JOINT EUROPEAN TORUS JET

JET-R(87)08

K. Behringer

Description of the Impurity Transport Code "STRAHL"

Description of the Impurity Transport Code "STRAHL"

K. Behringer

JET Joint Undertaking, Culham Science Centre, OX14 3DB, Abingdon, UK

Description of the Impurity Transport Code "STRAHL"

K Behringer

JET Joint Undertaking, Abingdon, Oxon OX14 3EA, UK

1. INTRODUCTION

"STRAHL" is an interactive, stand-alone impurity transport code, which has been used since 1980 on ASDEX and later on JET for the interpretation of spectroscopic measurements. It calculates the impurity ionisation balance on the basis of given plasma parameters and empirical transport models, using atomic physics data sets especially compiled for that purpose. A simple version of neoclassical transport is included but, failing to explain the experimental results, it was only used for occasional testing. Specific line radiation for diagnostic purposes and total radiated power are computed. Rise and decay times for impurity injection can be simulated and particle losses and particle confinement with and without sources are calculated. A change in ionisation balance due to charge-exchange (CX) recombination is implemented in a rudimentary fashion.

STRAHL was originally designed for a PDP11/70, a relatively small computer with limited program size but with overlay capability. Because of the limited computer memory, several versions existed of the same code, performing only slightly different tasks, ie. NODIFF for ground state distributions, NEDIFF for emission shells and STRAHL for total radiation losses. Due to the availability of virtual memories, such restrictions no longer apply and these programs have now been combined into one. For specific purposes, modifications of the code were used, eg. to simulate the behaviour of neon ions injected as a trace in a deuterium pellet, or to calculate the transport of the bulk plasma itself in an iteration scheme.

In general, being a working code for spectroscopic measurements, it was often amended to cope with different problems and applications. The present paper is intended to explain the basic ideas and formulas behind this code in order to allow potential users to design their own special version.

The units used in the code are cm, s and eV. Thus, velocities are in cm/s and diffusion coefficients in cm^2/s. Excitation and ionisation energies must be entered in eV. Ionisation stages are labelled '1' for the atoms, '2' for the singly ionised species, etc.

2. ELEMENTS OF THE CODE

Except for the main program STRAHL, the code consists of the following sub-routines:

PLAD calculation of the plasma parameters from fit coefficients and calculation of transport functions.

SAL calculation of ionisation and recombination rate coefficients, charge exchange (CX) effects and the attenuation of neutrals entering at the plasma edge.

DATO output of results into a file for later printing.

CRTØ11 plot output on the terminal screen.

INTP interpolation routine doubling the number of grid points for plotting.

RDPAR input of parameters from the terminal keyboard separated by commas (no inverted commas for character variables).

The input data and transport coefficients used in STRAHL, controlled by a particular data file, as well as several other useful parameters can be displayed by the utility program "NEPLO".

3. SOLUTION OF THE DIFFERENTIAL EQUATIONS

The code calculates, in cylindrical geometry, the solution of the following set of partial differential equations for the impurity ions:

$$\frac{dn_Z}{dt} = - \text{div}\ \Gamma_Z - n_Z\ V_Z + Q_Z \qquad (1)$$

In Eq.1, n_Z and Γ_Z mean the density and flux of the ionisation stage Z, respectively, V_Z is the loss by ionisation, recombination and parallel losses, and Q_Z is the total gain from other ionisation stages, which is assumed to be known when n_Z is computed. The flux is split into a diffusive part with a diffusion coefficient D and a convective part with a drift velocity v_D:

$$\frac{dn_Z}{dt} = \frac{1}{r}\frac{d}{dr}\ r\ \left(D_Z\ \frac{dn_Z}{dr} - n_Z\ v_{D,Z}\right) - n_Z\ V_Z + Q_Z \qquad (2)$$

As a rule, more mesh points are required at the plasma periphery, where gradients are steep, than in the plasma interior. Therefore, the coordinate $\rho = r^k$ is introduced, where k is a positive real number greater than or equal to one. After transformation to ρ, Eq.2 writes

$$\frac{dn_Z}{dt} = D_Z\ \frac{k^2\rho^2}{r^2}\ \frac{d^2n_Z}{d\rho^2} + \left(\frac{k^2\rho^2}{r^2}\ \frac{dD_Z}{d\rho} + \frac{k^2\rho}{r^2}\ D_Z - \frac{k\rho v_{D,Z}}{r}\right).\ \frac{dn_Z}{d\rho}$$

$$- \left(\frac{k\rho}{r}\ \frac{dv_{D,Z}}{d\rho} + \frac{v_{D,Z}}{r} + V_Z\right)\ n_Z + Q_Z \qquad (3)$$

A time-centred (Crank-Nicholson) difference scheme is used for the numerical solution. In the following, $n_{Z,j,i}$ is the density for the actual time j and the considered mesh point i.

