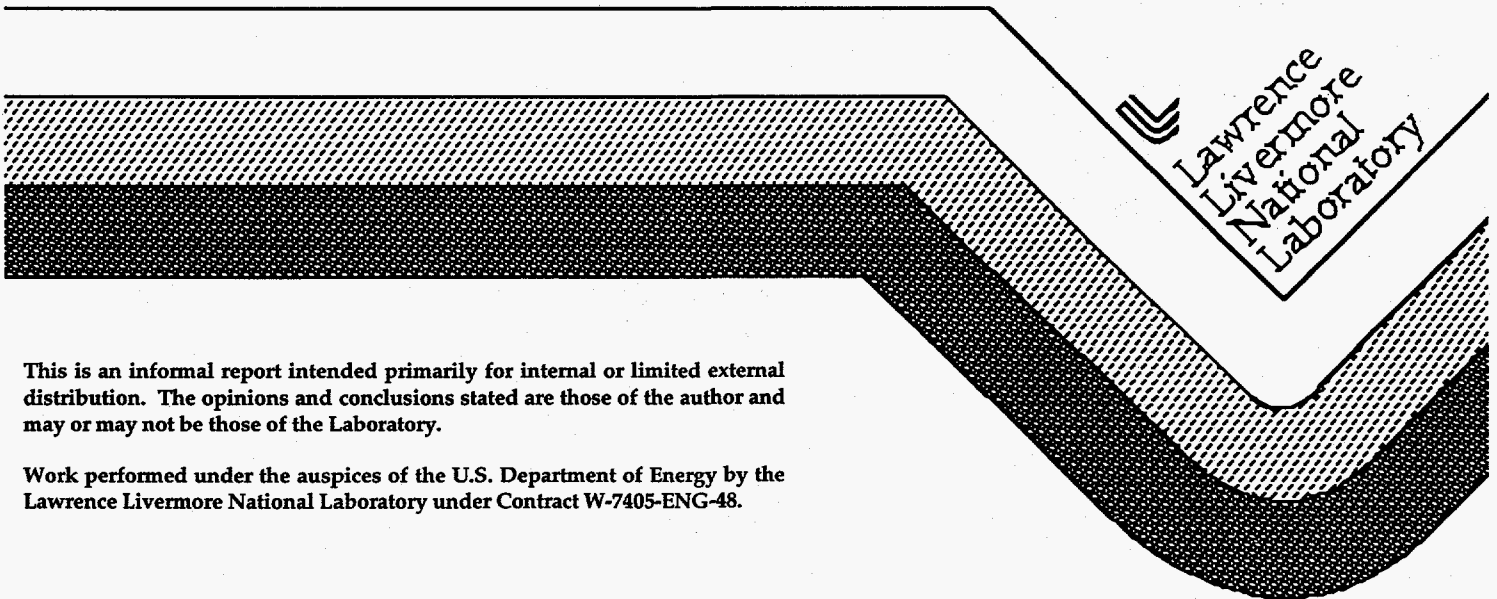


**Office of Fusion Energy Computational Review**

B.I. Cohen  
R.H. Cohen  
J.A. Byers  
J.A. Crotinger  
A.M. Dimits  
T.B. Kaiser  
A.E. Koniges  
N. Mattor  
L.D. Pearlstein  
T.D. Rognlien  
D.D. Ryutov  
G.R. Smith  
A. Tarditi  
X.Q. Xu

RECEIVED  
APR 18 1996  
OSTI

**March 6, 1996**



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.

#### 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 own 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

## OFFICE OF FUSION ENERGY COMPUTATIONAL REVIEW

B.I. Cohen, R.H. Cohen, J.A. Byers, J.A. Crotinger,  
A.M. Dimits, T.B. Kaiser, A.E. Koniges, N. Mattor,  
L.D. Pearlstein, T.D. Rognlien, D.D. Ryutov,  
G.R. Smith, A. Tarditi, and X.Q. Xu

**Magnetic Fusion Energy Theory and Computations Program  
Lawrence Livermore National Laboratory, University of California**

March 6, 1996

### 1. Computational Areas

The LLNL MFE Theory and Computations Program supports computational efforts in the following areas:

- (i) Magnetohydrodynamic equilibrium and stability
- (ii) Fluid and kinetic edge plasma simulation and modeling
- (iii) Kinetic and fluid core turbulent transport simulation
- (iv) Comprehensive tokamak modeling (CORSICA Project) - transport, MHD equilibrium and stability, edge physics, heating, turbulent transport, etc.
- (v) Other: ECRH ray tracing, reflectometry, plasma processing.

### 2. Code Development Decision Process

The decision process for code development is driven by both local and national MFE program needs as interpreted within the context of the LLNL MFE Theory and Computations Program research agenda. Specific research areas are identified in our Field Work Proposal in consultation with OFE program management in Washington and MFE program leadership at LLNL. The physics questions are defined within particular areas by the senior physicists in our Theory Program, and teams of physicists are established to address the problems (matched by interest, experience, and availability). The physics team then makes an assessment as to whether the physics problem is best suited to analytical, numerical or a mix of analytical and numerical treatments. If a significant numerical calculation is needed, we first survey what codes are suitable for the application of interest and already available in our group, elsewhere at LLNL, or from friendly groups at other institutions and possible collaborators. If nothing available is appropriate, we then build a new code. Of course, the necessary modifications or code extensions may be significant enough to be tantamount to building an entirely new code. It is sometimes more difficult to modify an existing code than it is to start over using just parts of the pre-existing code. Almost all of our code development in the last ten years has been to extend (sometimes profoundly) our own or imported pre-existing codes. We have undertaken major rewrites of existing codes to accommodate significant physics changes or extensions, or to allow the code to run efficiently on a massively parallel computer. This has often involved collaborations with other research groups.

Support of old codes is terminated when user demand falls sufficiently low. Termination of support typically means carefully archiving the source code, examples of data decks, and other material needed to resurrect the code for future use.

### 3. LLNL MFE Theorists

The following career, term and post-doctoral physicists work in the MFE Theory and Computations Program. Work for OFE accounts for approximately one-half of our funding. The percentages following the names indicate approximately how much of each physicist's time is spent on code development.

Ron Cohen (10%)  
 Bruce Cohen (50%)  
 Don Pearlstein (75%)  
 Jack Byers (75%)  
 Jim Crotinger (75%)  
 Gary Smith (75%)  
 Lynda LoDestro (75%)  
 Nathan Mattor (10%)  
 Xuegiao Xu (75%)  
 Alfonso Tarditi (75%)  
 Andris Dimits (75%)  
 Dimitri Ryutov (0%)  
 Alice Koniges (60%)  
 Tom Kaiser (75%)  
 Tom Rognlien (50%)

### 4. Code Inventory

An inventory of codes is given in the following table. For simplicity, closely related codes have been grouped under the name of the principal code in the family; and only major codes have been listed. More detailed information on each code is given following the table.

<u>Code Name</u>	<u>Date 1st Use</u>	<u>Code Authors</u>	<u>Frequency of Use</u>	<u>Application</u>
CORSICA	1992-96	J. Crotinger, S. Haney, D. Pearlstein, A. Tarditi, A. Shestakov, G. Smith, L. LoDestro, X. Xu, P. Brown	Heavily used	Comprehensive tokamak modeling
MCPAT/ORBIT	1980	T. Rognlien	Heavy before 1992	Transport/heating
UEDGE	1992-93	D. Knoll, T. Rognlien, G. Smith, et al.	Heavily used	DIII-D and ITER edge and divertor modeling
TEQ	1987	D. Pearlstein, S. Haney, R. Bulmer L. LoDestro	Frequent short runs	MHD equil. and stability for TPX, DIII-D, ITER, spheromaks



HAWC/HWDIA 1989-91	J. Crotinger A. Koniges, et al.	Heavily used '89-92	2D fluid drift- wave turbulence simulation
FLUEDGE 1995	X.Q. Xu	Frequent multi- hour runs on workstations	2D and 3D edge and SOL fluid turbulence simulation
PW1 1995	X.Q. Xu, J. Byers, O. Batishchev	Frequent multi-hour runs	Kinetic SOL plasma simulation
FOREV 1990	H. Wuerz, B. Bazelev, I. Landman	Sporadically heavy	Models divertor target erosion
CRETIN 1990s	H. Scott H. Dalhed	Sporadic	non-LTE radiation transfer
Gyrokinetics 1990-96	A. Dimits, B. Cohen, J. Byers, T. Williams	Heavily used '90-96	Kinetic simulation of core turbulent transport.
Reflectometry 1995-96	B. Cohen, T. Kaiser	Sporadically heavy	Full-wave modeling of diagnostic reflectometry
TORCH 1986	G. Smith, A. Kritz	Sporadically heavy	WKB ray tracing for ECRH study
BASIS 1988-1996	P. Dubois, et al.	Heavily used	ICF and MFE code dev. and production environ.

## 5-7. Code Descriptions

### CORSICA

Purpose: Comprehensive Simulation and Transport Modeling

Developed under LDRD at LLNL, FY93-96.

Authorship: James A. Crotinger, Ronald H. Cohen (PI's)

Corsica Group (current and former members):

J. Crotinger, R. Cohen, L.D. Pearlstein, L. LoDestro,  
S. Haney, P. Brown, G. Smith, T. Kaiser, A. Tarditi,  
T. Rognlien, A. Shestakov, X.Q. Xu

The Corsica Project is an internally funded (LDRD) project to develop algorithms for coupling disparate scale physical processes and to implement these algorithms in a prototype comprehensive simulation code for toroidal magnetic fusion devices. This comprehensive code, CORSICA, has been developed in three stages:

CORSICA 1 - coupled the free-boundary ideal MHD equilibrium code, TEQ, to a toroidal core transport code and to circuit equations for the external poloidal field coil and passive conductors. In modeling jargon, this is a "Free Boundary 1-1/2 D Transport Code"

CORSICA 2 - this release added the capability to couple the physics (currently with fixed magnetic geometry) in the CORSICA 1 code to the UEDGE 2D edge/SOL code.

