

LSC Users Manual

David W. Ignat
Princeton Plasma Physics Laboratory
and
Aaron J. Redd
Lehigh University Physics Department

January 1996

Acknowledgments

This user's manual and the development of the computer program LSC was funded by the United States Department of Energy, Office of Fusion Energy, under contract DE-AC02-76-CHO-3073. The computer code and the interfacing with TSC was done by D. W. Ignat, E. J. Valeo, and S. C. Jardin. Interface with TRANSP was done by D. W. Ignat, T. B. Terpstra, D. C. McCune, and S. M. Kaye.

A. H. Kritz guided and supported A. J. Redd in his work on many of the details of this manual.

F. W. Perkins devised the approach taken in the consideration of scattering. M. Ono, F. Paoletti, and H. Takahashi contributed discussions regarding the limited up-shift of parallel index of refraction. S. von Goeler, S. E. Jones, and D. P. Enright provided valuable help regarding the x-ray camera. V. Fuchs encouraged the approximate treatment of rf current diffusion.

Operation of the code by M. Shoucri (of the Tokamak de Varennes) in conjunction with TSC and by S. M. Kaye, R. V. Budny (of PPPL) and P. Stubberfield (of JET) in conjunction with TRANSP lead to improvements.

S. Bernabei and M. Okabayashi supported, encouraged, and advised this project over the past several years.

This users manual is patterned after the "TSC Users Manual" prepared by William Daughton under the guidance of S. C. Jardin and N. Pomphrey.

February 3, 1998

Contents

1	Introduction	3
2	Theoretical Background	5
2.1	Geometrical Configuration	5
2.1.1	Wave Propagation	5
2.1.2	Fokker-Planck Equation	9
2.1.3	RF-Driven Current	12
2.1.4	Broadening of rf-Driven Current	14
2.1.5	Heuristic Power Diffusion Estimate	16
2.1.6	X-ray Camera Image	16
3	LSC Usage	18
3.1	Preliminaries	18
3.2	Description of Parameters	21
3.2.1	Ray and Quasilinear Parameters	22
3.2.2	X-ray Camera Parameters	23
3.3	Description of Input	23
3.3.1	File “input.lhh”	24
3.3.2	File “input.xry”	33
3.3.3	NAMELIST Files	36
3.3.4	Output Files	37
3.3.5	Output Plots and Tables	38
3.4	Makefiles	43
3.4.1	Sun Makefile	43
3.4.2	Cray Makefile	44

Chapter 1

Introduction

The Lower hybrid Simulation Code (LSC) is a computational model of lower hybrid current drive in the presence of an electric field.[1] Details of geometry, plasma profiles and circuit equations are treated. Two-dimensional velocity space effects are approximated in a one-dimensional Fokker-Planck treatment.[2]

The LSC was originally written to be a module for lower hybrid current drive called by the Tokamak Simulation Code (TSC), which is a numerical model of an axisymmetric tokamak plasma and the associated control systems.[3] The TSC simulates the time evolution of a free boundary plasma by solving the MHD equations on a rectangular computational grid. The MHD equations are coupled to the external circuits (representing poloidal field coils) through the boundary conditions. The code includes provisions for modeling the control system, external heating, and fusion heating.

The LSC module can also be called by the TRANSP[4, 5, 6] code. TRANSP represents the plasma with a axisymmetric, fixed-boundary model and focuses on calculation of plasma transport to determine transport coefficients from data on power inputs and parameters reached.

This manual covers the basic material needed to use the LSC. If run in conjunction with TSC, the “TSC Users Manual” should be consulted.[8] If run in conjunction with TRANSP, on-line documentation will be helpful.

A theoretical background of the governing equations and numerical methods is given in Chapter 2. Information on obtaining, compiling and running the code is given in Chapter 3.

This manual can be found:

1. on the **PPPL Sun** network in `~ignat/tex/man/` in a file with a name

like lscman.tex ;

2. in the **NERSC CFS** (National Energy Research Super Computer common file system) in area /004554/man/ with a name like lscman-jan95.tar. The tarfile, created with the unix **tar** utility, puts creation date along with all text into the archive. Read permission has been granted to all users.

The LSC code can be found:

1. on the PPPL Sun network in ~ignat/lsc/ as a collection of fortran files (extension .F) which depend on included files (extension .inc) in directory ~ignat/lsc/inc/ .
2. on the NERSC CFS in area /004554/for/lsc/ .
3. on the PPPL Sun network in /soft/TRANSP-LSC/ and Vax network in ignat\$:[TRANSP] . These files have been processed with parameters suitable for TRANSP.

Chapter 2

Theoretical Background

The following description of the governing equations and numerical methods closely follows the original paper describing the LSC model.[1]

2.1 Geometrical Configuration

A cylindrical coordinate system with R as the radial coordinate, Z as the axial coordinate and ϕ as the toroidal symmetry angle is used. In LSC this triple is thought of in as (R, Z, ϕ) , whereas in TSC this is considered (R, ϕ, Z) . In other words, there can be confusion over the ϕ direction.

For LSC and TSC the poloidal flux is minimum in the center of the plasma and rises toward the plasma boundary. For LSC the B_Z at the outer midplane is positive for a positive current.

2.1.1 Wave Propagation

We assume that the plasma varies sufficiently slowly on a wave period and a wave length so that the WKB approximation is valid. We therefore assume that the wave electric field can be decomposed into a set of components indexed by j ,

$$\mathbf{E} = \sum_j \mathbf{E}_j(\mathbf{r}) \exp i\Phi_j, \quad (2.1)$$

where the rapid space and time variation occurs through the exponential

$$\Phi_j \equiv \int \mathbf{k}_j(\mathbf{r}) \cdot \mathbf{dr} - \omega t, \quad (2.2)$$

which depends on the local wave vector \mathbf{k}_j and the frequency ω . Additional time behavior is ignored because the transit time of a wave is short compared

to the time scale of plasma evolution. The mode amplitudes \mathbf{E}_j vary much less rapidly. Suppressing the index j , each mode locally satisfies the matrix equation,[7]

$$[\mathbf{k}\mathbf{k} - \mathbf{I}k^2 + \kappa_0^2 \mathbf{K}(\mathbf{r}, \mathbf{k}, \omega)] \cdot \mathbf{E} = 0 \quad (2.3)$$

where $\kappa_0 = \omega/c$ represents the free-space wavenumber, and \mathbf{K} is the dielectric tensor, which has a nontrivial solution for a vanishing determinant

$$\epsilon(\omega, \mathbf{k}, \mathbf{r}) \equiv |[\mathbf{k}\mathbf{k} - \mathbf{I}k^2 + \kappa_0^2 \mathbf{K}(\mathbf{r}, \mathbf{k}, \omega)]| = 0 \quad . \quad (2.4)$$

The dispersion relation (2.4) connects the vector \mathbf{k} and frequency ω at the spatial location \mathbf{r} (explicit time variation of the dielectric properties are assumed negligible from here on). To proceed it is convenient to choose a local Cartesian coordinate system[7], such that $\hat{\mathbf{z}} \times \mathbf{B} = 0$, and \mathbf{k} is contained in the $x - z$ plane:

$$\mathbf{k} = k_{\parallel} \hat{\mathbf{z}} + k_{\perp} \hat{\mathbf{x}}. \quad (2.5)$$

For lower hybrid waves, the ion cyclotron frequency is much less than the wave frequency which is much less than the electron cyclotron frequency: $\omega_{ci} \ll \omega \ll \omega_{ce}$, an inequality assumed throughout. In the Hermitian part of the dielectric tensor we keep only cold plasma terms, except that the dominant warm-plasma term is carried to guard against singularity near the lower hybrid resonance.

The anti-Hermitian part of the tensor is retained as a perturbation. For the case at hand, the principal such term enters as an imaginary correction to K_{zz} , and describes the interaction between the component of the wave electric field parallel to \mathbf{B} and electrons whose speed along \mathbf{B} matches that of the wave (Landau damping). Thus, the plasma dielectric behavior is described by the following tensor elements, all other elements being zero:

$$K_{xx} = K_{yy} = S - \alpha k_{\perp}^2 \quad , \quad (2.6)$$

$$-K_{yx} = K_{xy} = iD = i \frac{\omega_{pe}^2}{\omega \omega_{ce}} \quad , \quad (2.7)$$

$$K_{zz} = P + iK_{zz,i} \quad , \quad (2.8)$$

where

$$S = 1 + \frac{\omega_{pe}^2}{\omega_{ce}^2} - \sum_j \frac{\omega_{pi,j}^2}{\omega^2} \quad , \quad (2.9)$$

$$\alpha = \frac{3}{4} \frac{\omega_{pe}^2}{\omega_{ce}^4} v_{Te}^2 + 3 \sum_j \frac{\omega_{pi,j}^2 v_{Ti,j}^2}{\omega^4} \quad , \quad (2.10)$$

$$P = 1 - \frac{\omega_{pe}^2}{\omega^2} , \quad (2.11)$$

$$K_{zz,i} = -\pi \frac{\omega_{pe}^2}{\omega} \int dv_{\parallel} v_{\parallel} \frac{\partial f_e}{\partial v_{\parallel}} \delta(\omega - k_{\parallel} v_{\parallel}) , \quad (2.12)$$

and the new symbols are as follows: ω_{pe} is the electron plasma frequency $n_e e^2 / (\epsilon_0 m_e)$, $\omega_{pi,j}$ is the ion plasma frequency of the j^{th} species, v_{Te} and $v_{Ti,j}$ are electron and ion thermal velocities ($\equiv \kappa T / m$), and $f_e(v_{\parallel})$ is a one-dimensional electron velocity distribution function normalized such that

$$\int dv_{\parallel} f_e(v_{\parallel}) = 1 . \quad (2.13)$$

The permittivity of free space is ϵ_0 . The thermal term designated by α would be important near lower hybrid resonance, $S = 0$. However, in cases treated in this paper S does not approach zero, and α is not important. This would be true whenever $\omega^2 \gg \omega_{ce} \omega_{ci}$. The wave-particle interaction responsible for electron heating and current drive is in $K_{zz,i}$. In the event the lower hybrid resonance should become important, a thermal term representing the ion wave-particle interaction would have to be added to K_{xx} and K_{yy} , but here we assume such terms vanish.

