obvious candidates for early inclusion are magnetic stresses on conductors, erosion of wall materials by plasma bombardment and thermal stress, and nuclear activation.

We anticipate extensive diagnostics capabilities. Each physics package will include a database of significant quantities, any combination of which can be plotted in a format specified at execution, or, up to limits set by memory, stored for post processing. In particular, data would be processed to simulate experimental diagnostics of interest. Because the model involves a number of complex physics packages, several of which would be large standalone codes, the model will produce copious amounts of data which would tax the ability of a sizeable team of physicists to process and comprehend. A similar problem exists with large tokamak experiments, although the scope of the problem for an experiment is limited by a relatively modest number of operating diagnostics. We would seek to apply recently developed powerful tools for visualization and develop new intelligent techniques to sift through the data in search of interesting patterns and instructive ways of presenting them.

Note that the interfacing of the 1D core and 2D edge transport codes is basically uncharted territory. Questions concerning the sensitivity to the choice of the location of the common boundary, and the effect of the interface on timestep requirements, are among those that will surely require attention. This is an example of the general problem of interfacing validated standalone physics codes cited in Section I.

(b) Code Organization:

The comprehensive tokamak model will contain a number of substantial physics packages developed and capable of operating independently from the rest of the model. Different packages may run optimally on different computer architectures, and an individual package may well be run on different machines at different stages of its development. Consequently, the comprehensive tokamak model would (ultimately) operate in a distributed environment, with different packages running on possibly different machines, coordinated by and communicating through a shell controlled from a workstation. Since it is intended that the comprehensive model have a broad user community, including experimental and theoretical physicists as well as engineers, the workstation-based user interface should be as user-friendly as possible, which today would mean menu-driven specification of input/output options and control of the run. Since the user interface will be running on workstations at geographically diverse locations, and since other components of the distributed-computing network may well be geographically separated as well, high-speed data transmission and sophisticated network communications are essential.

It was noted earlier that LLNL's Computations Department is proposing to develop, in collaboration with other computer science centers, a "three-dimensional computational physics laboratory". This project would entail the development of a software and hardware to facilitate distributed, large-scale computing, with attention given to high-speed communications, visualization, optimization of overall performance (as a function of how modules are assigned to and scheduled on different machines), interfacing of codes, and efficient/intelligent storage, retrieval and processing of large amounts of data. The net result of such an effort should be a computing environment which would be well suited to the comprehensive tokamak model.

One particular aspect of this environment is the computational shell through which the individual packages would be linked to create the comprehensive model. This shell can be viewed as an application generation system which supplies the non-physics portions of the program. As we envision it, this system would have the following capabilities: (1) It would be portable to systems having certain characteristics; our strawman proposal for those requirements is: Unix (using a vanilla subset); X-windows; has Fortran 77; has C++ and C. (2) The user interface of a program created with this system would contain a programming language similar to that in LLNL's existing Basis system. In this language users could edit, plot, assign to, or inquire about the characteristics of, variables in the compiled program. In addition, the user could create procedures and variables. The system would support basic Fortran and C types, and offer a mechanism for the addition of user types and classes. Central to the proposal is the idea of attaching user-developed packages to a central core, as in Basis. Under the new system, such packages could be distributed over a network.

The rationale for such a shell is that giving full control of the program to its users allows maximum possible utility without recompilation and reloading. Since what we want to do with a program is the fastest changing part of any simulation, it is important that the application not be designed in the traditional way, in which every new use requires additions of more and more options and patches. Authors can be substantially more productive because non-physics issues are taken care of. This architecture also allows for a distribution of effort, in that different portions of the application can be developed independently without interfering with one another. Decisions about the nature of the distribution of the program over the hardware resources is made late and independently from the distribution of the pieces. The present Basis system provides a several of the capabilities required: it provides the facility for interfacing independent packages and provides flexible run-time pre- and post-processing (including plotting) of information. It is presently restricted to linking codes on a single computer, only has limited compatibility with C, is command- rather than menu-driven, and runs only on CRAY's and Sun workstations. These restrictions would be removed in the new system. We envision being able to produce such a system by combining our experience in creating Basis with new object-oriented technology, and adapting components of existing products where possible. We believe that this can be done in such a way as to hide the details from the standard Fortran programmer.

IV. Work Plan for LLNL MFE

The comprehensive model discussed in Sec. III is a monumental undertaking; several of the component pieces are major computational projects. As noted earlier, we propose to concentrate LLNL MFE resources on gyrokinetics and on the interface. Our base theory program also has a major commitment to scrapeoff-layer physics; hence we expect to have a key role in the implementation of the 2D edge plasma fluid code. Additionally we will be able to contribute MHD equilibrium and ECRH ray-tracing packages. Primary responsibility for several other packages would rest with other institutions, possibly as indicated in Fig. 1. Milestone goals can be assigned to individual packages as well as the overall model.