3.1 Relations for the Inner Meshpoints

Using the following abbreviations (omitting the first index Z for simplicity)

$$2\Delta n = n_{j,i+1} + n_{j-1,i+1} - n_{j,i-1} - n_{j-1,i-1}$$

$$2\Delta^2 n = n_{j,i+1} + n_{j-1,i+1} + n_{j,i-1} + n_{j-1,i-1} - 2n_{j,i} - 2n_{j-1,i}$$

$$\Delta D = D_{i+1} - D_{i-1}$$

$$\Delta v_D = v_{D,i+1} - v_{D,i-1}$$

Eq.3 becomes for the meshpoint i

$$\frac{n_j - n_{j-1}}{\Delta t} = \frac{k^2\rho^2}{2r^2\Delta\rho^2} 2\Delta^2 n + \left(\frac{k^2\rho^2\Delta D}{8r^2\Delta\rho^2} + \frac{k^2\rho D}{4r^2\Delta\rho} - \frac{k\rho v_D}{4r\Delta\rho}\right) 2\Delta n - \left(\frac{k\rho\Delta v_D}{4r\Delta\rho} + \frac{v_D}{2r} + \frac{V}{2}\right)(n_j + n_{j-1}) + Q \qquad (4)$$

It is convenient to use the abbreviation $p = k\rho/2r\Delta\rho$ (variable 'PRO' in the code) in this context, which also facilitates differentiation and integration with respect to ρ. Eq.4 is then written in the form

$$A \cdot n_{j,i-1} + (B + B') n_{j,i} + C \cdot n_{j,i+1} = D + D' \qquad (5)$$

The coefficients A, B, C and D are as follows (V_p characterizes parallel losses, which only apply to the scrape-off region)

$$A = p^2\Delta t \left(\frac{\Delta D}{2} - 2D + \frac{D\Delta\rho}{\rho} - \frac{v_D}{2p}\right) \qquad (6)$$

$$B = 1 + 4p^2\Delta t D + \frac{\Delta t}{2}\left(p\Delta v_D + V_p + \frac{v_D}{r}\right) \qquad (7)$$

$$C = -A - 4p^2\,\Delta t\, D \qquad (8)$$

$$D = -A\, n_{j-1,i-1} + (2-B)\, n_{j-1,i} + C\, n_{j-1,i+1} \qquad (9)$$

The coefficients B' and D' contain the ionisation and

recombination rates involving the next lower and next higher ionisation stages. If the system of equations is solved in ascending order, $n_{Z-1,j}$ is already known but $n_{Z+1,j}$ is not. However, recombination is often a small contribution, and in those cases, a few iterations at each time step may lead to a satisfactory result. This means

$$B' = \frac{\Delta t}{2} n_e \, (S_Z + \alpha_{Z-1}) \tag{10}$$

$$D' = - \frac{\Delta t}{2} n_e \, [n_{Z,j-1} \, (S_Z + \alpha_{Z-1}) - (n_{Z-1,j-1} + n_{Z-1,j}) \, S_{Z-1} - (n_{Z+1,j-1} + n_{Z+1,j}) \, \alpha_Z] \tag{11}$$

S_Z and α_Z are the ionisation and recombination coefficients, respectively, from and to the ionisation stage Z.