In the coordinate system of equation (2.5), we decompose the dispersion relation (2.4) into its real and imaginary parts using equations (2.6) – (2.12),

$$\epsilon = \epsilon_r + i\epsilon_i = 0 , \quad (2.14)$$

where

$$\begin{aligned} \epsilon_r = & -\alpha k_{\perp}^6 + k_{\perp}^4 S + k_{\perp}^2 [(P + S)(k_{\parallel}^2 - \kappa_0^2 S) + \kappa_0^2 D^2] \\ & + P [(k_{\parallel}^2 - \kappa_0^2 S)^2 - \kappa_0^4 D^2] , \end{aligned} \quad (2.15)$$

$$\epsilon_i = \frac{\partial \epsilon_r}{\partial P} K_{zz,i} . \quad (2.16)$$

If $n \equiv kc/\omega \gg 1$ a simplified dispersion relation can be found from the matrix equation leading up to equation (2.4) by asymptotically expanding

$$\begin{aligned} \hat{\mathbf{n}} & \sim \hat{\mathbf{n}}^0 + \frac{1}{n^2} \hat{\mathbf{n}}^1 + O\left(\frac{1}{n^4}\right), \\ \mathbf{E} & \sim \mathbf{E}^0 + \frac{1}{n^2} \mathbf{E}^1 + O\left(\frac{1}{n^4}\right). \end{aligned} \quad (2.17)$$

To lowest order, $\mathbf{I}_\perp \cdot \mathbf{E}^0 = 0$, where $\mathbf{I}_\perp = \mathbf{I} - \hat{\mathbf{n}}\hat{\mathbf{n}}$, so that $\mathbf{E}^0 = \hat{\mathbf{n}}^0 E^0$. In first order, there arises the solubility condition

$$\hat{\mathbf{n}}^0 \cdot \mathbf{K} \cdot \hat{\mathbf{n}}^0 = 0, \quad (2.18)$$

which gives the electrostatic limit of the dispersion relation:

$$\epsilon_r = -\alpha k_\perp^4 + k_\perp^2 S + k_\parallel^2 P \quad . \quad (2.19)$$

The consistency condition $|\mathbf{E}^1| = |\mathbf{K} \cdot \mathbf{E}^0| < n^{-2} |\mathbf{E}^0|$ is satisfied for $n_\parallel^2 > 1 + (\omega_{pe}^2/\omega_{ce}^2)$.

The extension of the local solutions to a spatially inhomogeneous plasma is accomplished by the eikonal method with the useful result that an initial wave field at \mathbf{r} with an initial propagation vector \mathbf{k} evolves according to Hamiltonian equations which preserve the local dispersion relation $\epsilon_r = 0$ along the ray trajectories

$$\frac{d\mathbf{r}}{dt} = -\frac{\partial \epsilon_r}{\partial \mathbf{k}} / \frac{\partial \epsilon_r}{\partial \omega} \quad (2.20)$$

$$\frac{d\mathbf{k}}{dt} = \frac{\partial \epsilon_r}{\partial \mathbf{r}} / \frac{\partial \epsilon_r}{\partial \omega} \quad (2.21)$$

As in Hamiltonian mechanics, the spatial coordinates denoted by \mathbf{r} are canonically conjugate to the wave number coordinates denoted by \mathbf{k} . For example, in a Cartesian frame one would have (x, y, z) and (k_x, k_y, k_z) and fields varying as $\exp i \int (k_x dx + k_y dy + k_z dz)$; and in a cylindrical frame one would have (R, Z, ϕ) and (k_R, k_Z, l) where R, Z have dimensions of length, k_R, k_Z have dimensions of inverse length (wave number), and l is the dimensionless toroidal mode number. In the study of the axisymmetric tokamak, it is the cylindrical coordinate system that is the most natural since the canonical momentum l is constant along the ray path ($dl/dt \propto \partial \epsilon_r / \partial \phi = 0$).

A spectral component of power W experiences a change in power ΔW over time interval $\Delta \tau$:

$$\begin{aligned} \Delta W &= -2 \epsilon_i / \left(\frac{\partial \epsilon_r}{\partial \omega} \right) W \Delta \tau \\ &= -2 \frac{\partial \epsilon_r}{\partial P} K_{zz,i} / \left(\frac{\partial \epsilon_r}{\partial \omega} \right) W \Delta \tau \\ &= 2\pi \frac{\omega_{pe}^2}{\omega} \int dv_\parallel v_\parallel \frac{\partial f_e}{\partial v_\parallel} \delta(\omega - k_\parallel v_\parallel) \frac{\partial \epsilon_r}{\partial P} / \left(\frac{\partial \epsilon_r}{\partial \omega} \right) W \Delta \tau \quad .(2.22) \end{aligned}$$

Note that equations (2.20), (2.21), and (2.22) also carry an index numbering the individual spectral component, which has been suppressed to improve readability. Given the velocity distribution and the profiles of macroscopic plasma parameters, the absorption of a lower hybrid spectrum can be computed. An actual incident wave spectrum is a continuous function of the parallel wave number. Computationally, this continuous spectrum is approximated by assigning the input power to a number of discrete rays as in equation (2.1), each ray having a definite initial k_{\parallel} and launched power. The number of rays and their distribution over k_{\parallel} can be chosen to adequately resolve the spectrum in cases of reasonably strong single-pass damping. On the other hand, if many passes are required to absorb the power, a very dense packing of rays into the spectrum might be required to obtain well behaved results.

The plasma is divided up into discrete shells between flux surfaces, typically 20 – 100. For each ray, equations (2.20) through (2.22) allow calculation of the needed quantities. The width of the shells and the detailed trajectory determine $\Delta\tau$ in equation (2.22). As a practical matter, one cannot ensure that the shells are fine enough to keep $|\Delta W|/W < 1$, so this quantity must be monitored and $\Delta\tau$ adjusted downward the amount necessary to keep $|\Delta W| < W$. The new $\Delta\tau$ is recorded for use in equation (2.25) of section 2.1.2.

2.1.2 Fokker-Planck Equation

Quasilinear Diffusion Coefficient

An incremental contribution to the quasilinear diffusion coefficient D_{ql} at velocity v_{\parallel} from a wave field of wave number k_{\parallel} is given by

$$D_{ql}(v_{\parallel}) = \frac{\pi}{2} \left(\frac{e}{m_e} \right)^2 E_{\parallel}^2 \delta(\omega - k_{\parallel}v_{\parallel}), \quad (2.23)$$

where E_{\parallel} represents the amplitude of the wave field parallel to the local static magnetic field, and the electron charge and mass are e and m_e . [7] One instructive way to find the relationship between field E_{\parallel} and wave power W is to equate P_{ql} , the energy per unit time per unit volume going into electrons and out of the wave from the quasilinear point of view,

$$P_{ql} = -n_e m_e \int dv_{\parallel} v_{\parallel} D_{ql}(v_{\parallel}) \frac{\partial f_e}{\partial v_{\parallel}} \quad (2.24)$$

with the similar quantity from the ray point of view as assembled from equations (2.12), (2.16), (2.22) to obtain

$$D_{ql}(v_{\parallel}) = 2 \left(\frac{\pi}{\epsilon_0} \right) \left(\frac{e}{m_e} \right)^2 W \frac{\partial \epsilon_r / \partial P}{\omega \partial \epsilon_r / \partial \omega} \left(\frac{\Delta \tau}{\Delta V} \right) \delta(\omega - k_{\parallel} v_{\parallel}) \quad (2.25)$$

for the incremental D_{ql} from a wave of power W traversing a flux shell of volume ΔV in time $\Delta \tau$.

Kinetic Equation

An electron kinetic equation can be written

$$\frac{\partial f_e}{\partial t} = \left(\frac{\partial f_e}{\partial t} \right)_c + \left(\frac{\partial f_e}{\partial t} \right)_w \quad (2.26)$$

Here the $(\)_c$ term is the Coulomb collision operator, and the $(\)_w$ term is the wave diffusion (quasilinear) operator. There is no term involving a steady electric field. The wave diffusion operator is the one-dimensional divergence of the rf-induced flux:

$$\left(\frac{\partial f_e}{\partial t} \right)_w = \frac{\partial}{\partial v_{\parallel}} D_{ql}(v_{\parallel}) \frac{\partial f_e}{\partial v_{\parallel}}, \quad (2.27)$$

where now the term D_{ql} signifies a sum over all waves in existence on a flux surface, with the appropriate powers and velocities. Note that the use of a simple sum means that we assume there are no interference effects.

We employ a one-dimensional collision operator of the form,

$$\left(\frac{\partial f_e}{\partial t} \right)_c = \frac{\partial}{\partial v_{\parallel}} \left[\left(D_c(v_{\parallel}) \frac{\partial}{\partial v_{\parallel}} + \nu_c(v_{\parallel}) v_{\parallel} \right) f_e(v_{\parallel}) \right]. \quad (2.28)$$

The collisional diffusion and drag coefficients are given by

$$D_c(v_{\parallel}) = \beta_Z \Gamma / v_{Te} [1 + (v_{\parallel} / v_{Te})^2]^{-3/2}, \quad (2.29)$$

$$\nu_c(v_{\parallel}) = \beta_Z \Gamma / v_{Te}^3 [1 + (v_{\parallel} / v_{Te})^2]^{-3/2}, \quad (2.30)$$

where $\Gamma = \ln \Lambda n_e e^4 / (4\pi \epsilon_0^2 m_e^2)$ and the normalization coefficient β_Z is chosen to yield the correct value for the electrical conductivity in the absence of rf. Approximately, $\beta_Z = (1+Z)/5$, with Z the effective ion charge. The collision operator has several correct properties, such as conservation of particles,

producing a Maxwellian (with thermal velocity v_{Te}) in the absence of wave power, and correct asymptotic velocity dependence of the coefficients for high speeds $v \gg v_{Te}$.

In solving for f_e we set $\partial/\partial t = 0$ because the time for equilibration between rf power and the electron distribution is short compared to the time for plasma to evolve. Then the solution for f_e is an integral in velocity space.

$$f_e(v_{\parallel}) = \frac{1}{\sqrt{2\pi v_{Te}^2}} \exp\left(-\int_0^{v_{\parallel}} \frac{\nu_c(v') v' dv'}{D_c(v') + D_{ql}(v')}\right) \quad (2.31)$$

Some Numerical Details

Our calculation requires a discrete grid in v_{\parallel} indexed by i_v and many discrete rays indexed by i_r which at each flux shell crossing indexed by i_z have a definite ω/k_{\parallel} and a power W at a definite location in the plasma indexed by i_p . The delta function $\delta(\omega - k_{\parallel} v_{\parallel})$ connects the discrete velocity quantity with the discrete wave number quantity and therefore must be interpreted to have a width that causes smooth variations in velocity while the spectrum of waves, represented by a limited number of rays, is not smooth. The width of the delta function in velocity should be such that for two nearby rays separated by Δk_{\parallel} the range in velocity Δv_{\parallel} is approximately the relative separation of two nearby rays $\Delta k_{\parallel}/k_{\parallel}$ times the phase velocity ω/k_{\parallel} . However, some judgment should enter the actual selection of the width of the smoothing function because k_{\parallel} and therefore Δk_{\parallel} can evolve considerably. If the smoothing range is too narrow in velocity, then non-physical “holes” can appear in the D_{ql} with a resulting ragged behavior of f_e . On the other hand, if the smoothing range is too great, then a wave of phase speed ω/k_{\parallel} interacts with electrons of much different parallel velocity v_{\parallel} , especially much lower parallel speed, since there are many more low speed electrons than high speed electrons. This can lead to spurious power absorption.