Within the gyrokinetics area, we would expect to have a 3D toroidal electrostatic code running on one or more parallel computers within the first year. Electromagnetic versions should be running within the following year (early FY93). These codes will not be running on teraFLOP machines, but will be useful research tools to learn about tokamak turbulence, about parallel computing, and about robustness and speed of algorithms.

In terms of the overall package, we propose to (a) initiate development of the new shell described above; (b) if the Computations Department's 3D Computations Laboratory proposal is

funded, interface with the Computations Department effort, assisting and assuring compatibility of that project with our needs, and (c) investigate the interaction of prototype standalone packages through a shell (which, initially, would be Basis). Note that LLL MFE's proposed shell-development effort does not include development of new visualization software, networking, communications, data storage and sorting, etc. We would be looking to the Computations Department's project, to NERSC, to commercial sources, and to other parts of the MFE community for these products.

A reasonable goal for our activity in the first year would be the demonstration of core and edge transport codes communicating with one another and possibly with an MHD equilibrium package; on a two-to-three-year timescale we would add interaction with transport-coefficient-generating packages based on one of the two interaction modes discussed in the preceding section. In conjunction with this latter goal, we would be researching the "smart data base" concepts required to make mode (1) of Sec. IIIa work. We expect that the base theory programs at General Atomics, ORNL, Princeton and LLNL will be able to contribute suitable core- and edge-transport codes. We would also expect to have a first version of the new shell working within a year, although with limited capabilities. The pace with which capabilities are added would depend on the degree of funding of both this project and parallel efforts such as that proposed by the LLNL Computations Department.

VI. Manpower/Budget (for LLNL MFE Contributions)

To perform the tasks specified for the LLNL MFE Program in this proposal in a timely fashion, we would require at least 11 FTE PhD's (2 for parallel computing, 1 1/2 for development of the toroidal electromagnetic code, 2 1/2 to do physics with the codes as they develop and improve algorithms for the physics, 2 1/2 to work on shell development and package interfaces for the comprehensive model, one to develop diagnostics, and 1 1/2 to do supporting transport/turbulence theory), mostly physicists but at least one computer scientist, plus a programmer. Our present effort consists of 5 1/2 FTE physicists (3 FTE effort on physics/algorithms/diagnostics, 1 1/2 on parallel computing, as a result of a recent shift in emphasis towards parallel computing, and 1 on supporting transport/turbulence theory). Thus at least an additional 5 1/2 FTE PhD's plus a programmer are required, for a total cost of about \$1110 K; we hope to obtain a portion of this from Laboratory Director's Initiative funding. (One or more of the new FTE's could be based in the Computational Physics Group at NERSC.) Because of inadequate funding of our base program and commitments to other activities, this increased effort will be possible only with new funding. The activities with highest priority for partial incremental funding are the toroidal electromagnetic gyrokinetic code and the package interface.

Fig. 1. Packages in Comprehensive Tokamak Model, and some candidates for contributing institutions.

APPENDIX: Projections for 3D Gyrokinetic Simulations of Tokamaks

In order to determine the feasibility of large-scale gyrokinetic simulation of tokamaks on the Cray 2 or a more powerful platform, e.g., a teraFLOP massively parallel computer, we should extrapolate from current 2 1/2 and 3 D simulations with adequate signal-to-noise characteristics. The criteria for adequate signal-to-noise are somewhat arbitrary, but conservative choices will be made so that we will not oversell gyrokinetics. Furthermore, in the spirit of conservatism, we will use a model with fully kinetic and nonlinear electrons. (This may be too conservative, because the use of partially linearized, i.e., δf , electrons or partially adiabatic electrons significantly relaxes the statistical constraints set by the electrons.) The two criteria for adequate signal-to-noise that we will invoke are that (1) the thermal fluctuation levels for the amplitudes of the important modes should be ten times smaller than the observed saturation amplitudes and (2) the total integrated electrostatic thermal fluctuation field energy should be ten times smaller than the observed integrated electric field energy at saturation. These criteria are not independent of one another, and they serve to set the total number of particles and the number of particles per cell. The cost of the simulation is the product of the number of particles times the cost of the timestep times the number of timesteps required to simulate the growth and saturation of the instability and a substantial fraction of an energy confinement time. The size of the timestep is set by stability and accuracy constraints inherent in the simulation method.