Another way of solving the system of equations is to alternate between ascending and descending order using the results of the previous time step for calculating recombination in the first and ionisation in the second cycle /1/. Then, B' and D' take the form

up:
$$B' = \Delta t \, n_e \, S_Z \tag{12}$$

$$D' = \Delta t \, n_e (n_{Z-1,j} \, S_{Z-1} + n_{Z+1,j-1} \, \alpha_Z - n_{Z,j-1} \, \alpha_{Z-1}) \tag{13}$$

down:
$$B' = \Delta t \, n_e \, \alpha_{Z-1} \tag{14}$$

$$D' = \Delta t \, n_e (n_{Z-1,j-1} \, S_{Z-1} + n_{Z+1,j} \, \alpha_Z - n_{Z,j-1} \, S_Z) \tag{15}$$

3.2 Boundary Conditions

The plasma edge R (or the outermost radius of the scrape-off layer 'RX') is positioned between the last two grid points (II and IR in the code, see Fig.1). The boundary condition consists of a (thermal) escape velocity, ie

$$- D_Z \left. \frac{dn_Z}{dr} \right|_R + n_Z \, v_{D,Z} - n_Z \, v_{esc} = 0 \tag{16}$$

Using the abbreviations (the index i refers to the outermost grid point)

$$2\bar{D} = D_{i-1} + D_i \qquad 2\bar{v}_D = v_{D,i-1} + v_{D,i}$$

$$4\bar{n} = n_{j,i-1} + n_{j,i} + n_{j-1,i-1} + n_{j-1,i}$$

and $$2\Delta n = n_{j,i} - n_{j,i-1} + n_{j-1,i} - n_{j-1,i-1}$$

one obtains

$$\frac{2\bar{D}k\rho}{rv_{esc}\Delta\rho}\, 2\Delta n + 4\bar{n}\left(1 - \frac{2\bar{v}_D}{2v_{esc}}\right) = 0 \qquad (17)$$

Similarly as before, and introducing $\Upsilon = k\rho/rv_{esc}\Delta\rho$ (called 'GRO' in the code), the coefficients A, B and D are given by

$$A = 1 - 2\bar{v}/2v_{esc} - 2\bar{D}\Upsilon \qquad (18)$$

$$B = 1 - 2\bar{v}/2v_{esc} + 2\bar{D}\Upsilon \qquad (19)$$

$$D = - An_{j-1,i-1} - Bn_{j-1,i} \qquad (20)$$

$$B' = D' = 0$$

In the code, the numerical value of v_{esc} is set equal to the diffusion coefficient. This corresponds to a fictional scrape-off width of about 1 cm (see Eq.16).

In the plasma centre $dn_Z/dr = 0$. Eq.2 is not transformed to ρ, in this case, but the transition to $r = 0$ is done in r :

$$\left.\frac{dn_Z}{dt}\right|_{r=0} = 2D_Z \left.\frac{d^2n_Z}{dr^2}\right|_{r=0} - \left(2\left.\frac{dv_D}{dr}\right|_{r=0} + V_Z\right) n_Z + Q_Z \qquad (21)$$

The result is for the coefficients B, C and D (the index i refers to r=0, Δr is the innermost radius interval ('DER')):

$$C = -2D_i \Delta t/\Delta r^2 \qquad ('PRO(1)' = 2/\Delta r^2) \tag{22}$$

$$B = 1 - C + \frac{\Delta t}{2} \left.\frac{dv_D}{dr}\right|_{r=0} \tag{23}$$

$$D = (2-B)n_{j-1,i} - Cn_{j-1,i+1} \tag{24}$$

Ionisation and recombination rates in B' and D' have to be added just as in the case of normal coefficients, ie. Eqs.10, 11 or 12-15.

After calculating the neutral source (see below), the system of coupled equations can be solved by the usual technique for a tri-diagonal matrix starting from a given initial distribution, normally all n_Z set to zero. The actual coding for the iteration technique and for the method of alternating ionisation and recombination steps is shown in Listings 2 and 3. Listing 1 contains some explanation concerning the definition of variables not mentioned here (note that the diffusion coefficients are in 'F(I)' and the drift velocities in 'E(I)').