The rays do not move monotonically in “radius” (which in our calculation is measured by poloidal flux ψ) and in fact typically make many in-out changes of direction. To compute needed quantities at each flux surface, the motion of ray number i_r must be recorded by a zone number i_z , which is sequential with the intersection of that ray with any flux surface. For each zone index, the index i_p of the ψ surface intersected is kept, where i_p is 1 at the center of the plasma, increasing toward the edge. For each trajectory the i_z starts at 1 and increases to a maximum value; but i_p starts near its maximum, decreases initially, and exhibits individual behavior thereafter.

Changes in direction can be of several types. An inward-going ray can change to an outward-going ray, or vice-versa, because:

1. the roots of the dispersion relation converge (mode conversion);
2. speaking by analogy to a mechanical Hamiltonian system, the radial penetration limit is reached owing to the impenetrability of a centrifugal barrier;
3. the root of the dispersion relation approaches zero (cutoff);
4. the ray is captured and specularly reflected near the plasma edge. This is accomplished by logic in the calculation which senses that a cutoff is approaching.

The quantities $D_{ql;i_v,i_p}$, $f_{e;i_v,i_p}$, W_{i_z,i_r} are all made consistent by iteration, beginning with very small $W_{i_z=1,i_r}$ and ramping up toward the full power launched. We use an under-relaxation algorithm in the iteration of f_e , in that the past f_e is averaged with the newly computed f_e to get the new function propagated to the next cycle.

2.1.3 RF-Driven Current

We calculate the current driven on each flux surface according to our equation (2.32) which follows the prescription given in equation (21b) of reference (2), except that we drop the term arising from a non-zero runaway probability, obtaining

$$J_{\text{rf}} = \frac{-en_e}{\nu_r} \int dv_{\parallel} D_{ql}(v_{\parallel}) \frac{\partial f_e(v_{\parallel})}{\partial v_{\parallel}} \cdot \frac{\partial W_s(u)}{\partial u} \quad . \quad (2.32)$$

In the above, $\nu_r = \Gamma/|v_r|^3$, $u = v_{\parallel}/v_r$, and $v_r = -\text{sign}(eE_{dc})\sqrt{m_e\Gamma/|eE_{dc}|}$, as explained in the reference.[2] These definitions give as J_{rf} the current density of stopped electrons, and use the function $W_s(u)/u$ which is given as a tabulation of coefficients fitting simple algebraic terms to solution of the complete Fokker-Planck equation in two velocity dimensions (v_{\perp} and v_{\parallel}).

The key quantity is $W_s(u)$, the energy (normalized to $m_e v_r^2/2$) imparted to the electric field E_{dc} by an electron as it slows down. The local electric field E_{dc} is either prescribed for a static simulation (usually set to zero) or supplied by TSC as part of an iteration to be described.

For very small electric field, $W_s(u)$ can be represented in a power series. From the first two terms, one finds

$$J_{\text{rf}} = \frac{-en_e}{\Gamma} \int dv_{\parallel} D_{\text{qt}}(v_{\parallel}) \frac{\partial f_e(v_{\parallel})}{\partial v_{\parallel}} \frac{v_{\parallel}^3}{5+Z} \left[\mu - \frac{1+Z/2+3\mu^2/2}{3+Z} \frac{v_{\parallel}^2}{v_r^2} \right], \quad (2.33)$$

where $\mu = -1$ for cooperative current drive, and $\mu = +1$ for current drive into an opposing field.

The function $W_s(u)$ behaves in such a way that for u approaching -1 , corresponding to rf current drive cooperating with the ohmic current driven by the ambient electric field E_{dc} at a velocity near the runaway velocity, the calculated current grows extremely large. Because of the runaway condition, this is to be expected. Whenever $u \approx -1$, the rf-driven current is not properly calculable from our approach.

It is often necessary to iterate between TSC and LSC for the field and current at the exact time that LSC is called for power and current information, as follows. Suppose at time $(n-1)$ the local total current density is $J^{(n-1)}$, the rf-driven current density is $J_{\text{rf}}^{(n-1)}$, and the local electric field is $E_{dc}^{(n-1)}$. We want to advance quantities to the new time, (n) , and be consistent with Ohm's Law: $J = \sigma E_{dc} + J_{\text{rf}}$, where σ is the parallel neoclassical conductivity. An initial guess at what the new electric field $E_{dc}^{(n-)}$ must be is formed from the new total current, as advanced by TSC, and the old rf-driven current and old electric field.

$$J^{(n)} = \sigma E_{dc}^{(n-)} + J_{\text{rf}}^{(n-1)}(E_{dc}^{(n-1)}) \quad , \quad (2.34)$$

From the $E_{dc}^{(n-)}$ obtained from equation (2.34) one can find the estimated new rf current density $J_{\text{rf}}^{(n-)}$ from equation (2.32). Its derivative with respect to E_{dc} , $(\partial J_{\text{rf}}/\partial E_{dc})^{(n-)}$, is formed by numerical differentiation. Algebra yields estimates for the new electric field and for the new rf-driven current density,

$$E_{dc}^{(n)} = E_{dc}^{(n-)} + \frac{J_{\text{rf}}^{(n-1)} - J_{\text{rf}}^{(n-)}}{\sigma + (\partial J_{\text{rf}}/\partial E_{dc})^{(n-)}} \quad , \quad (2.35)$$

$$J_{\text{rf}}^{(n)} = J_{\text{rf}}^{(n-)} + \frac{(\partial J_{\text{rf}}/\partial E_{dc})^{(n-)} (J_{\text{rf}}^{(n-1)} - J_{\text{rf}}^{(n-)})}{\sigma + (\partial J_{\text{rf}}/\partial E_{dc})^{(n-)}} \quad . \quad (2.36)$$

The new values $E_{dc}^{(n)}$ and $J_{\text{rf}}^{(n)}$ may not be consistent. This is checked by assigning them to the previous step labeled $(n-1)$ and repeating the process

until the result is stable to a chosen accuracy. At that point, it is desired that the electric field is small enough that the runaway situation does not exist for electrons interacting with the waves. An inequality expressing this condition, as suggested by equation (20) of reference 2, is

$$\frac{\tau_{r-\text{loss}} \dot{n}_{er}}{n_e} = \tau_{r-\text{loss}} D_{ql}(v_r) \left| \frac{\partial f_e(v_r)}{\partial v_{\parallel}} \right| \ll 1 \quad (2.37)$$

where $\tau_{r-\text{loss}}$ is a loss time for electrons pushed by rf waves into a runaway region and \dot{n}_{er} is rate of increase of the density of runaway electrons owing to the rf diffusion from a lower velocity. A simple estimate for $\tau_{r-\text{loss}}$ would be the confinement time (~ 10 msec). If equation (2.37) is not satisfied, then the calculation is inappropriate.

Iteration is not necessary if when the ambient electric field is low and the density is high, such that $\omega \ll |v_r k_{\parallel}|$. In that case the evolution of the electric field and the current is handled without special attention as the transport calculation evolves.

2.1.4 Broadening of rf-Driven Current

The rf-driven current is subject to diffusion across the confining magnetic field. Magnetic turbulence is often proposed as the dominant mechanism.

Diffusing a velocity distribution is beyond the scope of the LSC, but we have introduced a heuristic approach similar to that used by V. Fuchs and others.[9]

The idea is that the actual, diffused, RF-driven current J_d is modified from the computed RF-driven current J_{rf} by slowing-down, characterized by an inverse-time ν_{slow} , and a cross-field diffusion, characterized by D_{Jrf} .

Assuming for the moment slab geometry with x the spatial variable the diffusion-like equation we use is:

$$\partial J_d / \partial t = \nu_{\text{slow}} (J_{\text{rf}} - J_d) + \partial / \partial x (D_{\text{Jrf}} \partial J_d / \partial x) \quad (2.38)$$

This can be transformed to a normalized flux coordinate $\hat{x} \equiv (\psi - \psi_{\text{min}}) / (\psi_{\text{max}} - \psi_{\text{min}})$ with standard transformations for curvilinear coordinates. Under the reasonably good assumption that $\psi \sim r^2$ the result is:

$$\partial J_d / \partial t = \nu_{\text{slow}} (J_{\text{rf}} - J_d) + (4/a^2) \partial / \partial \hat{x} (\hat{x} D_{\text{Jrf}} \partial J_d / \partial \hat{x}) \quad , \quad (2.39)$$

where a represents the nominal minor radius of the plasma.

Equation 2.39 clearly has desirable properties: if D_{Jrf} is small then the J_{d} moves to J_{rf} on the ν_{slow} time scale; and if D_{Jrf} is large then J_{d} and J_{rf} can be quite different, in particular J_{d} will be smooth compared to J_{rf} over a length of order $\sqrt{D_{\text{Jrf}}/\nu_{\text{slow}}}$.

In LSC D_{Jrf} on input is one number, constant across the plasma, in units of meter² per second. Internally, D_{Jrf} is an array, so generalization would be simple. The ν_{slow} are found from equation 2.30 for a particular phase speed, corresponding to $n_{\parallel} = 2$, also constant across the plasma. The diffusion process is imagined to be complete at each call to LSC, so that the time derivative is set to zero. Under that condition, the solution for J_{d} comes from one inversion of a tridiagonal matrix.

The boundary condition on J_{d} at the outer boundary is a zero value. At the inner boundary, the center, the condition of zero flux (zero derivative with respect to radius) translates to the derivative with respect to \hat{x} being better behaved than $1/\sqrt{\hat{x}}$, which cannot be expressed in a manner practical for the finite difference equations. Instead, we force $dJ_{\text{d}}/d\hat{x}$ to be constant over the first two grid spacings. This is equivalent to specifying a zero value for a certain linear combination of J_{d} and $dJ_{\text{d}}/d\hat{x}$ at the first interior grid point.