CORSICA 3 - this release will include the capability to couple, in a distributed fashion, to copies of the IFS/PPPL GRYFFIN gyro-fluid code, with a copy of GRYFFIN running at each flux surface.

This project began in FY93 and is now in its fourth and final year of LDRD funding.

It is difficult to classify CORSICA because a comprehensive simulation code of this sort has many possible uses. Certainly the most obvious use is in modeling experiments and new designs. Although the code is designed primarily to be a predictive code, it is very flexible and has also been used in semi-interpretive studies. We also see CORSICA as a numerical testbed for both new theoretical ideas and for new code modules. It was designed to be easily extensible with the idea that one could try out new modules for certain physical processes and see how these modules interact with the rest of the system. The code is also very flexible, so such testing can be done with full simulation capabilities turned on, or with only a couple of modules interacting.

In order to attain this flexibility, CORSICA makes extensive use of the Basis system for developing scientific software. Basis provides an easy-to-use programmable interface, online help, portable binary save files, a flexible history package, and interactive graphics. CORSICA has nearly 13,000 lines of "scripts" written in the Basis language.

CORSICA's TEQ ideal MHD equilibrium module (which is covered in detail in a separate entry) can write EQDSK format equilibrium files. However, for most purposes CORSICA makes use of the Basis binary save file format. Other Basis codes can easily read this data, and conversion routines can be written to convert this data to other file formats, if needed.

While CORSICA 2 and 3 are still in the developmental stage, CORSICA 1 has been used by modelers and experimentalist since FY94:

The first physics applications of CORSICA 1 were to the ITER design. In FY94 we performed studies of shutdown and of the vertical control of ELMing plasmas. In mid-94 and early FY95 we applied the code to the analysis of ITER variants and reversed shear scenarios. Modifications to the Rebut-Lallia-Watkins transport model were also investigated,

including the impact of these changes on the machine's ignition characteristics in both H and L modes of operation. CORSICA has also been used to make accurate calculations of MHD equilibria using PRETOR transport profiles. The resulting equilibria were used by several teams to assess MHD stability. We have also collaborated with Alan Turnbull at GA, modeling the MHD stability of ITER-like DIII-D shots. Using the TEQ portion of CORSICA, we reconstructed EFIT equilibria and evaluated ballooning stability. These equilibria were then used in GATO to evaluate low-n stability. We also examined the sensitivity of these calculations to perturbations in the profiles and beta by making such perturbations in CORSICA and then re-evaluating stability. The code is currently being used at LLNL and General Atomics (GA) as part of an ITER task to study vertical and shape control for the ITER design. This work is being done in collaboration with Dave Humphreys at GA and with researchers at the ITER Naka Joint Work Site.

CORSICA is also being used as part of the ITER disruption modeling effort. It has been used to model the transport of impurities into the core after the thermal quench. The transport model is being extended to follow the plasma evolution as the plasma hits the wall, thus allowing the code to simulate the complete post-thermal-quench axisymmetric evolution of the core plasma.

CORSICA is also being used by LLNL and GA researchers to model the DIII-D experiment. In mid-FY94, we began working with Tom Casper and Barry Stallard (from LLNL) to simulate DIII-D advanced tokamak (AT) scenarios. Currently, they are using the code to simulate negative central shear (NCS) experiments, and to attempt to develop scenarios in which the NCS region is actively controlled using current drive. Also, as part of the ITER control task mentioned above, we have been collaborating with Dave Humphreys at GA to model DIII-D's control system. We have also recently begun work on studying NCS experiments in TFTR.

Finally, LLNL's spheromak working group has been using CORSICA to study spheromak geometry and is planning to use the code to follow the axisymmetric evolution of spheromaks.

Furthermore, we are just starting to do applications work with CORSICA 2, and plan to use it for both DIII-D modeling and for work on ITER divertor modeling.

Overall, CORSICA is a central piece of the LLNL MFE Theory Group's effort. We are currently writing a proposal to start a nationwide comprehensive computing initiative in which CORSICA would serve as a starting point on which to build.

The CORSICA core transport module, the Basis framework, and the coupling modules and associated Basis scripts were all developed at LLNL. Some of the modules, such as TEQ and UEDGE, have long histories that may include collaborations with people at other institutions. The GRYFFIN code that is being used for CORSICA 3 was written at PPPL.

By the end of FY96 we will have invested approximately 18 man-years in the development, testing, and initial application (within the Corsica group) of the CORSICA code. This excludes the initial development costs of the major modules, such as TEQ and UEDGE, although it does include some improvements made to these modules in order to support the CORSICA effort.

#### CODE MODEL, ALGORITHM, ETC.

The TEQ ideal MHD equilibrium module and the core transport module are the foundation on which the rest of the CORSICA system is built. TEQ is discussed as a separate item and will not be discussed here. The core transport package solves the flux-surface averaged transport equations for the macroscopic quantities, such as the particle densities, electron and ion temperatures, and the magnetic flux. We employ a finite-element solution technique for discretizing the spatial dimensions (both B-spline and linear elements are available) and we use a simple semi-implicit, first order time stepping algorithm. The time stepping algorithm employs a functional (or "fixed-point") iteration scheme to converge the nonlinear terms at the advanced timestep. While this is not as efficient as a Newton scheme, the major convergence bottleneck is caused by the coupling to the equilibrium, which is also a fixed-point iteration and which is considerably more expensive than that transport module. Thus there appears to be little to be gained from going to a Newton method.

The ideal MHD equilibrium is evolved quasistatically subject to the transport of flux and energy. The coupling between these turns out to be tricky. We have devised a feedback algorithm that attempts to ensure that certain consistency constraints are satisfied as the two modules are iterated to convergence.

CORSICA 2 can optionally couple to the UEDGE code. The current version couples a single density species and the electron and ion temperatures, and treats the edge quasi-statically (UEDGE is used to find steady-state solutions). The coupled fields and their fluxes must match at the core-edge interface. The matching is achieved by choosing a common edge boundary value, taking the timestep with each code, and calculating the difference between the flux at the interface as calculated by each code. A Newton iteration is then used to try to find the common boundary value that forces the flux difference to zero. We are working to extend this coupling to additional fields (impurity species, toroidal rotation, neutral gas, ...) and to allow for transients in the edge, which cannot be followed quasistatically.

CORSICA 3 couples attempt to solve the core transport equations using particle and heat fluxes calculated by the turbulence simulation code GRYFFIN. This is a very difficult task because the fluxes are noisy, depend nonlinearly on the fields and their gradients, are not necessarily diagonal, and are possibly non-local. The coupling algorithm operates by calculating average fluxes, using these to derive average diffusion coefficients, and advancing the transport equations. After each iteration of this scheme, the turbulence simulation code is stepped for a short amount of time with a new background profile. If this system converges, the result is a transport-scale timestep that is fully implicit.

The basic core transport module is written entirely in C++. Some of the source and transport modules are written in Fortran. The finite element package is written in C++. The time advance and much of the less-computational coding is written in the Basis script language.

As this code includes many pieces of physics, it is difficult to give a detailed list of physical assumptions. The primary assumption that provides the organizing theme for the comprehensive code effort is that core transport is approximately 1-D and that it occurs on a much slower timescale than many other plasma phenomena. The goal, then, is to design a system in which the effects of the fast scale physics are properly included in the slow-scale equations, and yet the slow scale equations can be stepped on their natural scale.

#### CODE RUN-TIME, TESTING, ETC.

Again, because of the flexibility of the code, this is somewhat akin to asking for the typical running time of Mathematica. That requires that there is a "typical" run, and there is not. But we'll do our best...

In the CORSICA 1 work on control system design we typically only evolve the free boundary equilibrium, the flux equation, and the circuit equations. Most of this work is done semi-interactively on workstations. Full simulations of vertical events and full transport simulations can take up to an hour of time on the C90, depending on the modules used.

Our experience with CORSICA 2 is still fairly limited. UEDGE is a rather expensive code, and we are calling it several times each timestep. The L-H runs that were presented at PET required about 2.5 hours of C90 CPU time, and this was with a fairly course UEDGE mesh. We have since identified several areas where work is being done unnecessarily and are working to optimize the overall code.

We have even less experience with the coupling to GRYFFIN. Our initial goal will be to use CORSICA 3 to find self-consistent steady state solutions of the coupled system. Our experience with prototype codes indicates that this can typically be done for 2-5 times the cost of a single saturated run of the stand-alone local turbulence simulation code. We have to run one of these codes at each flux surface, and will probably begin with 16-32 flux surfaces. Thus we estimate that the time requirement will be 30-150 times the amount of time required to do a single run with the GRYFFIN code. That, in turn, depends on the physical parameters at the flux surface and on the resolution of the turbulence simulation. Well converged runs probably require around 1 hour of C90 cpu time. But the GRYFFINs running on different flux surfaces are independent while they are taking their steps, and thus can be run in parallel.

On benchmarking, we have done comparisons of the core transport module to the PRETOR code. Furthermore, the code is being compared with experiment as explained in item 4. Our funding to date has required us to do a minimal amount of "programmatic" work with the code and to concentrate on algorithms. This certainly has left some holes both in our testing and in the list of physics modules that are currently in the code.

#### MCPAT/ORBIT

CODE NAME:	MCPAT and ORBIT
CODE AUTHOR:	Thomas D. Rognlien
CODE PURPOSE:	Test particle simulations to study transport and wave heating effects
FIRST USE:	1980
LANGUAGE:	Fortran

Developed - 1980-3; effort 1.5 FTE, Last used, 1992

Test-particle codes (1-D spatially; 2-v or 3-v for velocity) including Coulomb collisions (MCPAT) and prescribed RF fields [MCPAT (2-V) / ORPAT (3-v)]. MCPAT models the RF electromagnetic field interaction using an efficient reduced-equation description that only accurately follows resonant wave-particle interaction; ORPAT uses a second-order leap-frog scheme - both allow relativistic particle motion.

Used to study parallel plasma transport in transition regime between short and long mean-free paths for mirror devices and in the tokamak scrape-off layer region. Also used to study RF wave-particle heating for intense fields where the quasilinear diffusive models break down (FEL heating, for example).