3.3 Numerical Integration and Differentiation

Numerical differentiation in ρ is easily done, using $p_i = k\rho_i/2r_i\Delta\rho$:

$$\left.\frac{df}{d\rho}\right|_i \approx \left.\frac{\Delta f}{\Delta\rho}\right|_i = p_i\,(f_{i+1} - f_{i-1}) \tag{25}$$

For a simple integration with respect to the plasma radius, the innermost interval is calculated using Δr. The main part of the integral is performed over the ρ coordinate transforming the integrand by means of p_i. At the outermost interval, the integration must stop at the boundary, i.e. between grid points II and IR (see Fig. 1). This leads to the relation

$$\int_0^R f\,dr = \int_0^{\Delta r} f\,dr + \int_{\rho(\Delta r)}^{\rho(R)} f \frac{dr}{d\rho}\,d\rho \approx (f_1+f_2)\frac{\Delta r}{2} - \frac{1}{4}\frac{f_2}{p_2} + \frac{1}{2}\sum_2^{II}\frac{f_i}{p_i}$$

$$+ \frac{1}{16}\left(\frac{f_{IR}}{p_{IR}} - \frac{f_{II}}{p_{II}}\right) \qquad (26)$$

For integrals over the plasma area, the integrand is zero at r = 0 and therefore:

$$2\pi \int_0^R f r\,dr \approx \pi\left[\sum_2^{II}\frac{f_i r_i}{p_i} + \frac{1}{8}\left(\frac{f_{IR}\,r_{IR}}{p_{IR}} - \frac{f_{II}\,r_{II}}{p_{II}}\right)\right] \qquad (27)$$

It should be pointed out that gains from the ionisation of neutrals and parallel losses in the particle balance must just be multiplied by $\Delta\rho$ and added, i.e. using Eq.27 without the edge correction (shown for the scrape-off-layer in Fig. 1).

4. NEUTRAL SOURCE AND PARTICLE BALANCE

The code assumes a constant influx of impurities from the plasma edge present at time 0 and lasting up to time 'TØ', which can be greater than the simulated time.

Due to ionisation, the density of the neutral impurity atoms entering at the plasma edge decays with decreasing radius according to the relation

$$n_0(r) = n_{0,R}\,\frac{R}{r}\,\exp\left(-\int_r^R n_e\,S_0\,dr/v_0\right) \qquad (28)$$

(note that the neutrals have the index '1' in the code). The right-hand side of Eq.28 is calculated in SAL with $n_{0,R}$ to be determined later (returned in 'SINT'). It is possible to position the impurity source somewhere inside the plasma rather than at the edge. This position is determined by the value of 'RL' in the input data set. The total amount of ions produced per unit time and unit length ℓ is

$$Q_0 = 2\pi \int_0^{RL} n_0(r)\, n_e\, S_0\, r\, dr \qquad (29)$$

which must equal the total influx Φ_0/ℓ. A simple relation /2/ is being used for normalizing the resulting impurity density in the plasma to about one:

$$n_I = \frac{\Phi_0}{2\pi R\ell} \frac{\lambda_{Ion}}{D} \qquad (30)$$

In Eq.30, n_I is the sum of all ionisation stages and λ_{Ion} is the neutral ionisation length. By setting $n_I = 1 cm^{-3}$ and using Eqs.28, 29 and 30, $n_{0,R}$ is determined. Simply using $n_{0,R}\, v_0$, instead of Q_0 for the flux, leads to substantial numerical inaccuracies in the particle balance, particularly if the grid of mesh points is not very dense, due to the poor approximation of the exponential function in Eq.28.

The total flux entering the plasma per unit length is available in the variable 'GEZA'. It should be pointed out that, in previous versions of the code, the time sequence of the impurity pulse (from laser ablation) travelling through the plasma was calculated explicitly. This was abandoned and replaced by an instantaneous source. Also, the recombination to the neutral atoms has been removed and the respective loss in the singly ionised species has been set to zero.

Time centred particle losses at the plasma periphery are calculated after each time step by multiplying the sum of all edge densities by the escape velocity (stored in 'TSU'). For a later balance, these losses are integrated over time (in 'TVE'). Similarly, parallel particle losses are obtained by integration over the scrape-off region ('DSU' and 'DVA') and over time ('DVE'). In order to simulate recycling, a predetermined fraction of the losses η ('RCL') is added to the normal influx, which is done by multiplying the neutral density by the factor $[1 + \eta\, \Phi_{out}/\Phi_{in}]$ in the presence of external

sources and $[\eta\ \Phi_{out}/\Phi_{in}]$ in their absence. It should be mentioned that there is actually a delay of one time step between the calculation of the losses and the recycling, leading to an overshoot at rapid temporal changes. The true, time-integrated influx is kept in 'GERC'.