The rf-driven current found J_{rf} leads directly to the smoothed (diffused) J_{d} after the inversion of a tri-diagonal matrix.[10]

Note that the integral of J_{d} over the plasma cross section, or total diffused rf-driven current, is not constrained to have any particular relationship with the same integral over J_{rf} , the total undiffused current. In fact, it seems reasonable that diffusion can in some circumstances increase current by moving fast electrons to regions of lower collision rate. However, it appears that the diffused current is less than the undiffused current for many actual situations, typically having deposition at mid-radius and mildly peaked density profile.

The heuristic smoothing method of V. Fuchs[9] is similar to the method described here, but apparently treats D_{Jrf} and ν_{slow} as constants, sets the change in total current from an estimate incorporating the formula for ν_{slow} , and finds the value needed for D_{Jrf} as an eigenvalue.

Our model is not a simulation of energetic electron diffusion in that it does not transport fast particles or change the rf wave damping because of their motion.

2.1.5 Heuristic Power Diffusion Estimate

The undiffused current J_{rf} was computed from the ray tracing, quasi-linear development of an electron distribution function in the parallel velocity giving the local power deposition per unit volume P_0 , and finally the Karney-Fisch[2] electric field adjustment to current density.

As explained in the previous section, the diffused current density J_d can be quite different from J_{rf} , and therefore quite different in radial distribution from the P_0 found from ray tracing. This situation contrasts with the intuition that power density is proportional to current density times background number density. Therefore, we added an option to spread the deposited rf power according to

$$P(\hat{x}) = \alpha \frac{J_d(\hat{x})n_e(\hat{x})}{\int J_d(\hat{x})n_e(\hat{x}) dV(\hat{x})} \int P_0 dV + (1 - \alpha)P_0(\hat{x}) \quad , \quad (2.40)$$

where P is the diffused power, P_0 is the power deposited from the ray tracing and quasilinear calculation, α ranges from 0 (no spreading of P_0) to 1 (full spreading), dV is a volume element.

As far as we understand, spreading of current in the past[9] has not been accompanied by a spreading of the power.

2.1.6 X-ray Camera Image

The PBX-M experiment to which we have applied our calculation the most has a 2-dimensional x-ray camera with an axis roughly tangent to a mid-plane circle just inside the major radius of the plasma. The images found by this camera reveal clues to the behavior of the fast electrons in the plasma. At the same time, such images can be computed from the electron velocity distribution we establish in our model.

Taking account of obstructions, the actual location of pinhole, focal plane, etc., we evaluate the 2-d image with the following formula.

$$\begin{aligned} I_{x-p} = & \frac{1}{dA_p} \int dl dA(l) d\Omega_p(l) n_i(\psi(l)) n_e(\psi(l)) \times \\ & \int dv_{\parallel} v_{\parallel} f_e(v_{\parallel}, \psi(l)) \times \\ & \int_0^{E(v_{\parallel})} dE' E' \frac{\partial^2 \sigma_{ei}}{\partial E' \partial \Omega} [E(v_{\parallel}), E', \hat{\mathbf{n}} \cdot \hat{\mathbf{b}}] T(E') \quad (2.41) \end{aligned}$$

In the above, I_{x-p} is the x-ray intensity on pixel p of area dA_p at the camera focal plane, dl is an element of path length in the plasma on a

line passing through the pinhole and the pixel, dA is an area element for emission into the pixel p , $d\Omega_p$ is the solid angle subtended by the pinhole as seen from the emitting volume element, $n_{i(e)}$ is the ion (electron) density at the emitting location, E' denotes photon energy emitted by an electron of energy E , $\partial^2\sigma_{ei}/\partial E'\partial\Omega$ is the cross section for electron-ion bremsstrahlung differential in energy and solid angle, $\hat{\mathbf{n}}$ is a unit vector connecting the emitting volume with the camera element, and $\hat{\mathbf{b}}$ is a unit vector along the magnetic field, and $T(E')$ is the transmission factor through vessel windows and other absorbers. The differential cross section used in our calculation is that given by formula 2BN of Koch and Motz.[11] The significance of “2” is the differential in energy and angle; “B” is Born approximation; “N” is no screening of the ions. Note that the distribution function f_e is that computed from the one-dimensional Fokker-Planck model, and is itself not influenced by the electric field E_{dc} . The absorption from vessel and windows is computed from fits to experimental absorption data.[12]

Chapter 3

LSC Usage

3.1 Preliminaries

The system dependent details of this section apply to the NERSC system running UNICOS operating system.

The LSC code is written in standard FORTRAN and runs without modification on the Cray-2 and C-90 machines at NERSC, as well as PPPL clusters of Sun SparcStations and VAXes. There are no mathematical library calls; interpolators, integrators, root solvers, look-up searchers, and Bessel function approximations are part of the source code. Routines are taken directly from *Numerical Recipes*[10] are: LAGUERnr, ZROOTSnr, RAN3nr, HUNTnr, PIKSRTnr, and PIKSR2nr, and TRIDAnr. The trailing “nr” denotes the source, where as the preceding characters are the same as in the book.[10] There are, however, graphics calls embedded in the code. At NERSC a “TV80” library must be loaded; on the PPPL Suns, a “SG” library is required.

Source code can be found in the following areas:

1. on the PPPL Sun network in `~ignat/lsc/` as a collection of fortran files (extension `.F`) which depend on included files (extension `.inc`) in directory `~ignat/lsc/inc/` . This is the primary residence of code — changes to source are made here.
2. on the NERSC CFS in area `/004554/for/lsc/` . Read access is restricted.
3. on the PPPL Sun network in `/soft/TRANSP-LSC/` and Vax network

in `ignat$:[TRANSP]` . These files have been processed with parameters suitable for TRANSP.

In addition, `lsc` may be mounted on the NERSC a-machine in `~u4554/lsc` and `~u4554/lsc/inc` . Read access is restricted.

The following is a list of the FORTRAN source files for LSC:

AcDc.F : Main LSC flow control; Block data; etc
Rayini.F : Get an initial solution for ray starting
Rayio.F : Control I/O regarding rays
Raystore.F : Store ray velocity and polarization information
Raytrace.F : Trace the rays
Wr2TSC.F : Write output to TSC; some graphing
XrayCam2.F : Simulated X-ray camera
brambJES.F : Brambilla code, borrowed from J. E. Stevens
cycle.F : Control cycling of f_e , D_{ql} , P_{rf}
fe.F : Relates to $f_e(v_{||}, \psi)$
fits.F : Reproduce Karney-Fisch “fits” of ref[2]
grap.F : Interpolates on a general 2-D grid
grapgrd.F : Drives interpolator for plasma quantities
gridgen.F : Generates grids
grids.F : Contains general purpose grid/array routines
jrf.F : Computes J_{rf}
lscdrive.F : Drives the LSC subroutine in stand-alone mode
matr.F : Contains general matrix routines
pbxio.F : Contains general output routines
plasejv.F : Initializes plasma profiles
power.F : Computes power absorbed
ql.F : Relates to D_{ql}
contr80.F : Plots contour graphs in TV80
contrsg.F : Plots contour graphs in SGlib
ez80.F : EZplot shell over TV80
ezsg.F : EZplot shell over SGlib
pl80.F : Helps contour graphs in TV80
s80.F :
plotsg.F : Helps contour graphs in SGlib
round.F : Rounds to next highest nice value 1, 1.5, 2, 2.5 etc
fedqlout.F : f_e and D_{ql} output routines

Auxilliary code which can be usefule for some purposes includes:

- GGfe.F** : Reads, manipulates, plots a f_e from Gerardo Giruzzi.
WrCircEq.F : Creates a ‘practice’ equilibrium based on a cylinder

There are a number of **INCLUDE** files used for setting parameters and specifying common blocks. The most important of these for the end user is **params.inc** which is used to set the array sizes for the computational grid, poloidal field coils, etc. The settings for these parameters may be quite large, and often may be reduced for the problem at hand.

The files are:

CGSetc.inc	Labels.inc	WkAry.inc
Diags.inc	MKSetc.inc	dielec.inc
Doflags.inc	PIetc.inc	local1d.inc
DqlBins.inc	PIPr.inc	numerics.inc
Escan.inc	ProfBody.inc	params.inc
FeBins.inc	Ramppwr.inc	plcmx.inc
Implic.inc	RayBins.inc	plotf.inc
Inpval.inc	RayWrk.inc	power.inc
Jrf.inc	TSCgrap.inc	tscunits.inc

and, for setting up the X-ray camera:

Xray.inc	emitter.inc	xgraphs.inc
camera.inc	emparams.inc	xparams.inc

The following input files are required:

- input.lhh** : A NAMELIST file of input variables; see Inpval.inc
jardin.d : A highly formatted file of equilibrium plasma information written by TSC or TRANSP prior to the LSC run. Called lhcdoua by TSC.

These input files are not required:

- ray.dat** : Information on ray locations, phase velocities, polarization, etc
input.xry : A NAMELIST file specifying X-Ray camera parameters (optional)

feout.d : $f_e(v_{||}, \psi)$ for special tests (optional)

The code produces the following output files:

Wr2TSC.out : Power deposition and RF-driven current
ray.dat : Information on ray locations, phase velocities, polarization,
etc

3.2 Description of Parameters

The change a user might wish to make to LSC is commonly an adjustment of the dimensions on arrays. These are in files `params.inc` for the ray and quasilinear calculation; and in `emparams.inc` and `xparams.inc` for the x-ray camera. A way to make a parameter change while not touching the code as found in one of the archive areas, is to make your own copy of the parameter file you wish to change in the compile/load directory, and instruct the compiler where to find all the other `.inc` files with the `-I` switch.

Relevant text from the manual pages is reproduced below.

f77 (Sun Fortran)

`-Ipath`

Insert `path` at the beginning of the list of directories in which to search for `#include` files. Example: `f77 -I/usr/applib growth.f` searches for `#include` files in `/usr/applib`.

This path and directory list is for `#include` files with relative path names, not absolute path names. This option applies only to source files with a `.F` suffix. Note that this does not affect the FORTRAN `INCLUDE` statement, only the C preprocessor `#include` directive.

cf77 (Cray Fortran; not to be confused with `cft77`)

`-I inkdir`

Specifies a directory to be searched for files named in `INCLUDE` lines. Up to ten directories can be listed; each must have its own `-I` option as in `cf77 -I dir1 -I dir2 file.f`.

Directories can be specified by `-I` as full paths, or as paths relative to the working directory. If the `INCLUDE` line specifies an absolute name (that is, beginning with `/`), that name is used, and no other directory is searched. If the `INCLUDE` uses a relative name (not beginning with `/`), the compiler first searches for the file in the directory of the source file. If this directory contains no file of that name, the compiler then searches the directories named by the `-I` option, as shown on the command line from left to right.