### UEDGE

CODE NAME: UEDGE  
CODE AUTHORS: T. Rognlien, D. Knoll, J. Milovich, M. Rensink, G. Smith  
CODE PURPOSE: Comprehensive edge and divertor plasma simulation  
FIRST USE: 1993 at LLNL  
LANGUAGES: Fortran, MPPL, C

### Code Development

UEDGE came into existence in 1993 with the merger of LLNL's LEDGE and INEL/Sandia NEWEDGE code. LEDGE development began in 1990-91 with Rognlien, Milovich, and Rensink at the level of 2 FTE's for a two year period. The NEWEDGE development began several years earlier as Knoll's PhD thesis project at Univ. of New Mexico. Since 1993, Gary Smith and Peter Brown replaced Milovich in the LLNL effort, which is at the 1.5 FTE level per year (excluding application).

UEDGE has been a multi-institutional effort with Knoll (INEL) providing algorithm support, early impurity models and Navier-Stokes neutrals (0.25 FTE/year); Hirshman (ORNL) worked with us to incorporate his FMOMBAL module for collisional parallel friction for impurities (0.15 FTE); Wising and Krasheninnikov (MIT) helped formulate and implement Navier-Stokes neutrals (0.5 FTE); Karney and Stotler (PPPL) are aiding in coupling the DEGAS-2 Monte Carlo code and Coster (Garching) has provided the EIRENE Monte Carlo code (0.2 FTE); and we use the impurity atomic rate table generator developed by Braams (NYU). (Note: FTE's are estimates of times related specifically to UEDGE coupling, not development of the other codes)

### Code Usage

UEDGE is used extensively to model experiments in DIII-D and, to a lesser degree, Alcator C-Mod and TCV in Lusanne. It has also been used for divertor design studies of DIII-D radiative divertor upgrade, ITER, and TPX.

### Code Description

- Solves 2-D plasmas fluid transport equations in edge/SOL poloidal plane for multiple ion densities and parallel velocities, a common ion temperature,



electron temperature, neutral gas density and parallel velocity, and the electrostatic potential.

- Multi-charge state impurities are treated using Hirshman's FMOMBAL routine for non-trace collisional interaction along the B-field.
- Plasma transport is classical along B with flux limits, and anomalous perpendicular to B; classical parallel currents and ExB drifts included.
- Electrostatic potential calculated inside and outside magnetic separatrix using toroidal momentum balance and parallel Ohm's law.
- Neutral transport is via fluid model with parallel inertia (Navier-Stokes); simple coupling to Monte Carlo neutrals achieved and being upgraded.
- Finite-volume differencing scheme using flux coordinates from MHD equilibrium. Nonorthogonal mesh capability allows fitting oddly shaped divertor structures into the grid.
- Full system of equations solved implicitly with a numerical Jacobian; Krylov method employed with a preconditioning matrix using iterative algorithms together with reordering, and row and column scaling.
- Can perform Newton-type iterations to steady-state or time-dependent calculations for transients and to test stability of steady-state solutions.

The biggest uncertainty in the model is the anomalous radial diffusion coefficients which are most often chosen to fit experimental profiles (interpretive mode). Theory-based models of radial diffusion coefficients for the conducting-wall mode have been implemented; this predictive capability is at a preliminary stage. Other areas needing improvement are an efficient and accurate treatment of neutral gas including both kinetic and fluid regimes, together with a description of molecules and radiation transport; kinetic plasma models beyond flux limits; and coupling to core transport - see CORSICA-2. For numerical techniques, improvements are needed for adaptive mesh capability, improved efficiency such as multi-grid, domain decomposition, and parallelization.

#### Comparison with other codes and validation with experiment

At the early stage of code development (1991), we did some benchmarking with the B2 code for a hydrogenic plasma only. With the merger of LEDGE and NEWEDGE (Knoll) into UEDGE in (1992-3), we did a detailed benchmark between LEDGE and NEWEDGE. Limited benchmarking with B2/EIRENE (Garching) and EDGE-2D (JET) has been done as part of the ITER divertor design modeling, and this is continuing - differences appear related to the neutrals models. A favorable comparison for our Navier-Stokes neutrals has been done in slab geometry between the public version of UEDGE, capable of both slab geometry and toroidal geometry with only parallel neutral inertia, and a purely slab version of UEDGE where inertia is retained for all three velocity components of the neutrals (Knoll). Finally, a good comparison was found for a simple slab problem for diffusive neutrals only between UEDGE and a finite element code from LANL (Kuprat, Glasser).

Extensive UEDGE validation studies have been done for edge and divertor data from DIII-D (Porter, Rensink, Fenstermacher, Jong) and a more limited set from Alcator

C-Mod (Wising). Here the procedure is to adjust the anomalous diffusion coefficients to fit density and temperature profiles at the midplane and then compare profiles of density, temperature, and heat flux at the divertor plates. Agreement is reasonable, typically within a factor of two, but discrepancies remain in D-Alpha radiation and sometimes in particle flux.

References for experimental validation:

- Porter, et al., Contrib. Plasma Phys., Vol. 34, 454 (1994).  
Porter, et al., Plasma Physics and Controlled Nuclear Fusion  
Research (Proc. 15th Int. Conf. Plasma Phys. and Controlled Fusion Res.,  
Seville, Spain, (to be pub., IAEA, Vienna, 1995), IAEA-CN-60/D-P-I-1.  
Fenstermacher, et al. J. Nucl. Mat., Vol. 220-222, 330 (1995).  
Porter, et al., Phys. Plasmas, to be published (1996).  
Wising, et al., Contrib. Plasma Phys., Vol. 36, to be published (1996).

FLUEDGE

Code name: FLUEDGE  
Author: Xueqiao Xu  
Code Description: 3D fluid turbulence code in edge+SOL plasmas  
First Use: 1995  
Language: C + Fortran subroutines

Summary of Code

In this code, we develop theory and 3D simulations of scrape-off-layer (SOL) turbulence in detached plasmas. The detached plasmas are characterized as rapid variation of the equilibrium plasma profiles along the magnetic field line between the x-point and the divertor. Our analysis shows that the steep parallel gradients near the divertor might play the same role as surface impedance in the plasma sheath and thus drive a conducting-wall-like mode [1,2] or an axial shear mode [3]. The 3D Fluid Simulations of edge turbulence in the electrostatic limit are performed using a four-field model of the reduced Braginskii equations in a model x-point geometry, including drives for

- (1) conducting-wall modes [4],
- (2) Kelvin-Holmhotz modes,
- (3) curvature-driven modes,
- (4) radial gradient of parallel ion velocity driven mode and
- (5) dissipative axial shear modes.

By varying the axial background electron temperature, density and ion parallel velocity profiles in the code, we systematically investigate the change of turbulence properties. For uniform axial equilibrium profiles, we have obtained a flute mode structure along the field line. Fluctuating amplitudes for  $\epsilon\phi/Te > n/n0 > Te/Te0=0.2$ , and  $\chi_e = D_e = 2 \text{ meter}^{**2}/\text{second}$  are comparable to 2D results and experimental results. We also find  $V_i/C_s = Te/Te0$  and the momentum diffusivity  $\chi_m=De = 2 \text{ meter}^{**2}/\text{second}$ ; both are peaked at the sheath entrances. In a detached plasma with  $Te0/Ted=50$  and  $Ne0/Ned=0.2$ , and without magnetic shear, we find that either a conducting-wall-like mode or dissipative axial shear mode is the dominant drive even if turbulence drive from the sheath is small. The axial fluctuation between x-points are approximately flute. The relative fluctuating amplitudes are comparable to the case for uniform axial



equilibrium profiles. But diffusivities are much higher. The magnetic shear effect and x-point model has been developed.

#### Computational methodology and algorithms

We use an explicit predictor-corrector time advance. A fourth-order upwinding method has been used to solve the nonlinear convection term. A Fourier transform in poloidal-like direction  $y$  and finite difference in radial  $x$  and parallel  $z$  direction are employed. Since an implicit scheme is used for the parallel divergence term to stabilize collisional  $\omega_h$ -mode and since the radial ( $x$ ) and parallel ( $z$ ) coordinates are non-periodic, we resort to a dynamic alternating direction implicit (DADI) algorithm in  $x$  and  $z$  to invert the quasi-vorticity to give  $\phi(x,y,z)$ . The boundary conditions applied to  $T_e$ ,  $n_e$ ,  $V_i$ , vorticity and  $\phi$  in  $y$  direction are periodic. The boundary conditions in the  $x$  direction are  $T_e$ ,  $n_e$ ,  $V_i$ , vorticity and  $\phi = 0$  at  $x=0$  and  $x=L_x$ .

Careful tests have been completed over the entire range of parameters surveyed to verify that the results are not grid-sensitive. The simulation results show that the observed linear instability agrees well with theory. Furthermore, a doubly periodic ( $x$  and  $y$ ) version of the code has been benchmarked against a pseudospectral code[5], and was found to be in good agreement.

#### Development of the Fluedge

This code is an outgrowth of the 2D fluid turbulence code described in Ref. [1] with the improved numerics (fourth-order upwind differencing scheme described in Ref [6]). 3D extension includes field-line-following coordinates, magnetic shear/x-point geometry, two more time evolution equations for turbulence variables (density and parallel ion velocity), DADI for field solver, and an implicit scheme for the parallel divergence term to stabilize collisional  $\omega_h$ -mode. It took about 6 man months to develop the 3D extension (Fluedge) starting from a developed 2D code. It was first used in 1995.

It has been used by author and co-authors to investigate edge and SOL turbulence-related problems and L-H transition physics. The 2D code has been used as a turbulence module for Self-Consistent modeling of Turbulence and Transport under CORSICA project (a project funded by LLNL Director initiative) as a testbed for global turbulence problem. The 3D code will be used again to test multiple fields coupling and to self-consistently investigate L-H transition.