The particle confinement time τ_p is calculated from the equation

$$\frac{dN_I}{dt} = -\frac{N_I}{\tau_p} + \Phi_{in}, \qquad (31)$$

where N_I is the total number of impurity particles in the plasma and Φ_{in} contains the recycling flux. The integral N_I, which is performed numerically in the code ('SUM(1,M)') must be averaged between the previous and the actual time step, dN_I/dt is calculated as $(N_{I,j} - N_{I,j-1})/\Delta t$. An integration of Eq.31 yields:

$$N_I + \int_o^t \Phi_{out}\, dt = \int_o^t \Phi_{in}\, dt \qquad (32)$$

Using the total number of particles and the time integrated influx and outflux, Eq.32, divided by the right-hand side, is calculated and the result, which ideally is one, is displayed as a function of time in the last plot. To avoid a conflict with y=1 for other normalized signals, the balance is multiplied by 0.8 when plotted. As a rule, the respective result is very close to 1 or 0.8, respectively, and significant deviations are usually an indication of large oscillations or other instabilities during the run of the code.

5. PROGRAMMING OF TIME STEPS

The program carries out 'ITZ' time steps before displaying the result, if requested, and storing the respective set of parameters for the final plot of the time behaviour. More

precisely, the first cycle consists of only ITZ-1 steps (ITZ must be an even number) in order to show the results always after an ionisation run (ascending Z), because the deviations from the true values are smaller in this case. 'NTE' of such cycles are performed to cover the simulated time period T ('RTL'). The time increment is increased after each of these cycles by multiplication with a constant number α (1.2 at present), since later-on, close to a steady-state solution, longer Δt can be tolerated. If, during the simulated time, the impurity influx is switched off or CX recombination is added, rapid changes result and therefore, the time increment is reset to a small value. The program copes only with one event, unless CX is added from the very beginning (ie. 'T1' = 0).

The total time resulting from the above programming of time steps is

$$T = \Delta t\ (ITZ - 1) + \Delta t\ ITZ\ (\alpha + \alpha^2 + \ldots + \alpha^{NTE-1}) \quad (33)$$

and the initial time step Δt therefore becomes

$$\Delta t = T/\left(ITZ\ \frac{\alpha^{NTE} - 1}{\alpha - 1} - 1\right) \quad (34)$$

In the case of two time intervals, T_1 and $T-T_1$, the numbers of cycles, m and NTE-m, are distributed by means of an approximation to Eq.34

$$\frac{T_1}{T - T_1} \approx \frac{\alpha^m - 1}{\alpha^{NTE-m} - 1} \approx \alpha^{2m - NTE} \quad (35)$$

If two events at TØ and T1≠0 are requested from the program, it will reduce the time interval in both cases and not arrive at the requested simulated time.

6. PLASMA AND ANOMALOUS TRANSPORT PARAMETERS IN 'PLAD'

The electron temperature and density profiles for a particular plasma discharge are described by the following fit function

$$f(r) = [f(0) - f(R)]\,[1 - (\frac{r}{R})^{\alpha}]^{\beta} + f(R) \qquad (36)$$

where f(0) is the value on axis and f(R) is the value at the limiter or separatrix radius. Eq.36 has the advantage of allowing easy adjustment of the edge parameters without repeating the fit. It represents more or less a convenient interpolation formula between f(0) and f(R). For the electron density, f(R)/f(0) is used instead of the edge value in Eq.36. In order to cope with more structure in the profiles, in particular with hollow profiles, an alternative fit function can be used for the electron temperature

$$T_e(r) = T_e(0)\,\exp\,[\alpha(\frac{r}{R})^2 + \beta(\frac{r}{R})^4 + \gamma(\frac{r}{R})^6 + \delta(\frac{r}{R})^8] \qquad (37)$$

The code chooses the respective function automatically, because at least one of the coefficients in Eq.37 must be negative, while all coefficients in Eq.36 must be positive. A pedestal of 5eV is always provided for T_e. The variable names for the fit coefficients in Eqs.36 and 37, as well as some used later in this chapter, are explained in the comments of Listing 4.