This option activates the `cft77 -P` compiler option and equivalent options for other system phases. `INCLUDE` is described on man page `f77syn(7)`. This is not to be confused with the `cft77 -I` option.

When used with the GPP preprocessor, `cf77 -I` applies to `#include` files with relative names. Thus, `#include` files whose names are enclosed in `"` are searched first in the directory of the input file, then in directories specified by `-I` option, and finally in the standard directories. For `#include` files whose names are enclosed in `< >`, the directory of the input file is not searched.

Note that LSC uses the `#include` form; but that TSC uses the `INCLUDE` form.

3.2.1 Ray and Quasilinear Parameters

We begin with a description of the major parameters of interest in `params.inc`, along with a typical value.

`NPSIDIM = 100` sets the maximum number of flux (`psi`) surfaces used inside LSC to keep track of ray motion.

`NZONDIM = 2000` is the maximum number of zones that a ray can cross before the program gives up and moves on to the next ray.

`NTORDIM=40` is the maximum number of toroidal components of $n_{||}$, which is multiplied by `NPOLDIM=3` the maximum number of poloidal components to get `NRAYDIM=NTORDIM*NPOLDIM` which is the maximum number of rays.

NGRPDIM = 3 is the maximum number of couplers if a Brambilla spectrum is calculated or the maximum number of groups if the spectrum is given by Gaussian shapes characterized by a center, width, relative power.

NVELDIM = 201 is the number of velocity bins (in v/c).

The following “P” parameters save room for the plasma data to be passed from TSC or TRANSP. PIMP=3 is the maximum number of ion species. PPSI= 101 is the maximum number of flux surfaces to be used inside TSC and for LSC to have to interpolate between for the NPSIDIM internal surfaces. PNZ= 125, and PNZ= 139 and the maximum number of points in the flux grid for the x -direction (r -direction) and the z -direction.

The following parameters would not normally be changed by a user of the program. NITRDIM = 2 is the number of copies of the electron distribution function kept; it is used in iteration. NPLTDIM = 800; ZXDIM = 2 * PNZ * PNZ; NZRDIM = 3 * NRAYDIM * NZONDIM; NVPDIM = 2 * NVELDIM * NPSIDIM; NPZDIM = 2 * NPSIDIM * NZONDIM; NWKVDIM = NVELDIM+NPSIDIM+NZONDIM; PWORDS = 10; INUNIT = 3; OUTUNIT = 4; NWINDIM = 4; FALSE = 0; TRUE = 1; UNIFGRID = 1; STRGRID = 2; EXPGRID = 3.

3.2.2 X-ray Camera Parameters

The parameters of the x-ray camera are relatively few.

NENDIM = 200 is the maximum number of energy bins for the photons. NMUDIM = 21 is the maximum number of bins for $\mu \equiv \cos \theta$, the cosine of the angle between the magnetic field and the emission direction.

NPIXDIM = 50 is the maximum number of pixels in horizontally *and* vertically for forming the x-ray image.

NRDIM = 90 and NZDIM = 90 are the maximum number of cells the plasma cross section is divided into.

MAXPOINTS = 1000 is the number of emission sections along an emitting chord.

The remaining parameters should not be adjusted by a user. They are: ACCEPTABLE_ERROR = 1.e-3, SPACEDIM = 3, NCHORDIM = 6, NRZMXDIM=90.

3.3 Description of Input

User-controllable input to LSC is through the FORTRAN NAMELIST statement, data files, or answers to questions asked by a simple driver program

if LSC is being used in stand-alone mode. This section is concerned with NAMELIST input only, which can come in two separate files. It is important to realize that LSC declares all variables, and does not follow the FORTRAN standard naming convention with respect to reals (floating point) and integers. In LSC reals may begin with one of i,j,k,l,m,n; and integers do not necessarily begin with one of those letters.

The first file, **input.lhh**, which must always be present, contains information about rays, velocity grids, computing and plotting options. The second file, **input.xry**, which does not have to be present, contains information about the x-ray camera. The x-ray camera is not executed unless a switch (DoXcam) is thrown to unity.

3.3.1 File “input.lhh”

The following is an alphabetical list, with parentheses () indicating an array:

Do1Rpr	DiffuJrf	iRememFe	nsmoo
DoAdFs	PrfSpred	idiag()	nsmw
DoBern	HstpLH	incLabls	nstep
DoBram	PIFlg()	nFlat	ntors
DoBret	PrFlg()	nGrps	nv
DoComm	ScatKdeg	nRampOld	nzones
DoDeBg	StpRange	nRampUp	phaseDeg()
DoEVsm	TurnNegs	nfreq	powers()
DoFish	WeghtItr	npaNLB	rstNLB
DoGrzi	centers()	npamax	thet0
DoHarv	couplers()	nparmin	v1minus
DoPaus	degNLB	npolmax	v1plus
DoRFDa	epsvgr	npolmin	v2minus
DoRFHe	fghz	npols	v2plus
DoScrn	frv1minus	npsi	vgrtype
DoTRAN	frv1plus	nrays	widths()
DoXcam	frv2minus	nslices	zstNLB
	frv2plus		

The TRANSP code supplies LSC with these variables through common blocks according to variables in the TRANSP namelist. The TRANSP namelist names are different from the LSC namelist names, but in a generally systematic way: Integers begin with **N**; and all variables end in **LH**.

More Important Inputs — Definition

Do1Rpr	This INTEGER flag modifies the way LSC recalculates rays when it is being called from TSC or TRANSP. If Do1Rpr is equal to 1, then LSC recomputes only 1 ray per call from TSC/TRANSP after the initial computation of all rays; this smooths the dynamic behavior. If equal to 0, then LSC recomputes all rays when called by the driving program.
DoBram	INTEGER: If DoBram is equal to 1 (the default), then LSC will perform the Brambilla calculation, using nGrps as the number of couplers, and the array powers() as the relative distribution of power between the couplers. If DoBram is equal to 0, LSC uses Gaussian models for the launched spectrum.
DoTRAN	This INTEGER flag (which has a default of 0) indicates to LSC how it is being called; if LSC is being called by TRANSP, then this flag should be set to 1.
DoXcam	INTEGER flag: If DoXcam is set to one, then LSC will give pictures and plots just like the 2-Dimensional X-Ray camera (see the section on LSC Outputs, below, for more details); if set to zero (the default), then LSC will not perform the X-Ray calculations.
HstpLH	The length of the ray integration step, in meters. The default is 0.005 (meters).
PIFlg()	This INTEGER array is made up of the flags that inform LSC as to which (if any) of the possible plots should be produced as output; if a particular plot is desired, then the corresponding PIFlg element should be set to one (1). The default settings are all zeroes (no plots). For a full description of the possible output plots, see the section below on LSC Outputs.

PrFlg()	This INTEGER array is similar to PIFlg, above, but holds the flags governing LSC “screen” output. LSC has three output channels which have their destinations by the driving program, and conceivably any or all of these channels could be linked with the screen. Alternatively, these channels could be linked to additional output files. For a more complete description of the channels and the possible outputs, see the section below on LSC Outputs. The default setting of this array is all zeroes.
ScatKdeg	The angle (in degrees) by which the perpendicular component of the wave vector, k_{\perp} , may be rotated when the ray “bounces” (i.e. as in off of the wall). If ScatKdeg is 0.00 (the default), then there is no scattering – the ray is simply reflected specularly.
WeghtItr	The proportionality of the weighting of the old and new electron distribution functions during each ramp-up iteration; the default is 0.20 – 20 percent new f_e and 80 percent old f_e .
nGrps	Is an INTEGER giving the number of groups (or couplers) used in calculating the launch spectrum; the default is 3 groups.
centers()	Location of the peaks in the Maxwellian-model spectrum, in n_{\parallel} ; the three defaults are all 4.0 .
couplers()	The dimensions of certain couplers are built into LSC in DATA statements; hence, all the user must specify is the eight-character name(s). Presently, LSC will recognize ‘PBXMFAST’, ‘PBXMSLOW’, ‘TOKDEVAR’ and ‘TOR-SUPRA’. By default, the first three elements of couplers() are set to ‘PBXMFAST’, ‘PBXMSLOW’ and ‘TOKDEVAR’, respectively.
widths()	Widths of the model spectrum peaks, as Δn_{\parallel} ; the three defaults are all 1.0 .
phaseDeg()	The relative phases (in degrees) applied to the couplers; used in computing the Brambilla spectrum; the default values are 90.0, 180.0 and 135.0, respectively.

powers()	The relative powers in either the Maxwellian-model spectrum components (if DoBram is zero) or in the couplers (if DoBram is equal to unity); as an example, if powers(1) and powers(2) are both 1.0, LSC will assign equal powers to the first two components/couplers, whereas if powers(1) = 1.0 and powers(2) = 0.5, LSC would assign twice as much total power to the first component/coupler as the second. The default values are 1.0, 0.1 and 0.1 for elements 1, 2 and 3, respectively.
fghz	The launched LH wave frequency (in GHz). The default is 4.6 GHz.
idiag()	This INTEGER array specifies at which ramp-up iterations the results of f_e and D_{ql} calculations will be ouputted; that is, the first element of idiag tells LSC at which iteration to produce the first set of plots, the second element tells at which iteration to produce the second set, etc. idiag starts out with all IDIAGDIM elements set to zero; the user should (at the very least) set the first element of idiag to nRampUp – otherwise, LSC does not know “when” to produce certain outputs, and will not (regardless of the values set in PIFlg).
nFlat	The (INTEGER) number of ramp-up iterations with a flat (constant) power; the default is 5 iterations.
nRampUp	The (INTEGER) number of steps taken in ramping up the power for the iteration of f_e , D_{ql} and P_{rf} ; some of these steps may be with a constant power (controlled by nFlat, above). The default is 10 steps.
nparmax nparmin	These are floating point numbers giving the maximum and minimum values of $n_{ }$ over which to distribute the ntors number of toroidal values when using the Maxwellian-model spectrum (hence, these inputs are ignored when DoBram is set to 1). The defaults are 5.5 for nparmax and 2.5 for nparmin.

npolmax npolmin	Floating point numbers giving the maximum and minimum values of n_θ over which to distribute the npols number of poloidal values when using either the Brambilla or Maxwellian spectrum; their default values are 1.0 for npolmax and -1.0 for npolmin.
npols	Number of poloidal variations to each toroidal variation; in part, this is a way to increase the number of rays launched from the same Bramilla spectrum, but also effective for the model spectrum; the default value of this INTEGER is 1.
npsi	The (INTEGER) number of psi bins used in LSC (during the D_{ql} calculation); npsi can be either more or less than the npsii from the TSC grid. The default is 40 bins.
nrays	The (INTEGER) number of rays LSC will trace; its most natural value (also the default) is the product of ntors with npols, but if nrays is given as something different, then a sensible action is taken.
nsmoo	The number of velocity bins over which smoothing is to be done; the default for this INTEGER is 9 (bins).
nsmw	INTEGER giving the characteristic width (in number of velocity bins) of the smoothing; the default is 3.
nstep	The maximum number of integration steps (each having length HstpLH) through which LSC will track a ray; the default is 500 steps.
ntors	The INTEGER number of distinct toroidal components in the number of rays (nrays); the default setting for ntors is NTORDIM.
nv	INTEGER number of parallel velocity bins; the default is 199 bins.
nzones	The number of zone (psi surface) crossings through which LSC will trace a ray; even if not all of the ray power has been damped away, LSC will not trace a ray any further than nzones crossings. The default setting for nzones is NZONDIM.

vgrtype	INTEGER that specifies the type of velocity grid: if 1, LSC uses a uniform (linear) grid; if equal to 3, LSC uses an exponential grid (interval between velocity bins increases geometrically). The default value is 1; the exponential velocity grid is a non-recommended option, in light of the fact that a uniform grid seems to work fine, and is simpler to work with.
DiffuJrf	This is a heuristic diffusivity for rf-driven current in meter squared per second. It balances a collision rate, as shown in equation 2.39. Default is 0.
PrfSpred	This governs the heuristic spreading of rf power as given by α in equation 2.40. Dimensionless, between 0.0 and 1.0. Default is 0.