#### Description of current environment

- W1 originally developed on workstations platform
- Simple port to C90:  
(done-- but the code is slow, not being either vectorized or parallelized)
- Optimize for C90 (vectorize/parallelize; in process)

#### Code timing

- On SGI Power Challenge or Sun hyper Sparc: 3 CPU hours for grid resolution at  $32 \times 32 \times 32$ .

## REFERENCES:

- [1] Xu, X.Q., Phys. of Fluids B5 (1993)3641;  
Cohen, R.H., Xu, X.Q., Physics of Plasmas Vol. 2 (1995) 3374.
- [2] Xu,X.Q.,Cohen,R.H., Plasma Phys. Control. Fusion 35 (1993) 1071.
- [3] Tsidulko Yu,Berk H L,and Cohen R H,Phys. of Plasmas 1 (1994) 1199.
- [4] Berk H L, Cohen R H, Ryutov D D, Tsidulko Yu, and Xu X Q,  
Nuclear Fusion 33 (1993) 263--282
- [5] A. E. Koniges, J. A. Crotinger, and P. H. Diamond, Phys. of Fluids  
B4, 2785 (1992).
- [6] X.Q.Xu,R.H.Cohen, J.A.Crotinger, and A.I.Shestakov, Physics of Plasmas  
2 686-701(1995).

## HAWC and HWDIA

CODE NAME: HAWC and HWDIA  
HAWC AUTHOR: James A. Crotinger  
HWDIA AUTHORS James A. Crotinger, Alice Koniges,  
Paul Amala, Bill Dannevik  
SUPERVISOR: Alice Koniges  
CODE PURPOSE: Fluid turbulence simulation  
FIRST USED: 1989-1990

HAWC is a 2D fluid turbulence code that solves the 2-field Hasegawa-Wakatani (HW) equations. HWDIA solves the full DIA equations for the HW model. HAWC and HWDIA were developed mainly during FY89-FY91.

These codes are meant for doing basic research in the area of plasma turbulence, and don't interact with any other codes directly (aside from comparisons between the two). However HAWC was used to test turbulence-transport coupling algorithms in a prototype for the Corsica project.

These codes are still being used sporadically to study HW turbulence, and other related types of drift-wave turbulence, and will continue to be on the back burner for the rest of this fiscal year at least.

HAWC probably required 0.5 man-years to develop. HWDIA probably required 2 man-years. These were mostly LLNL postdoc /grad-student years, not full FTEs. The work was done entirely at LLNL.

## CODE MODEL, ALGORITHM, ETC.

### a. HAWC

HAWC solves the Hasegawa-Wakatani model equations for dissipative drift wave turbulence on a periodic 2D grid. The equations are solved using a pseudo-spectral technique in the spatial dimensions, and using an explicit Runge-Kutta solver to do the time advance. The code is written entirely in Fortran, and makes extensive use of the

Basis system. The latter provides an easy-to-use programmable interface, online help, portable binary save files, a flexible history package, and interactive graphics.

The code uses the "constant k-parallel" approximation for treating the third dimension in the HW equations, and it adds artificial dissipation at high  $k_{\text{perp}}$  to keep cascading energy from building up in a non-physical fashion. Good 3D simulations of the HW model have not been performed, so it is difficult to assess the impact of the "constant k-parallel" assumption. The results are found to be fairly insensitive to the high- $k_{\text{perp}}$  dissipation, so long as that dissipation is sufficient to keep energy from accumulating.

#### b. HWDIA

HWDIA solves the full multi-field, two-time, direct interaction approximation (DIA) equations that have been derived from the HW model equations as solved by HAWC. The DIA is a turbulence closure theory that consists of coupled, nonlinear, integro-differential (with time-history integrals) equations for the two-time correlation functions and the "infinitesimal response functions" for all the modes and for all field combinations. The HWDIA code uses a pseudospectral method very similar to that found in HAWC to solve the DIA equations. It uses a partially implicit second-order technique for the time advance. HWDIA is written entirely in Fortran, and also uses the Basis system extensively. (The overall operation is very similar to that of HAWC.)

While the periodic 2D grid is not the optimal grid for solving the DIA equations for this problem, it was chosen to keep to an absolute minimum the differences between HWDIA and HAWC. The goal was to be able to solve the primitive HW equations and the DIA equations on the identical grids with identical physics parameters, and compare the solutions. Because of the complexity of the equations and the presence of the time history integrals, this was found to be very difficult without further assumptions (indeed, only very low resolution grids could be solved on the NERSC CRAY C90). The solution was to impose a limited memory on the time-history integrals. Data older than this time cutoff was simply forgotten, the theory being that data older than an autocorrelation time should not significantly contribute to the integrals anyway. Studies were done that indicate that this can be a very good approximation.

#### CODE RUN-TIME, TESTING, ETC.

HAWC requires approximately 1.5 seconds per drift wave period to run with "normal" parameters at 128x128 resolution on the Cray C90 at NERSC. A typical run time at this resolution would be 15-60 minutes. However we also do many ensemble runs at 40x40 resolution for purposes of comparison to the DIA. These typically require hundreds of realizations to get good statistics, and typically take about 3 minutes per realization. Thus these ensemble runs can require 10's of hours to generate good statistics. Furthermore, recent interest has led us to push HAWC into the near-adiabatic regime. Here an 80x80 run can take 5 or more hours of Cray C90 CPU time to saturate because of the vast disparity between the very slow linear growth rate and the very fast relaxation of the density to its adiabatic value. (This problem could be fixed by switching to a semi-implicit solver, but I have not had the time for such an undertaking.)

The DIA code takes roughly 50-100 times longer than the simulation code, per timestep, and even with the time history cutoff is limited to 40x40 (perhaps 64x64) modes on the C90. While this is a big factor, it is not so bad when one considers how poor the statistics are from an ensemble of 100 runs.

Both codes have been tested by a variety of means, including checking linear growth rates, global conservation laws, and the ability to find dissipationless stationary states whose spectral properties can be predicted by equilibrium statistical mechanics.

Publications based on HAWC and HWDIA:

The only paper dealing with HWDIA is still in preparation:

J.A. Crotinger, "DIA closure studies of 2-D drift wave turbulence and the effect of coherent structures on transport," in preparation.

The following papers contain contributions from work done with HAWC:

D.E. Newman, P.W. Terry, P.H. Diamond, Y.-M. Liang, G.G. Craddock, A.E. Koniges, J.A. Crotinger, "The dynamics of long wavelength electrostatic turbulence in tokamaks," Phys. Plasmas, 1, 1592 (1994).

G.G. Craddock, A.E. Koniges, J.A. Crotinger, P.H. Diamond, D.E. Newman, P.W. Terry, "Effects of nonlinear electron dynamics in a fluid model of collisionless trapped electron mode turbulence," Phys. Plasmas, 1, 1877 (1994).

A.E. Koniges, J.A. Crotinger, and P.H. Diamond, "Structure Formation and Transport in Dissipative Drift Wave Turbulence," Phys. Fluids B, 4, 2785 (1992).

F.Y. Gang, P.H. Diamond, J.A. Crotinger, A.E. Koniges, "Statistical Dynamics of Dissipative Drift Wave Turbulence," Phys. Fluids B 3, 955 (1991).

A.E. Koniges, J.A. Crotinger, W.P. Dannevik, G.F. Carnevale, P.H. Diamond, and F.Y. Gang, "Equilibrium Spectra and Implications for a Two-Field Turbulence Model", Phys. Fluids B 3, 1297 (1991).

L.L. LoDestro, et al., "Comparison of Simulations and Theory of Low-frequency Plasma Turbulence," (Proc. of 13th Int. Conf. on Plasma Phys. and Cont. Fusion Res.) IAEA, Washington D.C., IAEA-CN-53/D-1-3, 1990.

P.H. Diamond, et al., "Developments in the Theory of Trapped Particle Pressure Gradient-Driven Turbulence in Tokamaks and Stellarators" (Proc. of 13th Int. Conf. on Plasma Phys. and Cont. Fusion Res.) IAEA, Washington D.C., IAEA-CN-53/D-1-2, 1990.

## PW1

Code name:	PW1
Author:	Xueqiao Xu and Jack Byers Lawrence Livermore National Laboratory
Code Description:	Parallelized 1D kinetic SOL PIC-MCC code
First Use:	1995 at LLNL
Language:	Fortran + PVM

## Summary of Code

This code is a highly parallelized implementation of the 1D2V collisional PIC-MCC code (W1 by Batishchev[1]) for Scrape-Off-Layer (SOL) plasmas on the Cray T3D using the PVM message-passing library. The particles are divided equally among the processors, and grid quantities (temperatures, etc) are averaged from all processors. The electric field is solved using the electron momentum equation. The logical sheath potential is calculated in a serial fashion. For a fixed number of particles (1 million) in the simulations, we have obtained a factor of 31 speed-up by using 32 processors compared to 1, and a factor of 90 speed-up by using 128 processors. Through our efforts on Cray T3D with 256 processors we have learned that massively parallel MIMD systems offer a new supercomputing capability for kinetic SOL physics modelling.

PW1 is used to facilitate our ongoing investigation of the effects of neutrals on divertor plasma detachment phenomena and parallel heat and particle fluxes in the presence of strong gradients where fluid descriptions break down. We have conducted a serial simulation runs for detached and attached plasma flow simulations and compared them with solution from a one-dimensional fluid model and with experimental detachment data.[3]

Our research for parallel computing has two components: (1) Port and parallelize Batishchev's 1D kinetic collisional particle-in-cell Monte-Carlo code (PIC-MCC)[1], W1. Porting and initial parallelization of W1 has been completed. (2) Develop a 2D version of W1. This is much larger undertaking, requiring both algorithm development and much longer running times.

### Computational methodology and algorithms

We have parallelized the 1D2V collisional PIC-MCC code for SOL plasmas on the Cray T3D using the PVM message-passing library. This style of parallel programming communicates shared data via explicit requests to send data (messages) from one process to another, or to a group of processors. We describe below the efforts which were undertaken to efficiently parallelize the code on the Cray-T3D.