In the scrape-off region, both electron density and temperature are assumed to fall off exponentially

$$f(r) = f(R)\,\exp[(R-r)/\lambda], \qquad (38)$$

where λ_T is an input parameter ('RLT'), while λ_n is calculated from the parallel losses in the scrape-off, characterised by a parallel confinement time $\tau_{/\!/}$, and the diffusion coefficient

$$\lambda_n = (D\ \tau_{//})^{1/2} \qquad (39)$$

The parallel losses V_p in the equations of chapter 3.1 are then given by $1/\tau_{//}$. In the present version of the program, T_i is set proportional to T_e. T_i and its derivative with respect to r are only required for calculating neoclassical transport coefficients.

For the anomalous diffusion coefficient, a constant value was mostly used, which is taken to be in units of 10^4 cm²/s. A radial variation can be requested according to the relation

$$D(r) = D(R)\ [1 + \alpha(\frac{r}{R})^{\beta}]/(1 + \alpha) \qquad (40)$$

resulting in $D(0) = D(R)/(1+\alpha)$ on axis. The anomalous drift velocity is always calculated in units of $2D(r)r/R^2$ and the multiplier was usually -1. For specific purposes, this factor has been changed or a functional relation introduced between certain limits in radius ('PV2', 'PV3', 'PV1' = multiplier in the region near the centre). However, the radial variation of the drift velocity must be tailored to the individual application.

7. NEOCLASSICAL TRANSPORT COEFFICIENTS

For a simple calculation of neoclassical transport, the relation of Rutherford et al. /3/ for a single impurity species has been used (in cgs units):

$$\Gamma_I = \frac{4(2\pi)^{1/2}}{3} \frac{e^2 c^2 (m_i)^{1/2}\ \ell n\ \Lambda}{(kT)^{3/2}\ B_T^2} \cdot$$

$$\left[2q^2 \left(Z\ n_Z \frac{dp_i}{dr} - n_i \frac{dp_Z}{dr}\right) - 3q^2 Z\ n_i\ n_Z \frac{dkT}{dr}\right] , \qquad (41)$$

where q is the safety factor of the plasma and n_i and p_i are the density and the pressure of the plasma ions. T is the ion temperature which is assumed to be the same for all species. The constant in front of the bracket is, in the case of hydrogen, k = 0.0113 cgs units ($\ell n\Lambda$ = 16), if T is in units of eV. Then, the following expressions result for the neoclassical spreading and drift terms:

$$D_n = 2n_i k q^2/(T^{1/2} B_T^2) \qquad (42)$$

$$v_{Z,n} = 2kq^2/(T^{1/2} B_T^2).\left[Z \frac{dn_i}{dr} - \frac{n_i}{T}\frac{dT}{dr}\left(1 + \frac{Z}{2}\right)\right] \qquad (43)$$

In the code, q=1 on axis and q=4 at the boundary is used with a fit function in between. Then, $kq^2/T_i^{1/2} B_T^2$ is returned by PLAD for a toroidal field of 2T (in 'Q(I)'), and the transport parameters are calculated in the main program according to Eqs.42 and 43. Adjustments for other fields and deuterium instead of hydrogen can be made in 'QF'. The derivative $dv_{Z,n}/dr$ at r=0, required for the inner boundary condition, is done when calculating B(1) (Eq.23) using the respective second derivatives. Neoclassical and anomalous diffusion and drift parameters are added in 'F(I)' and 'E(I)', respectively.

Without any impurities, the hydrogen ion density is equal to the electron density from the experiment. As the impurities build up in the plasma as a consequence of neoclassical transport, they reduce the number of plasma ions there. Therefore, a reduction in n_i and dn_i/dr must be taken into account in the calculation of the neoclassical transport coefficients. The plasma ion density is given by the quasi- neutrality

$$n_i = n_e - \Sigma\, n_Z Z \,, \qquad (44)$$

provided that the absolute number of impurity ions n_Z is known. In the code, the impurity level can be adjusted by the input variable 'VGR'. It is calculated as $n_I.n_e(0).VGR$, where

n_I is the code result for the sum of all ionisation stages (the impurity concentration on axis is $n_I(0).VGR$). The modified plasma ion profile is evaluated after each time step according to Eq.44, and this profile and its radial derivative are used for calculating D_n and $v_{Z,n}$ for the next time step. Due to this quasi-neutrality condition, neoclassical spreading and drift coefficients are reduced and accumulation mitigated. It should be pointed out, however, that this procedure introduces non-linear terms in the equations. Therefore, higher impurity concentrations may lead to numerical instabilities during the run of the code and may require a large number of timesteps.