Less Important Inputs — Definition

DoAdFs	If DoAdFs (“Add Fast Fraction”) is equal to one, then the electron distribution function that LSC will work with is the sum of a normal Maxwellian function (with values at every point in the plasma determined by the plasma density, temperature, etc.) which characterizes most of the electrons and a non-Maxwellian function (read in from the input file feout.d which describes the distribution of a relatively small fraction of the total number of electrons. The default value for this INTEGER flag is zero.
DoBern	If this INTEGER flag is set to 1, then LSC will adjust the density, temperature, and magnitude of the magnetic field according to values given at the end of the TSC-generated input file. This option is intended for stand alone runs, when investigating the dependence of power deposition upon the density, etc. The default value is zero (no adjustments).

DoBret	This INTEGER flag governs the user option of starting a LH ray at some point in the plasma interior; if equal to zero (the default), then LSC will begin tracing the rays from the edge of the plasma (as if the waves were launched from waveguides); if DoBret is set to 1, LSC will start a single ray propagating from a specified point within the plasma, with a specified initial wavevector (as though the LH wave was started by an instability). For details on setting the initial ray conditions, see rstNLB, zstNLB, etc., below. (These parameters were suggested by Norton L. Bretz)
DoComm	This INTEGER flag can be used to suppress certain outputs. If DoComm is set to one, then all outputs will be controlled by the array of input flags PrFlg; if DoComm is set to zero (the default), then LSC will not output certain lists and tables, regardless of the values set in PrFlg. See the section on LSC Outputs, below, for more explanation.
DoDeBg	This INTEGER flag controls the screen output that is meant for testing and debugging LSC. When DoDeBg is set to unity, LSC will output extra ray information as each ray is being traced. On the other hand, if it is set to zero (the default), then there will be no special outputs.
DoEVsm	This INTEGER flag controls how LSC smooths out discontinuities in P_{rf} , J_{rf} , and f_e – if DoEVsm is set to 1, then LSC smooths the quasilinear diffusion coefficient, and then uses this continuous function to find the quantities above. The default value of DoEVsm is 0.

DoFish	If this INTEGER flag is set equal to 1, then LSC will perform some extra calculations pertaining to alpha particle diffusion (for Nathaniel Fis(c)h of PPPL). Basically, for each ray, at every point along the ray trajectories, LSC will be keeping track of the components of the LH wave electric field that are perpendicular to the magnetic field (two electric field components: one that is parallel to the plasma density gradient, and the other perpendicular to that gradient). This information will be appended to ray.dat , and possibly given as part of the plotted output; for details, see the section on LSC Outputs.
DoGrzi	This INTEGER flag (if set to 1), when the user has already requested plots of f_e and D_{qt} (by setting PIFlg flags), will direct LSC to produce detailed plots of these functions. These detailed plots were originally designed for analysis of electron diffusion in the plasma (by Gerardo Giruzzi). If DoGrzi is zero (the default), then even if f_e and D_{qt} information is requested, these special plots will not be produced.
DoHarv	If DoHarv is equal to 1, then LSC will produce an output file of ray information called rayCQL.dat ; this ray information has (in the past) has been used (by Bob Harvey of General Atomics) to calibrate LSC against a full Fokker-Planck treatment (CQL3D). If this INTEGER flag is zero (the default), then LSC will not produce this file.
DoPaus	If the INTEGER flag DoPAUS is set to 1, LSC modifies its “screen” output by pausing for a “Return” between screens. This flag has a default of 0 (no pausing – don’t blink!).
DoRFDa DoRFHe DoScrn	These three do-flags are defunct.
nfreq	After each ray has been traced for an nfreq number of steps, LSC takes stock of what has occurred. This INTEGER has a default value of 100 (steps).
nslices	This INTEGER is the number of slices on the Brambrilla spectrum; nslices has a default value of 301.

TurnNegs	If this INTEGER flag is set to one, then any “negatively-directed” Brambrilla spectrum elements will be turned around; literally, the corresponding n_{\parallel} components will be multiplied by a negative one, which will reverse the direction of the spectrum element. TurnNegs has a default value of zero.
StpRange	This REAL parameter is used in setting up a exponential velocity grid; specifically, the ratio of the interval between velocity bins at the edge of the grid to the interval between bins at zero velocity will be equal to StpRange. The default setting for StpRange is 4.0, which would give a velocity grid with some interval between bins around zero velocity, and an interval 4.0 times as wide at velocities near the maximum.
iRememFe	This INTEGER flag tells LSC whether or not to keep the electron distribution function between LSC calls; if set to one, then the function will be “remembered”; if zero (the default), then LSC will start from scratch each time.
incLabls	When incLabls is set to one, LSC will put extra labels on the axes of certain output plots; the default of this INTEGER flag is zero (no labels).
nRampOld	The INTEGER number of ramp-up iterations starting with the old distribution function; typically, nRampOld is equal to nflat, but can be either greater or smaller. The default is 6.
degNLB npaNLB rstNLB zstNLB	The initial position and orientation of a LH wave which LSC starts in the plasma interior: rstNLB and zstNLB give the initial position of the ray, in terms of a radial distance from the axis of toroidal symmetry (default of 3.0 meters) and a height above the midplane (default of 0.0 meters, respectively; npaNLB is the initial n_{\parallel} (default is 3.0); degNLB is the initial angle (in degrees) between the wave vector and the magnetic flux surface (the default value is 0.0: k_{\perp} is inside the flux surface).
lfast	This INTEGER flag is equal to 1 if a “fast” wave is being launched; the default value is zero (normal LH wave).

thet0	The angle of ray launch, in units of radians divided by $\frac{\pi}{2}$; thet0 equalling zero (the default) corresponds to the ray being launched from the midplane, while thet0 equalling 0.25 would correspond to the ray being launched from directly above the plasma.
epsvgr	These nine parameters are used in specifying a “stretched” velocity grid; the “stretched” velocity grid is an archaic LSC option which is no longer supported.
v1minus	
v2minus	
v1plus	
v2plus	
frv1minus	
frv1plus	
frv2minus	
frv2plus	

3.3.2 File “input.xry”

The following is an alphabetical list of the X-Ray camera parameters:

E_max	R_tangent	dlxray	npoints_max
E_min	Rpinhole	focal_length	nr_source
E_ph_min	Z_bound_max	iAbsXray	nz_source
FoilCode	Z_bound_min	nEbins	phi_pinhole
PusherMajor	Zpinhole	nMUbins	pinhole.size
PusherMinor	dE_ph	n_pixel_x	screen_d
R_bound_max	dFoilTCM	n_pixel_y	z_tangent
R_bound_min			

Definitions of Inputs

E_max	To render the calculations tractible, only the contributions from electrons in the plasma with kinetic energies within a certain range will be considered; E_min and E_max (REAL numbers) specify (in keV) the minimum and maximum electron energies of this range. If the user does not specify these energy bounds, LSC will set E_min to 0.01 keV and E_max to 2.0 keV.
E_min	

E_ph_min	E_ph_min specifies the minimum energy that a photon must have in order for LSC to add its contribution into the line integral. The default value of E_ph_min is 1×10^{-6} keV.
FoilCode	FoilCode is a two-CHARACTER parameter specifying the type of metal foil that the X-Rays must pass through in going from the pinhole to the film; FoilCode can be one of the following: “CU”, “AG”, “TA”, “MO”, or “00” for no foil at all. By default, FoilCode is set to “AG” – a silver foil.
PusherMajor PusherMinor	These are the radial positions of the major and minor “pusher” field coils, as measured from the axis of toroidal symmetry. Their default values are 1.152 meters and 0.190 meters, respectively.
R_bound_max R_bound_min	These inputs specify the radial extent of the plasma; that is, the plasma will extend no further in than R_bound_min, and no further out than R_bound_max. The default values for R_bound_min and R_bound_max are 1.10 meters and 2.00 meters, respectively.
Z_bound_max Z_bound_min	Similar to the R_bound inputs above, Z_bound_min and Z_bound_max specify the furthest extents of the plasma along z. If the user does not specify them, Z_bound_max and Z_bound_min are set to 1.00 meters and -1.00 meters. Basically, these six inputs specify the boundaries at which LSC would stop calculating the line integrals, since there is no plasma beyond the R or Z bounds or next to the “pusher” coils.
R_tangent z_tangent	The pinhole aperture of the X-Ray camera and the point at the center of the image screen define a line in space; clearly, this line will be tangent to some circle about the axis of toroidal symmetry; R_tangent is the radius of this circle (with a default value of 1.524 meters), while z_tangent is the z-position (height) of this circle (with a default value of 0.0 meters).