#### 1) Division of Tasks

The particles are divided equally among the processors and the grid quantities (temperatures, etc.) are then averaged from all processors. The electric field is solved using the electron momentum equation. The logical sheath potential is calculated in a serial fashion.

#### 2) Global Average

In message-passing parallel programming models, all variables and allocated memory are located in the private processors. Each processor runs its own copy of the code with number of particles  $N/NPE$ , where  $N$  is total number of particles and  $NPE$  is number of processor elements. Each processor accumulates its share of array of sums into a private subtotal grid array. Then it sends its grid array to other processors and receives other processor's arrays and averages each element of the shared total array in turn.

Every time step we need global average of following 15 quantities: density, perpendicular and parallel temperature, parallel pressure, parallel velocity (mean), friction force, parallel heat flux for each species, and ionization frequency over the all processors. In addition, in order to conserve number of particles, momentum and energy in each grid

point in Coulomb collisions, we also average number of particles, momentum and energy among the processors. These summations are expensive and need to be minimized.

### 3) Logical Sheath Boundary Conditions (in serial)

The algorithm for the logical sheath boundary conditions is as follows:

I. Advance the trajectories of all particles in parallel.

II. Each processor accumulates its share of array of electrons and ions, which cross the plate boundary location each time step into a private temporary array; then it sends its temporary array to a designated processor. This designated processor receives all other processor's temporary particle array and counts the total number of ions  $N_i$  and the total number of electrons  $N_e$ .

III. Compare  $N_e$  to  $N_i$ :

Case A. If  $N_e \geq N_i$  then (most probable condition)

i. Order the electrons by parallel velocity from fastest to slowest,

ii. Absorb all  $N_i$  ions and the fastest  $N_i$  electrons.

iii. Reflect the slowest  $N_e - N_i$  electrons.

iv. Distribute the remaining electrons to all processors by the number of ions lost in each processor to keep equal number of electrons and ions.

Case B. If  $N_e < N_i$  then reflect all electrons in each processor.

### Development of PW1

W1, written by Batishchev, is a non-stationary 1D2V pure kinetic model of high recycling SOL plasmas. Through LLNL-MIT collaboration, W1 has been used to demonstrate the transition between attachment and detachment of divertor plasmas from high power density to low power density; the transition occurs around  $1000 \text{ MW/M}^2$  for C-MOD. We have also made comparisons between W1 and 1-D version of the UEDGE fluid code with and without flux-limited thermal conduction. However, W1 is time-consuming. For a typical production run (100 microsecond, or 138,900 time steps and 20,000 particles), it took 7-14 days on single processor SGI Power Challenge, depending on input power density. SGI Power Challenge has a similar speed as HP9000/750, and Cray C90 for non-vectorized code.

We find that 90% of the run time is spent in one do-loop of COUCOT, calculating Coulomb collisions of ions and electrons. In the Monte Carlo collision algorithm, particles are tracked through a background plasma and their interactions with background plasma are determined by the combination of the background plasma properties and a stochastic process. Due to the stochastic process and complicated coefficients in Rosenbluth potentials, this do-loop involves three levels of nested subroutine-function calls and a deeply nested set of if-statements. This complication makes the code difficult to vectorize or parallelize via compiler directives. From 1D to 2D, we may need to add at least 10 times more particles and the code will run at least 10 times longer. Clearly, there is a need to speed up the kinetic SOL code

With a substantial effort (6 man months), we have created a parallelized version of our PIC-MCC code which runs at a parallel efficiency of 96% on Cray-T3D using 32 processors, and we have obtained an absolute speed of 14.38 times the performance of the scalar version of the code running on a single CPU of Cray-YMP C90 supercomputer. The same code can run at a parallel efficiency of 70% on Cray-T3D using 128 processors and 1 million particles. In this case, we have obtained an absolute speed of 41.92 times the performance of the scalar version of the code running on a single CPU of Cray-YMP C90 supercomputer.

It has been used by author and co-authors to investigate edge and SOL kinetic effect and divertor detachment physics. It is a multi-institutional project: LLNL, MIT and LODESTAR. Workers at these institutions and perhaps other DOE contractors will use these codes. Description of current environment. -- W1 originally developed on PC/486 platform -- Port to workstations -- port to C90: -- parallize on Cray T3D

Present and Future Work.

Extension to Two Dimensions.

Extension of PW1 to two dimensions is starting now (Mar 96). The first 2D model will consist of a simple coupled set of 1D problems, wherein the particles will be enabled to diffuse radially from one field line to another. The first attempt will be to get this going on a uniprocessor version and once this is working, to then implement the PVM structure in PW1 to parallize it. The 2D code will get intensive development in FY97.

References

- [1] Batishchev, O.V., Krasheninnikov, S.I., Sigmar, D.J., Sigov, Yu.S., and Soboleva, T.K., Contrib. to Plasma Physics, 34 (1994), 436.
- [2] X.Q.Xu, O.V.Batishchev, J.A.Byers, R.H.Cohen S.I.Krasheninnikov, T.D.Rognlien and D.J.Sigmar, Parallelization of and Results from Kinetic Edge Plasma Code W1, accepted by Contributions to Plasma Physics, June (1996).
- [3] O.V.Batishchev, X.Q.Xu, J.A.Byers, R.H.Cohen S.I.Krasheninnikov, T.D.Rognlien and D.J.Sigmar, Kinetic Effects on Particle and Heat fluxes in Detached Plasmas Submitted to Physics of Plasmas.

### TEQ

NAME: TEQ  
PURPOSE: TEQ addresses the area of toroidal axisymmetric MHD: plasma equilibrium and poloidal-field-coil design, and vertical stability analysis and control.  
FIRST USED: 1987  
AUTHORS: Don Pearlstein, Scott Haney, Lynda LoDestro, Dick Bulmer  
FREQUENCY OF USE: Many short runs daily by several users.  
IMPORTANCE: TEQ is a very important axisymmetric MHD equilibrium and stability code. It is one of the primary modules in the CORSICA comprehensive tokamak modeling code.  
LANGUAGES: Fortran and C++

TEQ came into being to fill a need for an in-house tokamak design tool as LLNL's MFE program made the switch from mirrors to tokamaks. The engineering & magnet-



design effort here began with a collection of closely related codes from ORNL (HEQ, VEQ, and NEQ, developed by Dennis Strickler). When difficulties arose and more capabilities were required, they came to the theory group; there was a short period evaluating other equilibrium codes, and a decision to work with and develop the ORNL codes.

Within TEQ itself, development directions have been in response to (1) needs, both for new calculations and to expedite others, of the design or modelling efforts; (2) an overall vision heading toward a Grad-Hogan transport code (e.g., this required solving the community-wide long-standing problem over using the safety factor as input to the Grad-Shafranov solver); and (3) changes in CORSICA itself.

TEQ was started in the fall of 1987 by combining the ORNL codes HEQ, VEQ, and NEQ (Dennis Strickler) under BASIS and then making substantial modifications (the present code bears little resemblance to the original ORNL codes). The principal authors, in chronological order, have been L. L. LoDestro, L. D. Pearlstein, R. H. Bulmer, and S. W. Haney. There are approximately 8 man-years of effort in developing TEQ, and development continues. In 1993, when the core-transport package was added, the overall code was named CORSICA, which was and remains directed by J. A. Crotinger.

TEQ is both an application tool to do analysis and prediction of experiments, a physics module to be used in integrated modelling, and a code for machine design. TEQ creates and reads EQDSK files, in order to communicate with a variety of codes at other institutions which use that format, for example, the General Atomics equilibrium code EFIT. TEQ is the workhorse equilibrium design tool (an important part of LLNL's OFE-funded effort); and it is the core of CORSICA (increasingly used for simulating ITER as well as modelling TFTR and DIII). There have been two significant contributions to TEQ from outside LLNL. An inverse-equilibrium solver developed by Drozdov in Russia, has been added as an option. The ballooning-stability analysis was developed in close collaboration with Glasser at LANL. His version (different algebra) is in DCON.

The TEQ code is a free-boundary axisymmetric equilibrium code. By "free boundary" is meant that it calculates the required coil currents to obtain a specific equilibrium; the magnetic field external to the plasma is thus also available. It is a highly flexible and fully interactive code operating within the LLNL "BASIS" shell. The code typically runs with a "direct" (i.e., it solves for the flux as a function of spatial variables) Grad-Shafranov solver in R-Z coordinates. It also has an "inverse" solver, which can be used instead; here the spatial coordinates are obtained as a function of flux and angle; this is the POLAR1 code mentioned in the preceding. Typically the latter solver is used for more physics-oriented problems. The code needs two arbitrary functions to provide the source; there is no rotation in the code at present. These profiles can be: pressure and R Btor as a function of poloidal flux; q (safety-factor) profile and entropy-density profile; or the flux-averaged toroidal current  $\langle J.B \rangle / \langle B.\text{grad } \phi \rangle$  and pressure. All profiles are either analytic with many profile forms or provided as data on a poloidal or toroidal flux grid. In addition to the equilibrium package the code has a complete linearized vertical stability and feedback package. This enables us to ascertain the growth rates and feedback requirements for real tokamak designs and experiments. In addition the code has a ballooning, Mercier and resistive-interchange package. These packages are typically run for each equilibrium. The ballooning calculation makes full use of the asymptotic analysis. We extract the coefficient of the large asymptotic solution, whose sign determines stability. This avoids errors associated with not integrating far enough.

In addition to the above-mentioned physics packages, the code provides extensive poloidal-field-coil analysis---accurately computing the field and forces within the coils.



This capability allows for the evaluation of additional diagnostics for superconductors, which, in turn, can be used as constraints on the equilibrium shape or flux linkage. A recent addition enables the code to handle ferro-magnetic material in the PF system. This overall capability is a critical part of the design work provided by LLNL.

TEQ serves as the MHD engine for the resistive-MHD part of the CORSICA code. Specifically, we use the TEQ code in free- or fixed-boundary mode. Geometry information and the plasma current are then fed to the Ohm's-law equation. The Ohm's-law equation then provides the  $q$  and entropy-density profiles, and the poloidal flux across the plasma, to the equilibrium. These two equations, along with any subsidiary transport equations are iterated together to convergence.