The criterion that the thermal fluctuation amplitude be ten times smaller than the saturation amplitude observed in the simulations is given roughly by

$$(e\phi_k/T)^{\text{noise}} \sim (1/k_{\perp}\rho)(1/N_p)^{1/2} < 0.1(e\phi_k/T) \sim (0.1)^{3/2}(1/k_{\perp}\rho)\rho/L \quad (\text{A1})$$

where $k_{\perp}$ is a typical wavenumber near the peak of the spectrum, N_p is the total number of particles, ρ is the ion Larmor radius, L is the plasma gradient scale length, and we recognize the right side as a slight reduction from the usual mixing length estimate. The result of Eq. (A1) is

$$N_p > 10^3(L/\rho)^2. \quad (\text{A2})$$

The criterion that the total electrostatic field energy due to thermal fluctuations be ten times smaller than the saturated field energy observed in simulations of unstable drift modes is

$$\begin{aligned} \frac{\langle (E^2/8\pi)^{\text{noise}} \rangle}{(3/2)n(T_i + T_e)} &= \frac{(2\pi)^{-3} \int d^3k (E_k^2/8\pi)^{\text{noise}}}{(3/2)n(T_i + T_e)} \sim \frac{10^{-3}(1 + \omega_{pi}^2/\omega_{ci}^2)^{-1}}{N_{\text{cell}}} \\ &\leq \frac{0.1E^2/8\pi}{(3/2)n(T_i + T_e)} \sim 10^{-3}(1 + \omega_{pi}^2/\omega_{ci}^2)^{-1} \rho^2/L^2 \end{aligned} \quad (\text{A3})$$

in the long wavelength limit for $\omega_{pe} \approx \omega_{ce}$, where N_{cell} is the number of particles per cell. The approximate equalities on both sides of the inequality in Eq. (A3) are based on our simulation experience in 2 1/2 D simulations of drift waves and ion temperature gradient modes, and the right side represents a reduced mixing length scaling. Equation (A3) determines a constraint on the number of particles per cell

$$N_{\text{cell}} > L^2/\rho^2. \quad (\text{A4})$$

Equations (A2) and (A4) are consistent with our simulation experience with kinetic electrons and gradients $\rho/L \sim 0.1 - 0.2$, where we used $\mathcal{O}(10^5)$ particles and 64-100 particles per cell. Sydora (UCLA) has used $\mathcal{O}(1 - 4 \times 10^6)$ particles and 20-60 particles per cell for gradients $\rho/L \sim 0.05 - 0.2$ in three dimensional kinetic electron simulations. A potentially important caveat in these projections is that our simulation data in 2 1/2 D comes from cases with kinetic electrons that have saturated in coherent structures and not from cases with incoherent turbulence.

The number of timesteps is set by consideration of the stability constraint for the numerical integration and by the instability and transport timescales. In an explicit electrostatic simulation $\omega_h \Delta t < 1$, where $\omega_h = (k_{\parallel}/k_{\perp})(m_i/m_e)^{1/2}\omega_{ci}$. For the modes of interest $(k_{\parallel}/k_{\perp})(L/\rho) \sim \mathcal{O}(1)$. The wave frequencies are set by the ion diamagnetic drift frequency $\omega_i^* = k_y v_i^* = k_y \rho(\rho/L)\omega_{ci}$. The transport time scale of interest is the energy confinement time, which is of order 10^3 to 10^4 wave periods for medium to large tokamaks.

We determine the required number of mesh points from the requirement that we resolve inverse perpendicular wavenumbers equal to a gyroradius transverse to the magnetic field and parallel wavenumbers up to a value such that ion Landau damping becomes large. We assume a "smart" coordinate system which follows equilibrium field lines; then the required number of cells transverse to the field is about $\pi(L/\rho)^2$, while the number of cells along the field is $2\pi R/L$.

For a moderately large tokamak with $\rho/L = 0.01$ and $R/L = 4$ and hence a grid of $100 \times 314 \times 25$, we need 8×10^5 cells and 8×10^9 particles ($N_{\text{cell}} = 10^4$) to satisfy Eqs.(A2) and (A4). In contrast, current 2D and 3D simulations have addressed systems with $\rho/L \sim 0.1$ using $N_p \sim 10^5 - 10^6$ and $N_{\text{cell}} \sim 40 - 100$, values which are in rough agreement with Eqs.(A2) and (A4). For the moderately large tokamak, $\omega_i^* \Delta t_e = \mathcal{O}(10^{-3})$ for $k_y \rho = 0.1$. To run the simulation to saturation of the turbulence, $\omega_i^* \tau = \mathcal{O}(10)$, requires $10^4 \Delta t_e$. With our vectorized subcycled code, $\Delta t_e = 20 \times 10^{-6}$ sec/particle δt on the Cray 2; and the simulation will consume $8 \times 10^9 \times 10^4 \times 20 \times 10^{-6}$ sec $= 1.6 \times 10^8$ sec $\sim 4 \times 10^5$ hr. Simulation of a transport time scale would require approximately 4×10^8 hr. A teraFLOP computer is 2×10^3 faster than the Cray 2 and would require 200 hr to simulate saturation of the turbulence and 10^5 hr to simulate the transport timescale. The simulation of a tokamak as big as CIT, $\rho/L \sim 0.003$, would require $100\times$ more particles, and $100\times$ more simulation time for the saturation timescale or $1000\times$ more simulation time for the transport timescale.