Rpinhole Zpinhole phi_pinhole	These three inputs completely specify the coordinate location of the pinhole within the torus. The default values for these three are 2.661 meters, 0.0 meters, and 0.0 degrees, respectively.
dE_ph	The interval between photon energy bins, in keV. The default is 0.01 keV.
dFoilThickCM	The thickness of the metal foil, in units of centimeters. The default value is 0.00 centimeters.
dlxray	The default value of this REAL parameter is 0.005 meters.
focal_length	This is the focal length in meters of the X-Ray camera – literally, this is the distance from the pinhole to the screen upon which the image is formed. The default value is 0.4905 meters.
iAbsXrays	This INTEGER flag tells LSC whether or not to take account of the absorption of X-rays by the foil when calculating the X-ray camera image; the default value of iAbsXrays is 1 (absorption).
nEbins	This INTEGER specifies the number of X-Ray energy bins. If left unspecified, the default value is NENDIM.
nMUbins	This INTEGER specifies the number of mu bins used in the calculation of the line integral through the plasma to the pinhole. By default, nMUbins is set to NMUDIM.
n_pixel_x n_pixel_y	These REAL input parameters are (respectively) the extent of the X-Ray camera image, in the x- and y-directions, in number of pixels; that is, the camera image will be n_pixel_x pixels "long" along x, and n_pixels_y "long" along y. Their default values are both NPIXDIM.
npoints_max	By default, npoints_max is set to MAXPOINTS.
nr_source nz_source	Are the number of sections the plasma cross-section is divided into, radially and height-wise. The default settings for nr_source and nz_source are NRDIM and NZDIM, respectively.

pinhole_size	The physical size (diameter in meters) of the pinhole; the default is 0.00625 meters.
screen_d	The "diameter" in meters of the screen upon which the X-Ray camera image is formed; screen_d is the diameter of the largest circle that can be inscribed in the screen. By default, screen_d is set to 0.215 meters.

3.3.3 NAMELIST Files

Here we give typical input NAMELIST files.

The (required) file input.lhh:

```

$inpval
nv = 199, weghtitr = .2, vgrtype = 3, nfreq = 100, nstep = 10000,
npsi = 40, nzones = 500, nsmoo = 7, nsmw = 3, fghz = 4.6,
nrays = 3, ntors = 3, npols = 1, HstpLH = .005,
nparmin = 2.0, nparmax = 4.0, npolmin = 2.0, npolmax = 2.0, thet0=0.0,
nGrps = 2, couplers(1)='PBXMSLOW', couplers(2)='TOKDEVAR',
centers(1) = 3.0, centers(2) = 5.0, widths(1) = 2.0, widths(2) = 2.0,
powers(1) = 1.0, powers(2) = 1.0,
phaseDeg(1) = 0., phaseDeg(2) = 0., nslices=601,
DoPaus = 0, DoDeBg = 0, DoGrzi = 0, DoFish = 1,
DoBram = 0, DoBern = 0, DoEVsm = 0, DoComm = 1,
DoBret = 0, DoXcam = 1, DoHarv = 0, Do1rPr = 0,
nrampup = 25, nflat = 5, nrampold = 10, iRememFe = 0,
scatKdeg=0.0, incLabls = 1,
rstNLB = 1.70, zstNLB=0.2, npanLB=3.0, degNLB = 0.0,
PlFlg(1) = 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,
PlFlg(11) = 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
PlFlg(21) = 0, 0, 0, 0, 0, 0,
PrFlg(1) = 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
PrFlg(11) = 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
PrFlg(21) = 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
idiag(1) = 25, idiag(2) = 0,
$end

```

The (optional) file input.xry:

```

$inpxry
  nr_source = 89, nz_source = 89,
  n_pixel_x = 39, n_pixel_y = 39, dlxray=0.005,
  nEbins=30, E_min= 0.02, E_max = 0.32,
  iAbsXrays = 0,
  nmubins = 20,
$end

```

3.3.4 Output Files

LSC will normally produce two files as it runs: **Wr2TSC.out** and **ray.dat**.

The first of these files, **Wr2TSC.out**, contains information which LSC would be communicating to TSC if they were being run in tandem: the total RF power deposited in the plasma (as a function of ψ , where ψ is the magnetic flux index) and the RF-driven current density (also reported as a function of ψ).

The second file, **ray.dat**, contains for each ray the ray location, velocity, rate of power loss and polarization at every point in its trajectory.

Additionally, if LSC has been directed to produce output plots (by the setting one or more PIFlg elements to unity), then LSC will produce a Tektronix 4014 plot file, under the name specified in the users' PLOT environment variable. For example, if the PLOT variable were set to '-f plt.plt', then LSC would build a Tek file called **plt.plt** which would contain all of the output plots. All of the plots themselves and their relation to the array PIFlg are detailed below.

Further, LSC has three output channels in addition to the two channels used to dump information to **Wr2TSC.out** and **ray.dat**; the channel numbers themselves are contained in the COMMON memory area "tscunits" in the variables nLSCcomm, nLSCcom2, and ns30c1. The important point to note is that these output channels must be set up in the driving program before LSC is called, and that each channel may be linked with either a file or the screen. Most of the data which could be outputted by LSC along these channels is "screen" output controlled by the array PrFlg, detailed below.

Finally, LSC will produce the output file **rayCQL.dat** if the input flag DoHarv is set to one. **rayCQL.dat** contains, in a highly formatted form,

a rather extensive amount of ray information for each ray at every point as it travels around the torus, including the ray positions, power remaining, length of the optical path to that point, ray polarizations, etc.

3.3.5 Output Plots and Tables

Regardless of the actual flag values set in the arrays PIFlg and PrFlg, LSC will put a listing of the LSC input parameters (the values set in **input.lhh**) into the Tektronix plot file and report any warnings or error messages to either the screen or the output channel nLSCcomm (see the previous section for a description of LSC output channels).

Many of LSC's output plots are grouped together to form filled pages of related information; if any one of the plots on a page are selected as part of the output, the entire page will be produced.

What follows is the plotting output flags, and the plots which LSC would produce if they are set to a value of unity:

PIFlg(1) For each ray, a page of six plots of trajectory information (See Fig. 3.1 for an example): the ray path around the toroidal plasma; the ray position projected onto poloidal cross-section; n_{\parallel} and n_{\perp} versus $\sqrt{\psi}$ (where ψ is the distance from the plasma center); and, depending upon whether or not LSC is simulating ray scattering, two plots that will display either information on the polarization of the ray or the angles of scattering. Regardless of the number of rays, LSC will also produce a single page of contour plots which will display the magnetic flux surfaces, regions in the plasma of n_{\parallel} enhancement, and plasma regions of ray (in)accessibility, all on poloidal cross-sections of the toroidal plasma.

PIFlg(2) During a power ramp-up iteration in which diagnostics are to be performed, if either of these flags are set, then LSC will produce a single page of two contour plots. One is of the quasilinear diffusion coefficient D_{ql} plotted as a function of $\sqrt{\psi}$ and $v_{\parallel} \times \sqrt{\psi}$, where ψ is the distance from the plasma center; the other contour plot (See Fig. 3.2 for an example) is of the electron distribution function f_e as a function of $\sqrt{\psi}$ and v_{\parallel} .

PIFlg(4)	These flags will prompt LSC to produce one-page contour plots of either DColl, nuColl, PRay or P_{ql} as functions of ψ_{Ary} and v_{\parallel} ψ_{Ary} , respectively. Recall that DColl is the collisional diffusion coefficient, nuColl is the collisional drag coefficient, PRay is the total ray power, and P_{ql} is the plasma heating power due to quasilinear damping of the rays. Note that ψ_{Ary} is the coordinate used by TSC to give the distance from the plasma center.
PIFlg(5)	
PIFlg(19)	
PIFlg(20)	
PIFlg(6)	A one-page plot showing the quasilinear smoothing function as a function of velocity.
PIFlg(7)	Setting either of these flags will prompt the production of a graph of the ray trajectories plotted on a v_{\parallel} versus ψ grid; PIFlg(7) controls the output of the continuous trajectories, while PIFlg(8) controls the “discrete” (bin to bin) ray trajectory plot.
PIFlg(8)	
PIFlg(12)	A one-page plot of the effective ion charge Z as a function of ψ , the distance from the plasma center.
PIFlg(10)	Setting any one of these flags will prompt LSC to produce a single page with six plots: the electron density N_e , the electron temperature T_e , the total current I_{tsc} , the static electric field E_{dc} , $d \ln J / d \ln E$ and the minimum and maximum ray phase velocity components v_{\parallel} , all as functions of ψ , the distance from the plasma center.
PIFlg(11)	
PIFlg(24)	
PIFlg(21)	If any of these flags are set, then at the end of a run LSC will provide a single page with six plots: the power absorbed by quasilinear damping P_{ql} , the RF heating power P_{rf} and the RF driven current J_{rf} as functions of ψ , and plots of these three quantities integrated over ψ as functions of ψ . Also, if either PIFlg(23) or PIFlg(24) are set, then these same plots will be produced at every ramp-up iteration where the user has requested diagnostics; then, the plots will be of the power deposited and current driven so far.
PIFlg(23)	
PIFlg(24)	

- PIFlg(21) Setting any of these flags will give a one-page set of three
PIFlg(22) plots showing the power spectrum of the launched LH waves:
PIFlg(23) the power is plotted (unsmoothed) versus n_{tor} and also ver-
PIFlg(24) sus wave velocity; a smoothed power spectrum is also plotted
against wave velocity alone.
- PIFlg(23) For each ramp-up iteration at which diagnostic information
is requested, LSC will plot the LH power of each ray versus
zone number; thus, the ψ zones at which quasilinear damping
occurred would be indicated by drops in the ray power.
- PIFlg(25) Setting this flag will prompt LSC to output the actual ψ
versus velocity grid used in calculations; in this fashion, a
user could doublecheck that the grid used has fine enough of
a resolution in the regions of interest.
- PIFlg(26) If this flag is set, LSC will produce a single page with three
plots: the pitch angle in the magnetic field, the electron den-
sity N_e , and the deposited RF power P_{rf} ; these are all given
in the midplane of the torus, and plotted versus R_{Major} , the
radial distance from the axis of toroidal symmetry.

Some plots, though, will only be produced if the input flag PrFlg(27) is
also set to unity:

- PIFlg(13) A plot of the average electron v_{\perp}^2 as a function of ψ , where
 ψ is the distance from the plasma center.
- PIFlg(14) The electron thermal speed v_{th} as a function of ψ .
- PIFlg(15) One slice of the final electron distribution function, plotted
as a function of ψ (specifically, $f_e(v_{\parallel} = 0, \psi)$).
- PIFlg(17) The initial electron distribution function (nu0psi), plotted as
a function of ψ .

When both DoFish and PIFlg(3) are set to one, the output will include
two additional plots: another plot of the quasilinear diffusion coefficient D_{qt}
(incrementally calculated using the component of the LH wave electric field
that is parallel to the magnetic field – see equation 2.23 on page 9) and a plot
of a similar function, D_{rt} (calculated in a fashion identical to D_{qt} , except
that the electric field component used is the one that is perpendicular to

both **B** and the plasma density gradient).