TEQ is written in Fortran, apart from the vertical stability package, which is written in C++. Special features of the code:

- \*Uses a Buneman solver for the elliptic Grad-Shafranov equation (fast Fourier).
- \*The Drozdov inverse equilibrium uses up-winding in the magnetic partial- differential equation.
- \*The code runs on a multitude of platforms: C90, SUN, HP, IBM and soon on the SGI.
- \*We use a portable data-base file-format so binary files from oneplatform are readable on all platforms.
- \*The code will find equilibria subject to virtually any set of constraints the user can dream up without any code modification.
- \*It is highly vectorized and is quite fast and robust.

A typical equilibrium on a 33 x 65 mesh (up-down asymmetric) with residual errors of  $1.e-8$  requires about 30 iterations or 1-4 seconds (depending on the profile option) on the C90. The vertical-stability calculation has minimal impact on the time of a typical run; the MHD analysis is basically free.

TEQ has been used extensively for tokamak designs: TPX, ITER, and the PCAST copper machine. The various pieces have been extensively benchmarked, the equilibrium with a vast array of codes both domestic and international. The MHD stability packages have been compared with: Princeton codes in the TPX design and GA codes for DIII-D, ITER, and more recently with their tight-aspect-ratio designs. There have been many comparisons on the ITER designs, national and international. Benchmarking with the vertical stability has been less extensive. The primary reason for this is that the majority of the codes in use make the simplifying assumption of the rigid-filament model. The only benchmarking done has been with the NOVA-W code, which also uses ideal MHD. These two codes agreed quite well.

Publication:

L.L. LoDestro and L.D. Pearlstein, "On the Grad-Shafranov equation as an eigenvalue problem with implications for  $q$ -solvers," Phys. Plasmas 1, 90 (1994).

## FOREV

Code name: FOREV  
Author: Karlsruhe group, some modifications and runs being made at LLNL (Craddock, Koniges, et al.)  
Code Description: 1D radiation hydro / ablation  
Language: Fortran and Turbo Pascal  
Summary of Code

This code solves one dimensional hydro with ablation, B field diffusion, and radiation. It models the divertor target erosion due to intense heat caused by singular MHD events, such as disruptions or Edge Localized Modes. Physically, divertor material is evaporated into a thin (.2-2 micrometers) vapor layer. The main energy transfer mechanism of the deposited MHD beam energy in the layer to the target is radiative. FOREV models this radiative hydro problem using the forward reverse method for radiative transport.

### Computational Methodology and Algorithms

The processes modelled include:

- 1) Lagrangian motion
- 2) magnetic field diffusion
- 3) Heating and compression of vapor by external beam
- 4) radiation transport
- 5) radiative heat conduction
- 6) energy exchange between free electrons and atoms
- 7) energy exchange between free and bound electrons
- 8) vapor production

All these processes are modelled using splitting methods and each time step uses the results of the previous time step as an initial condition. The Lagrangian mesh uses equal masses for the cells. The method used is that of large particles [A. Belcherkovsky and V. D. Mavidov, "Methods of Large Particles in Gas Dynamics," Moscow, (1985)]. Magnetic diffusion, electron energy diffusion, and radiative energy diffusion are solved implicitly. The vapor evolution is calculated by introducing a temporary vapor cell and by rebuilding the mesh with a change in mass (the vaporized material). Radiative transport is multigroup using the forward reverse method.

### Development of FOREV

FOREV has its origins in Troitsk, Russia. I. Landman brought the code to Germany. The code has been worked continuously since 1990.

### Description of Current Environment

FOREV runs on HP, SUN, and IBM workstations. FOREV also runs on CRAY PVP machines (C90, YMP, etc). FOREV can run on the T3D front end or any single node. The code is not parallelized. PVM hooks are in progress to the parallel radiative transport code CRETIN.

### Code timings

On the workstations, for 32 zones, a typical run takes 6 hours at single precision. On the C90, the run takes an hour clock time with no hand optimization.

## CRETIN

Authors: H. Scott, H. Dalhed (T3D port in progress by Moon, Scott, and Koniges)

Languages: FORTRAN, C, MPI

Summary:

CRETIN is a one- and two-dimensional non-LTE atomic kinetics / radiation transfer code. It has been used as a postprocessor to the code UEDGE to investigate the structure and efficiency of radiative divertors for ITER (with A. Wan), and to try to understand radiative diagnostics from DIII-D. It is also being used to study the effects of vapor shielding due to divertor plate ablation during disruptions (with D. Eder and A. Wan).

Computation Methodology and Algorithms:

CRETIN self-consistently follows the time evolution of atomic populations and photon distributions as radiation interacts with a low-density plasma. Each element in the plasma is modelled with numerous atomic states, corresponding to the distribution of electrons in various atomic levels. Transitions between different atomic and free-electron states are caused by interactions between electrons, ions, atoms and photons. Radiation transfer is calculated with deterministic algorithms in planar, cylindrical or spherical geometries in 1-D, or in xy or rz geometries in 2-D. Continuum radiation is treated separately from line radiation, with a linearization procedure used to achieve consistency between the line radiation and atomic populations. The physical models and algorithms used for the kinetics and radiation transfer are identical to those in the code GLF, which is described in H.A. Scott, R.W. Mayle, Applied Physics B, Vol. 58, pp. 35-43 (1994). Both the atomic kinetics and radiation transfer calculations can be extremely time-intensive. However, they are also highly parallel. For many problems, the atomic kinetics calculations are completely local and can be parallelized across spatial zones. The zones are couple through the radiation transfer calculations, which can be parallelized across energies (for continuum radiation) or across lines (for line radiation). CRETIN is currently parallelized in this manner. One additional physical process important to divertor plasmas which has been added to CRETIN is diffusive transport of neutral atoms, which is used to simulate neutral recycling or gas puffing. The transport is closely coupled to the atomic kinetics and spatially couples all the zones. The current solution technique is to simultaneously solve the coupled kinetics and transport equations, directly in 1-D and iteratively in 2-D. Operator splitting approaches have so far been unsuccessful.

Current Environment:

CRETIN currently runs on PC's, Unix workstations (HP, IBM, DEC, Sun, SGI), and CRAYs. Parallel versions run on the BBN T2000, Meiko CS2 and multi-processor SGI workstations. We are currently porting the MPI version of CRETIN to the T3D. CRETIN is written mostly in Fortran 77. On some systems, a few C routines are used to provide capabilities not available in standard Fortran 77. Compilation options also include using CFT-style pointers or Fortran 90 allocatable arrays for memory management.

Code Timings:

The size of a small test case depends upon the physics being addressed. A divertor energy balance test can be done using only a small hydrogen model with several levels, a

couple strong lines and a few dozen zones in 1-D or several hundred zones in 2-D. This test would only require a few 10's of MBytes memory and a couple minutes of time on a C90 (for the 2-D test). Including a small impurity model would increase the number of levels by an order of magnitude, with roughly similar increases in the memory and run times.

Resources required for production:

Scaling the resource requirements from the test cases to full production problems is not straightforward, as the algorithms for the different physical processes all have different scalings. The dominant scalings are likely to be roughly linear in the number of zones, energies, and strong lines (for both memory and time) and roughly quadratic/linear (for memory/time) in the number of levels for the regime we expect to operate in. These scalings result in memory requirements up to the 10 GByte range and run times of several hours for full 2-D simulations. With the exception of the diffusive transport, these simulations should be highly parallel, allowing us to spread the memory and time requirements over a large number of processors. Possibilities for parallelizing the diffusive transport remains to be determined.

Other Resources:

CRETIN can use the PACT libraries and utilities for much of its output, postprocessing and graphics, and can use XGRAFIX for interactive graphics, if these are available.

Development:

Cretin is an unclassified code first developed to study accretion disks in stellar atmospheres. (Thus the name cretin). It was developed in the 90's.

### Gyrokinetics

CODE NAME: Gyrokinetics and related particle codes  
CODE AUTHORS: A.M. Dimits, B.I. Cohen, J.A. Byers, T.J. Williams  
PURPOSE: Core turbulent transport simulation, Numerical Tokamak Project  
LANGUAGES: Fortran + Message Passing, C  
FIRST USE: 1990 at LLNL

Gyrokinetic simulation of turbulence and transport in tokamaks; Advanced algorithm development applicable to both kinetic and fluid simulations of tokamak turbulence, transport, and other phenomena (e.g., quasiballooning and related spatial representations, quiet-delta-f PIC+Monte-Carlo Codes, Implicit-PIC codes)

### 3D GYROKINETIC CODES

None of the gyrokinetic codes at LLNL have been developed from scratch (1990-1996). The gyrokinetic code development started with a 3D slab gyrokinetic code provided by R.D. Sydora in 1990, and has proceeded along the following path (1) Implementation of flux-tube geometry and quasiballooning representation. These developments enabled us to do the first gyrokinetic simulations of large tokamaks (TFTR) (reported at IAEA 1994 and U.S. Conferences since). (2) Implementation of toroidal-ion gyrokinetic physics. This is the minimum physics of any relevance to tokamak core transport (IAEA '94). These developments have been completed and reported in several conferences and published papers.

The following are in various stages of progress: (3) Addition of noncircular geometry. This is essential for relevance to most present-day and proposed tokamaks. (4) Addition of a bounce-averaged electron model. This is strongly motivated by the fact that gyrokinetic simulations of toroidal ITG turbulence with adiabatic electrons fail to agree with some experiments. (5) Development of a massively parallel gyrokinetic code. This is dictated mainly by the need for the memory offered by massively parallel computers to do demonstrably converged kinetic simulations, by the performance offered by such computers, and by the fact that much computer time is available on these machines.

#### OTHER CODES

Used mainly for algorithm development, testing and demonstration, and are supported as needed during the algorithm development and testing.

(A) 2D gyrokinetic code with slab and local toroidal physics. This code has been used as a platform to develop an ion-ion collision model and implicit orbit-averaged schemes for electron dynamics (both electromagnetic and electrostatic).