A significantly improved scaling of the simulation running time is obtained by using either adiabatic electrons (unacceptable for some forms of turbulence and questionably acceptable for others), partially adiabatic electrons (kinetically follow only electrons expected to deviate from adiabatic; the specific implementation depends on the type of turbulence under study); partially linearized electrons and ions (δf method; not useful for runs on the energy-confinement timescale), or semi-implicit orbit-averaged electrons. The last option may well be the most attractive when implemented because the electrons are fully nonlinear and kinetic, but their statistical requirement is relaxed; and the simulation statistics and timestep constraints are set by the ions. The timestep constraint is $k_{\parallel} v_i \Delta t \sim \omega_i^* \Delta t < 1$. The timestep Δt here is the ion and field solve timestep; the electron timestep is smaller by $(m_e/m_i)^{1/2}$. Thus, $\omega_i^* \Delta t = \mathcal{O}(0.1)$ for $k_y \rho = 0.1$; and the timestep Δt is bigger by 10^2

than for the explicit algorithm with kinetic electrons. Furthermore, averaging the electron response and filtering the high frequency ω_h mode leads to a large reduction in the thermal noise:

$$(e\phi_k/T)^{\text{noise}} \sim 1/N_p^{1/2} \quad (\text{A5})$$

and

$$\frac{\langle (E^2/8\pi)^{\text{noise}} \rangle}{(3/2)n(T_i + T_e)} \sim 10^{-3} \frac{\langle k_{\perp}^2 \rho^2 \rangle (1 + \omega_{pi}^2/\omega_{ci}^2)^{-1}}{N_{\text{cell}}} \quad (\text{A6})$$

This results in a reduction of $\langle k_{\perp}^2 \rho^2 \rangle$ in the requirements on N_p and N_{cell} , which is $\mathcal{O}(10^{-2})$ for the applications in which we are interested. Therefore, the combined reduction in running time taking into account algorithm improvement is a factor of 10^4 , which exceeds the improvement obtained by going to a teraFLOP computer! Thus, with algorithm improvements, running to saturation for a system with $\rho/L = 0.01$ would require 40 hr on the Cray 2 and ~ 1 min on a teraFLOP computer. Simulating the transport time scale would require 4×10^4 hr on the Cray 2 and 20 hr on a teraFLOP computer. A CIT simulation would require about 4000 hrs on the Cray 2 or 20 hrs on a teraFLOP computer for the saturation timescale, and about 10^3 times longer for the transport timescale. A combination of semi-implicit, orbit-averaged electrons and δf techniques for both species could result in a further reduction of running times by order $\delta f/f$ for simulations on the turbulence-saturation timescale; for $\delta f/f \sim \rho/L$, the running time would be around 10 min for the CIT simulation on the transport timescale using the teraflop computer. Note that running times can be reduced by at least another order of magnitude to simulate local transport coefficients in a toroidal annulus (rather than the entire tokamak). Finally, we comment that hybrid fluid- δf simulations, in which only departures from *perturbed* Maxwellians are included in δf , could further reduce the running time.

In finite-beta, electromagnetic simulations, the Courant condition on the shear Alfvén wave replaces the ω_h criterion. The ratio of v_e to $v_{\text{Alfvén}}$ is

$$v_e/v_{\text{Alfvén}} = (\beta m_i/m_e)^{1/2} \sim \mathcal{O}(10)$$

for typical tokamak parameters. Of course, $v_e/v_i \sim \mathcal{O}(43)$ for hydrogen and $T_e = T_i$. Thus, the timestep constraint in an explicit finite-beta gyrokinetic code with kinetic electrons would be set by the electron parallel Courant condition, which is essentially the same as the ω_h condition (for $k_{\perp} \rho \leq 1$). With an adiabatic or partially adiabatic electron response (which removes all or most of the electron noise), the Alfvén Courant condition would set the timestep, which would be about a factor of 4 more severe than the ion parallel Courant condition. An implicit or semi-implicit field solution would allow the use of the bigger timestep set by resolving the ion motion. In addition, combining the use of the δf method or orbit averaging with implicitness would recover electron kinetic effects, but still allow the use of the ion timestep. Analysis and simulations by W. Lee at Princeton indicate that the constraints set by thermal noise in the finite-beta simulations are determined by the electrostatic noise whose amplitude is roughly equal to the estimates in the preceding discussion. Use of the Alfvén-Courant-condition-determined timestep and the numbers of particles set by the noise constraints given above leads to projections of code running

times that are much the same as those that we have already described for the semi-implicit, orbit-averaged or partially linearized electrostatic algorithm.

END

DATE FILMED

02 / 06 / 91