Also, if the input flag DoXcam is set to one, then LSC will produce four more pages related to the X-Ray camera option: the first page listing the relevant input parameters (set in **input.xry**); the second page having one plot that is the actual X-Ray camera image, and another plot showing the regions of Bremsstrahlung emission on a poloidal cross-section of the plasma (see Fig. 3.3 for an example); the last two pages are details of the X-ray and emission intensity plots, showing horizontal and vertical slices. For every plot that LSC produces as part of the output (including any related to the X-Ray camera), a short (one-line) description of the plot will be sent to the output channel ns30c1.

Regardless of the PrFlg values, nLSCcomm receives certain LSC warnings (i.e. “Maxwellian power=0; stop early” when all of a ray’s power is damped away); also, if DoDeBg is set to one, then LSC will send extra information to nLSCcomm as it traces the rays.

There are tables of information that LSC could send to nLSCcomm, depending upon the values of particular PrFlg flags; these flags, along with the tables that they would prompt LSC to produce, are:

- PrFlg(1) LSC will list out ray information (number of zones traveled, distance from the plasma center, components of the wavevector, etc) as each ray is being traced.
- PrFlg(28) Listing of the fractions of “fast” electrons in each ψ zone.

Additionally, there are two more summary bits of information that LSC could send to the channel nLSCcomm, but only when both the input flag DoComm and the appropriate PrFlg flags are set to one:

- PrFlg(21) The total power damped into the plasma
- PrFlg(22) The total RF heating power transmitted to the plasma.

Like the output channel nLSCcomm, the channel nLSCcom2 will always receive some LSC warnings (i.e. “PLOT ARRAY IS FILLED”), ray trajectory information, and (if there are any plots requested on a ψ versus v_{\parallel} grid) the extent of plotted data in ψ and v_{\parallel} .

LSC will send various tables of information to channel nLSCcom2 if the

appropriate PrFlg flags are set to unity:

- PrFlg(1) A table of plasma parameters (magnetic field components and the ratio of electron plasma frequency to electron cyclotron frequency) at every point in the plasma, and a table of ray information (ray position, wavevector at that position, the distance traveled by the ray to that point, and the magnetic field and electron density at that position).
- PrFlg(6) Lists the quasilinear smoothing function versus velocity.
- PrFlg(9) A table of ray information (rates of power loss, etc.) versus ray number and zone number.
- PrFlg(23) Lists all the v_{\parallel} grid points and the inverses of the intervals between these grid points.
- PrFlg(24) Lists all the ψ_{Ary} grid points; recall that ψ_{Ary} is the distance from the plasma center that is used by TSC.
- PrFlg(25) Lists out two tables: the first includes for each ray its ψ bin for each zone – essentially, its distance from the center of the plasma versus the distance traveled; the second table represents information that is no longer used by LSC.
- PrFlg(26) Lists plasma parameters (including electron temperature T_e and electron density N_e) versus ψ , the distance from the plasma center.
- PrFlg(27) Output the NAMELIST “fecnl”, which contains a series of normalization factors used in the f_e calculation.
- PrFlg(29) A set of ray powers versus toroidal wavevector components

There are more tables that LSC could send to nLSCcom2, but only when both the input flag DoComm and the appropriate PrFlg elements are set to unity:

- PrFlg(1) List the roots of the dispersion equation; if no roots are found, then the ray will be unable to propagate.
- PrFlg(19) List, for each ψ zone, the velocity zones at which the rays were damped, and the power damped away.

PrFlg(20) For each ψ zone, list by velocity bin the power from quasi-linear heating.

The output flags P1Flg(9), P1Flg(16), P1Flg(18), PrFlg(7), PrFlg(8), PrFlg(10), PrFlg(12) and PrFlg(18) are defunct; their values will be ignored by LSC.

3.4 Makefiles

The UNIX `make` utility is the best way to create an executable file from the source code. Instructions are given in a `Makefile` on the location of source, include files, libraries; and where the executable is to be placed.

Because details can change, it is good to find a current working version of a `Makefile`, and modify it as required. As examples, we give below `Makefiles` that work on the Sun and the Cray.

The Sun command `.KEEP_STATE:` causes all dependencies on include files to be figured out. Unfortunately, the Cray can keep track of dependencies on include files only when they are stated explicitly.

3.4.1 Sun Makefile

```
SHELL=/bin/csh
PTH = /tmp/myjunk/
INC = -I/u/home/lsc/inc/

SOURCES.F= lscdrive.F AcDc.F Rayini.F Rayio.F Raystore.F \
Raytrace.F Wr2TSC.F brambJES.F contrsg.F cycle.F fe.F fits.F \
grap.F grapgrd.F gridgen.F grids.F jrf.F matr.F pbxio.F \
plasejv.F power.F ql.F fedqlout.F ezsg.F plotsg.F XrayCam2.F

OPTIONS = -cg89 -dalign -libmil -r8 -i4

OPTIONS_G = $(OPTIONS) -g

tst = $(PTH)lsc

OBJECTS = $(SOURCES.F:%.F=$(PTH)%.o)

.KEEP_STATE:
```

```

$(tst): $(OBJECTS)
-rm -f $(tst)
date >> ld_err
nice f77 -v $(OPTIONS_G) -o $(tst) $(OBJECTS) \
        -L/u/valeo/bin/ -lsplibd >>$ ld_err

$(PTH)%.o: %.F
-rm -f $(*)_err
nice f77 $(INC) $(OPTIONS_G) -c -o $$ $< >& $(*)_err
-date >> $(*)_err
-grep -c Error $(*)_err

```

3.4.2 Cray Makefile

The need to give all dependencies explicitly causes the Makefile to be cumbersome and hard to maintain. We do not give the entire Makefile.

```

# Makefile for making LSC stand-alone or T_LSC
SHELL=/bin/csh
#
CFOPTS = -G -Wf"-a static"
LSCDIR = ./
TSCDIR = ./
LSCFLAGS = $(CFOPTS)
TSCFLAGS = $(CFOPTS)
COMPLSC = cf77 -c $(LSCFLAGS) -- $(LSCDIR)$(@F:.o=.F)
COMPTSC= cf77 -c $(TSCFLAGS) -- $(TSCDIR)$(@F:.o=.f)

LSCDRIV = lscdrive.o
LSCOBJS = \
AcDc.o Rayini.o Rayio.o Raystore.o Raytrace.o Wr2TSC.o \
brambJES.o cycle.o fe.o fits.o grap.o grapgrd.o \
gridgen.o grids.o jrf.o matr.o pbxio.o plasejv.o power.o ql.o \
ez80.o pl80.o s80.o contr80.o \
XrayCam2.o
TSCOBJS = unx9.08a.o unx9.08m.o unxmiss.o

```

SEGLDRLSC = \$(TV80) \$(NCARGKS)
SEGLDRTSC = \$(TV80) \$(NCARGKS) \$(LIBMATH) \$(NAG14)

T_LSC: \$(TSCOBJS) \$(LSCOBJS)
segldr -o \$@ \$(SEGLDRTSC) \$(LSCOBJS) \$(TSCOBJS)

LSC: \$(LSCDRIV) \$(LSCOBJS)
segldr -o \$@ \$(SEGLDRLSC) \$(LSCDRIV) \$(LSCOBJS)

unx9.08m.o: \$(TSCDIR)unx9.08m.f
\$(COMPTSC)
unx9.08a.o: \$(TSCDIR)unx9.08a.f
\$(COMPTSC)
unxmiss.o: \$(TSCDIR)unxmiss.f
\$(COMPTSC)

lscdrive.o : \$(LSCDIR)lscdrive.F
\$(COMPLSC)
AcDc.o : \$(LSCDIR)AcDc.F
\$(COMPLSC)
Rayini.o : \$(LSCDIR)Rayini.F
\$(COMPLSC)
Rayio.o : \$(LSCDIR)Rayio.F
\$(COMPLSC)
Raystore.o : \$(LSCDIR)Raystore.F
\$(COMPLSC)
Raytrace.o : \$(LSCDIR)Raytrace.F
\$(COMPLSC)

et cetera, et cetera.

Bibliography

- [1] D. W. Ignat, E. J. Valeo, and S. C. Jardin, “Dynamic Modeling of Lower Hybrid Current Drive,” Princeton Plasma Physics Laboratory Report PPPL-2929 (September 1993), Nucl. Fusion **34** 837-852 (1994).
- [2] Charles F. F. Karney and Nathaniel J. Fisch, “Current in wave-driven plasmas,” Phys. Fluids **29** 180 (1986).
- [3] S. C. Jardin, N. Pomphrey, and J. DeLucia, “Dynamic modeling of transport and positional control of tokamaks,” J. Comput. Phys. **66** 481 (1986).
- [4] R. Hawryluk, in *Physics of Plasmas Close to Thermonuclear Conditions*, (Varena, 1979), Commission of the European Communities, Brussels, **I** 61 (1979).
- [5] R. J. Goldston, D. C. McCune, H. H. Towner, S. L. Davis, and R. J. Hawryluk, *et al.*, J. Comput. Phys. **43** 61 (1981).
- [6] S. M. Kaye, R. J. Goldston, M. Bell, K. Bol, M. Bitter, R. Fonck, B. Grek, R. J. Hawryluk, D. Johnson, R. Kaita, H. Kugel, D. Mansfield, D. McCune, K. McGuire, D. Mueller, M. Okabayashi, D. Owens, G. Schmidt, and P. Thomas, “Thermal energy confinement scaling in PDX limiter discharges,” Nucl. Fusion **24** 1303 (1984).
- [7] T. H. Stix, *Waves in Plasmas*, (American Institute of Physics, New York, 1992).
- [8] William Daughton, “TSC Users Manual,” internal PPPL working document dated August 1993, available in PPPL Sun Network under directory `~pomphrey/Tsc/tsc.tex` .

- [9] V. Fuchs, I. P. Shkarofsky, R. A. Cairns, and P. T. Bonoli, “Simulations of lower hybrid current drive and ohmic transformer recharge,” *Nucl. Fusion* **29** 1479 (1989).
- [10] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling, *Numerical Recipes; The Art of Scientific Computing – FORTRAN Version*, (Cambridge University Press, New York, 1989).
- [11] H. W. Koch and J. W. Motz, “Bremsstrahlung Cross-Section Formulas and Related Data,” *Rev. Mod. Phys.* **31** 920 (1959).
- [12] W. H. McMaster, N. Kerr Del Grande, J. H. Mallett, and J. H. Hubbell, “Compilation of X-ray Cross Sections,” University of California, Livermore, California, Report UCRL 50174, Sec. II rev. 1, May 1969.