(B) 3D Finite-Difference fluid code. This code originated at the University of Maryland, and was used for the first implementation and demonstration of the quasiballooning representation in 1990-1993.

#### 3D GYROKINETIC CODES

##### G3ES

Slab 3D electrostatic gyrokinetic code. First used in 1990. This code evolved from one provided by R.D. Sydora of UCLA in 1990, and has various physics model improvements based on current understanding, as well as numerical algorithm improvements such as vectorization of key portions of the code and multitasking for the C90. Was used to study slab ITG turbulence. Authors: T.J. Williams, J.A. Byers, A.M. Dimits. This code was used extensively by us up to 1993 to study slab ITG turbulence, and was our main gyrokinetic code prior to the development of toroidal versions of the code.

##### PG3ES

Massively parallel version of G3ES. Parallelization carried out by T.J. Williams. Used to develop and demonstrate parallelization methods on a wide range of massively parallel computers. Developed and used since 1993. Still used to develop optimal parallelization methods and performance benchmarks on new parallel computers as they become available.

##### G3EQ

Toroidal quasiballooning-representation 3D gyrokinetic code. This has been our main gyrokinetic physics/production code and has produced a definitive study of ITG turbulence in TFTR. The code was first used for physics studies in late 1993. Developed by A.M. Dimits, T.J. Williams, and J.A. Byers. This Code was used extensively in 1994-1996, and is still in use for studying the effects of negative magnetic shear and toroidal velocity shear, and is evolving into the following versions with the inclusion of

more physics and implementation on massively parallel machines. These versions will be merged once each of the new developments has been thoroughly tested.

**G3EQG** General noncircular-cross-section arbitrary-aspect-ratio version of G3E. Developed by A.M. Dimits. Presently undergoing testing and benchmarking against linear calculations carried out by G. Rewoldt for DIII-D.

**G3EQE** Version of G3E that includes bounce-averaged electrons. Presently in development by A.M. Dimits.

**PG3EQ** Massively parallel version of G3E. Developed by T.J. Williams and A.M. Dimits. Some production runs have been carried out on T3D. This code will become our main physics/production code once more efficient I/O is implemented, and some key diagnostics have been ported. The noncircular-cross-section and bounce-averaged electron capabilities will be ported into this code.

### G2E

2D gyrokinetic code with slab and local toroidal physics. This code has been used as a platform to develop an ion-ion collision model and implicit orbit-averaged schemes for electron dynamics (both electromagnetic and electrostatic). (see Brackbill and Cohen, Multiple Time Scales, Computational Techniques, Academic Press Inc., 1995).

### 3DFL

Finite-difference fluid code. This code originated at the University of Maryland, and was used for the first implementation and demonstration of the quasiballooning representation in 1990-1993.

### ES1DF

This is a 1D electrostatic quiet-delta-f general hybrid code. This code was developed to provide a testbed for the investigation, development, testing and demonstration of powerful quiet delta-f methods in a wide range of applications. This code was developed in 1995, and is in use for development of various multiple-timescale algorithms for treating electron dynamics.

### Code Development:

The G3E and G2E codes evolved from a code provided by R.D. Sydora of UCLA in 1990, which in turn was developed in collaboration with W.W. Lee at PPPL. Development of above versions was carried out at LLNL. Very approximately, there have been ~2 man-years of code development in G2E, ~2 man-years of code development in the slab G3ES, ~2 man-years in PG3ES, and ~3 man-years in the G3EQ family. 3DFL originated at the University of Maryland. The quasiballooning- representation development and implementation was done at LLNL. The ES1DF code is based on the ES1 code developed by A.B. Langdon and coworkers, and on a 1D implicit fluid code developed by J.F. Drake and P.N. Guzdar at the University of Maryland.



## Code descriptions:

### G3EQ, G3EQE, G3EQG

The physical model used is electrostatic, with a single fully toroidal nonlinear gyrokinetic ion species that has equilibrium temperature, density, and velocity gradients. The electrons in G3EQ and G3EQG are adiabatic. This is a key simplifying assumption that typically lowers the simulated transport rates and makes the code unable to address particle and electron thermal transport. This assumption is being fixed by the development of the new bounce-averaged delta-f electron model in G3EQE.

The gyrokinetic Vlasov equation is solved using the partially linearized delta-f particle method (Dimitis and Lee, *J. Comp. Phys.* 107, 309 (1993)), with four-point gyroaveraging (Lee, *J. Comp. Phys.* 72, 243 (1987)). The electrostatic potential is obtained from the gyrokinetic Poisson equation. The electron response is taken to be adiabatic, with a zero response to the flux-surface-averaged potential. Most of the particle simulation methods used in our codes are grounded in standard techniques that are documented in numerous publications, e.g., C.K. Birdsall and A.B. Langdon, *Plasma Physics via Computer Simulation*, McGraw-Hill, 1985.

The main free energy source for instability is the ion temperature gradient. Self-generated turbulent-Reynolds'-stress-driven flows, along with their dominant collisionless damping are included fully and self-consistently. External sheared parallel, ExB, or toroidal flows are also included, if needed.

The simulation domain used is a flux tube of small perpendicular extent, but which spans one or more poloidal circuits in the parallel direction. The field quantities in the code are defined on a quasiballooning-coordinate grid. The radial differences, interpolation, deposition, and smoothing are formed using shapes in configuration space (not ballooning-coordinate space) that are independent of poloidal location. This choice prevents grid collapse and resolution loss in the presence of magnetic and velocity shear, and is necessary to allow a smooth implementation of the toroidal periodicity condition across the parallel boundary. Because of the above two algorithm advances, we have been able to undertake the only 3D nonlinear kinetic simulation studies so far for large tokamaks of fusion interest (e.g. TFTR).

Profile relaxation is prevented by making the simulation volume periodic in minor radius to in such a way as to give a seamless radial connection even in the presence of external velocity shear.

PG3EQ: This code has the same physics as G3EQ, but is massively parallel.

### Earlier-Version Codes:

G3ES, PG3ES: 3D Slab gyrokinetic codes. The neglect of toroidal effects results in significant reductions in transport rates below predictions of toroidal codes and experiments. PG3ES is still used for algorithm performance testing since its structure is more typical of PIC codes than that of PG3EQ.

G2E: 2D Slab gyrokinetic code. 2D assumption strongly affects results. Use of code is mainly for algorithm development. This is a good platform since the 2D assumption makes the k space more controllable than in a 3D code.

3DFL: This is an explicit 3D finite-difference fluid code. The physical model consists of long-wavelength fluid equations which are physically reasonable, but are not expected to give quantitatively accurate results. This was used mainly as a simple platform for development of the quasiballooning representation. The quasiballooning version was shown to agree in detail with a standard-coordinate version of this code wherever the latter was not too expensive to run.

ES1DF: This is a 1D electrostatic general hybrid quiet-delta-f code. The distribution function is split into a local fluid response and a kinetic remainder. The fluid portion evolves via fluid equations which are closed by higher moments if the kinetic remainder. The kinetic remainder is solved for using direct delta-f particle methods. This code was developed mainly as a testbed for multiple-timescale algorithms for a wide range of applications. The validity of the 1D and electrostatic assumptions is dependent on the application, but is expected to be quite good for the tokamak scrapeoff layer, for example.

Code results:

G3EQ, G3EQE, G3EQG

We have used the code to do the first systematic gyrokinetic simulation study ITG turbulence with global parameters in the regime of fusion-relevant tokamaks. GyroBohm scaling is observed and the transport rates are too low to account for some TFTR L-mode discharges. This code has undergone extensive comparison with other gyrokinetic codes (mainly that of S. Parker at PPPL) and with gyrofluid codes (of R. Waltz at GA, and G. Hammett and Coworkers at PPPL) as part of the NTP code-comparison exercise. Linear growth rates agree with the other codes. Reasonable agreement was found in the nonlinear phase with all of these codes except the PPPL gyrofluid code which gives generally larger transport rates, typically by a factor of 2 to 4.

Code timings vary widely depending on the size of the problem and the computing platform. The two-dimensional testbed code G2E and the one-dimensional code ES1DF can consume a few minutes to 0.5 hour typically on the C90. They are both carefully vectorized. The three-dimensional codes like G3EQ typically consume multiple hours of cpu time on the C90 or massively parallel computer. Williams has put a lot of effort into making the code efficient as a parallel code on various platforms and has demonstrated scaling with processor number.

PG3EQ

This code has been tested for detailed numerical agreement to machine roundoff against G3EQ.

ES1DF

Detailed agreement has been demonstrated for the frequency and Landau damping rate for plasma oscillations and for the growth rates for beam-plasma instabilities with ES1, although ES1DF is much quieter for a given number of particles.

The following publications describe results from the research using these codes contributing to the understanding of core transport in near-fusion tokamaks and to the development of improved algorithms for performing these simulations.

B.I. Cohen and T.J. Williams, "Semi-Implicit Particle Simulation of Kinetic Plasma Phenomena, J. Comp. Phys. 97, 224 (1991).



B.I. Cohen and T.J. Williams, "Implementation of a semi-implicit orbit-averaged gyrokinetic particle code," J. Comp. Phys. 107, 282 (1992).

A.M. Dimits and B.I. Cohen, "Simulation models for tokamak plasmas," in Proceedings of the IAEA Technical Committee Meeting on Advances in Simulation and Modeling of Thermonuclear Plasmas, Montreal, Canada, (June 1992).

B.I. Cohen, T.J. Williams, A.M. Dimits, and J.A. Byers, "Gyrokinetic simulations of ExB velocity-shear effects on ion-temperature-gradient modes," Phys. Fluids B 5, 2967 (1993).

A.M. Dimits "Fluid Simulations of Tokamak Turbulence using Quasiballooning Coordinates," Phys. Rev. E 48, 4070 (1993).

Timothy J. Williams, "3D Gyrokinetic Particle-In-Cell Simulation of Fusion Plasma Microturbulence on Parallel Computers," Proc. 1993 SCS Simulation Multiconference High Performance Computing Symposium, Arlington, VA, March 29-April 1, 114-119.

M. Kotschenreuther, H.L. Berk, ... B.Cohen, "Simulations for confinement in near-fusion experiments, in Proceedings of the Thirteenth International Conference on Plasma Physics and Controlled Nuclear Fusion Research, Wurzburg, Germany, Sept. 30-Oct. 7, 1992, IAEA-CN-56/D-1-2 (IAEA, Vienna, 1993).

A.M. Dimits and B.I. Cohen, "Collision operators for partially linearized particle simulation codes," Phys. Review E 49, 709 (1994).

B.I. Cohen, D.C. Barnes, et al., "The Numerical Tokamak Project: simulation of turbulent transport," Comp. Phys. Commun. 87, 1 (1995).

B.I. Cohen, A.M. Dimits, J.J. Stimson, and D.C. Barnes, "Implicit-moment partially linearized particle simulation of kinetic plasma phenomena," (LLNL Report UCRL-JC-121734, September 1995), accepted for publication in Phys. Rev. E.

A.M. Dimits, T.J. Williams, J.A. Byers, and B.I. Cohen, "Scalings of Ion-Temperature-Gradient-Driven Anomalous Transport in Tokamaks," (LLNL Report UCRL-JC-122244, Submitted to Physical Review Letters, Oct. 1995).

A.M. Dimits, J.A. Byers, T.J. Williams, B.I. Cohen, et al., "Gyrokinetic and global fluid simulations of tokamak microturbulence and transport," Fifteenth International Conference on Plasma Physics and Controlled Nuclear Fusion Research, Sept. 26, 1994, Seville, Spain, IAEA-CN-60/D-P-I-5 (IAEA, Vienna, 1995).

## TORCH

Name:	TORCH (TOroidal Ray tracing, Current drive, and Heating)
Purpose:	Calculates ray trajectories of high-frequency waves in a tokamak plasma and computes driven current and power deposition on each flux surface and the spatial distribution of transmitted power
Year first used:	1986 (ancestors RAYS and TORAY were used in 70's and early 80's)
Authors:	Gary R. Smith and Arnold H. Kritz
Frequency of use:	Hundreds of (short) runs each year

**Importance:** The only ECH/ECCD code presently maintained in the U.S. The only code worldwide that implements the weakly relativistic ray-tracing approximation.

TORCH was developed with A.H. Kritz (Lehigh University). Total investment in code development (including ancestors): 20 man-years

Ray trajectories are computed within the geometric-optics model, which includes refraction effects by neglects diffraction. The tokamak magnetic field and radial plasma profiles are taken from analytic models or from the output of equilibrium codes. The ray equations contain derivatives of the plasma dielectric-tensor elements, which are computed in one of three limits: cold-plasma, weakly relativistic approximation, or fully relativistic. Solutions of the ray equations are obtained by using solvers for sets of coupled ordinary differential equations. Absorption can be computed either weakly or fully relativistically. Dielectric- tensor elements and their derivatives require careful computation for the code to be accurate and robust. Current-drive efficiency is calculated fully relativistically within the adjoint approximation. The TORCH code is well-suited for use within a predictive transport code or, if used iteratively, within an interpretive code. Appropriate ray-tracing and absorption models are available for all parameter regimes encountered in fusion devices. The current-drive calculation does not include the momentum-conservation effect, which causes modest underestimation of driven current.

#### Code comparisons and validation:

(a) The various ray-tracing and absorption models within TORCH have been intercompared to assess their limits of validity.

(b) Further comparisons of absorption models have been done between TORCH and the GA version of TORAY and with Fokker Planck codes written at LLNL and GA and in France, England, and the Netherlands.

(c) Diagnostics have rarely been available to determine absorption fractions and current-drive efficiencies or profiles of absorption and current drive in experiments. Absorption fractions measured with a calorimeter in MTX at LLNL agreed with TORCH, even into the nonlinear heating regime. Profiles of transmitted power in Tore Supra near cyclotron resonances show the qualitative features predicted by the weakly relativistic ray-tracing approximation within TORCH.

#### Publications based on TORCH:

A. H. Kritz, G. R. Smith, W. M. Nevins, and R. H. Cohen, "Power Deposition and Current Drive by Intense Microwave Beams in Tokamaks," *Phys. Fluids B* 1, 142 (1989).

R. C. Myer, M. Porkolab, G. R. Smith, and A. H. Kritz, "Electron Cyclotron Wave Propagation and Absorption in the Compact Ignition Tokamak," *Nucl. Fusion* 29, 2155 (1989).

G. R. Smith, M. E. Fenstermacher, and E. B. Hooper, Jr., "Modelling of Electron-Cyclotron Power Profiles on a Calorimeter in the Microwave Tokamak Experiment," *Nucl. Fusion* 30, 2505 (1990).

Gary R. Smith, Daniel R. Cook, Allan N. Kaufman, Arnold H. Kritz, and Steven D. McDonald, "Scattering of Ordinary-Mode Electron-Cyclotron Heating Waves by Density Fluctuations in Tokamaks," *Phys. Fluids B* 5, 4299 (1993).

## Reflectometry Codes

**Name:** O1D, X1D, O2D, X2D, HELM1D, SOFTSTEP  
**Purpose:** Suite of codes to calculate the propagation and reflection in plasma of microwaves to model plasma diagnostic reflectometry.  
**Year first used:** 1994  
**Authors:** Bruce I. Cohen, Thomas B. Kaiser, Bedros B. Afeyan, Albert E. Chou (UCLA/UC Davis), and John C. Garrison  
**Frequency of use:** Hundreds of short and moderate length runs per year.  
**Importance:** This is the most comprehensive suite of codes supporting conventional and ultra-short-pulse reflectometry, theory and experiment. These are the only codes addressing ultra-short-pulse reflectometry modeling.

The suite of reflectometry codes was largely developed under LLNL LDRD funding as a collaborative project between LLNL MFE and the UC Davis Plasma Diagnostics Group directed by Prof. Neville C. Luhmann, Jr. Each code in the suite required 1-2 man months to develop. Small changes and refinements continue to be made. An attempt was made to import the full wave two-dimensional code AMOS from another division at LLNL; however, incompatibilities in the source code were identified and there was inadequate support from the authors to make a success of AMOS. After a period of considerable frustration, we opted to write our own simple codes using the BASIS system due to Paul Dubois to streamline the code development and production use of the resulting codes. The codes use standard finite-difference methods such as are described in textbooks like Richtmyer and Morton, e.g., Crank-Nicholson integration schemes used in the full-wave ultra-short-pulse codes, and operator splitting and spectral (fast Fourier transform) methods in SOFTSTEP. The one-dimensional codes are fast-running codes that can execute on good workstations in 30 minutes or less. The two-dimensional codes execute in 30 minutes or less on the C90 at NERSC for typical grid resolutions 2000x1500 and 2000 time steps.

HELM1D is a one-dimensional Helmholtz equation solver for monochromatic reflectometry that has been extended to accommodate exponentially tapered profiles and to model the extraordinary (X) mode as well as ordinary (O) mode. SOFTSTEP 1D and 2D are time-dependent codes using a slowly varying envelope approximation which has been tested and applied to the basic laboratory plasma reflectometry experiment. A three-dimensional massively parallel version of SOFTSTEP has been begun which includes new coherence and correlation function diagnostics. We have also developed a suite of fully time-dependent one-dimensional (Cohen, et al. 1995) and two-dimensional O-mode and X-mode codes that have been used to model laboratory ultra-short-pulse reflectometry experiments.

SOFTSTEP 1D was used to show the fundamental difference in effects produced by fluctuations at the cut-off layer that produce amplitude and phase changes in the reflected signal and fluctuations at Bragg resonance that produce only a phase shift in the reflected signal. This had a bearing on interpretations of the data from the laboratory plasma reflectometry experiment. HELM1D was used to study further Bragg resonance effects on reflectometry and the departures of the scattering from the description given using the first Born approximation (Afeyan, et al., 1995; Chou, et al., 1995).

Ultra-short-pulse reflectometry was studied with the suite of full-wave one- and two-dimensional O-mode and X-mode codes. Density-profile reconstructions were modeled, and their robustness in the presence of large-amplitude density perturbations was studied to determine operating limits on the reconstructions. With X-modes, magnetic

profiles were reconstructed in addition to density profiles. Furthermore, it was demonstrated that Bragg resonance effects due to density perturbations could be identified in the reflected signals in addition to and distinct from the signals from cut-off layers that were used for the profile reconstruction, thus providing data on the density perturbations. A more robust and reliable method for determining the dispersion of the group delays used in the reconstructions was implemented in the suite of ultra-short-pulse codes based on a windowed fast Fourier transform (a spectral method). The spectral method is analogous to the laboratory techniques, allowed us to accurately reconstruct a wider variety of plasma density profiles, and was reliable in the presence of a higher amplitude density perturbation than was the case using the zero-crossing method in our earlier calculations. We have also included an embedded conducting surface in our two-dimensional computations to model a waveguide launcher and detector more closely resembling the experiments, and have obtained good reconstructions using the windowed FFT method. We have explored resolution issues relevant to profile reconstructions in the experiments which give guidance on how many frequency channels should be used in the detectors.

Code benchmarking and comparisons to analytical theory (each code was compared quite carefully in several test cases against analytical theory of wave propagation in one or two dimensions as applicable ) and experiment (selective) are described in the following publications:

B. B. Afeyan, et al. (1994), Proceedings of the Second IAEA Topical Workshop on Plasma Reflectometry, Princeton, NJ, Jan. 1994.

B. B. Afeyan, A. E. Chou, and B. I. Cohen (1995), Plasma Phys. Control. Fusion 37, 315.

A. E. Chou, et al. (1992), Rev. Sci. Instrum. 63, 4669.

A. E. Chou, et al. (1995), Rev. Sci. Instrum. 66, 1216.

B. I. Cohen, B. B. Afeyan, A. E. Chou, and N. C. Luhmann, Jr. (1995), Plasma Phys. Control. Fusion 37, 329.



*Technical Information Department · Lawrence Livermore National Laboratory  
University of California · Livermore, California 94551*