8. ATOMIC DATA AND RADIATION

In the subroutine SAL, ionisation rate coefficients are calculated according to Lotz /4/. Two subshells are used only, and the constants required are read from the atomic data set. For the exponential integral $E_1(y)$, the following approximation has been used:

$$\exp(y)\ E_1(y) = \ln\left(1 + \frac{1}{y}\right) - \frac{0.4}{(1 + y)^2} \qquad (45)$$

For Na-like ions, where inner-shell ionisation is known to be important, correction factors to the Lotz formula are implemented by means of modified coefficients in the data sets.

Radiative recombination is treated in the usual way (see eg. von Goeler /5/) calculating the contribution from the valence shell with an effective quantum number $n^* = (Z^2 E_H/E_Z)$, and summing up the contributions from the excited levels, which are considered to be hydrogenic. In SAL, this sum is calculated up to the collision limit (Griem /6/) but stops at 20

$$n_c = 126.Z^{12/17}\ (T_e/E_H)^{1/17} \qquad (46)$$

For dielectronic recombination, the Burgess prescription /7/ is applied, taking into account modification by Merts et al. /8/.

Excitation energies and oscillator strengths for two effective lines from the atomic data set are used, which, later on, serve for the calculation of the total line radiation from this ionisation stage, too, making use additionally of the effective, average Gaunt factors of these transitions. In order to distinguish $\Delta n=0$ and $\Delta n\neq 0$ transitions, the Gaunt factor is entered as a negative number in the latter case. Dielectronic recombination can be bypassed by setting the multiplier 'ALM' equal to zero.

A density dependence of dielectronic recombination rates has been adopted following Post et al. /9/. It can be bypassed by using a negative number for the dielectronic recombination multiplier 'ALM'.

The total charge-exchange recombination rate coefficient is calculated according to Puiatti et al. /10/, ie., assuming a scaling of the CX cross-section σ_{cx} with $Z^{1.07}$. The scaling of the relevant energy with $Z^{-0.464}$ has not been implemented because of the weak dependence of σ_{cx} on energy in the relevant range. Instead, the value for fully stripped oxygen and a typical beam energy of 40 keV/amu has been used. It turns out that in the calculation of σv, the variation of the velocity with energy is just compensated, within reasonable limits, by the respective variation of σ going into the opposite direction. Therefore, to a first approximation, the value of σv used also applies to 1/2 and 1/3 of the beam energy and even to the halo neutrals at a few keV ion temperature. The radial profile of the beam neutrals is, in this rudimentary treatment, taken to be constant (in 'ADH') or approximated by a simple formula programmed directly into SAL.

Line-of-sight integrals I* of specific impurity lines, normalized to a density of 1 cm^{-3} on axis, are calculated by the code according to a van-Regemorter-type formula using the excitation energies ΔE_L provided by the atomic data set (in 'DEE'). The code does not, however, divide by ΔE_L, because it

might be zero. Furthermore, it uses an oscillator strength f and a Gaunt factor $\bar{g}$ for hydrogen L_α (0.42 and 0.2) and divides by 10^6 in order to obtain reasonable numbers for the line integrals. Thus, the actual intensity of the line, in photons/cm²s, for impurity density 1 cm^{-3} on axis is

$$I = 10^6 \cdot \frac{f_L \cdot \bar{g}_L}{0.084} \cdot \frac{B_L}{\Delta E_L} I^* \ , \qquad (47)$$

where B_L is the branching ratio of the observed transition. The factors in front of I* have been called K_L and a set of these K_L values for the most important impurity lines has been compiled. If the excitation rate coefficient X_L is known, K_L is calculated from the relation

$$K_L = 7.4.10^{11} \ X_L \ T_e^{1/2} \ \exp(\Delta E_L/T_e).B_L \qquad (48)$$

for the approximate temperature at which the respective ion is most abundant in the plasma. In the case of hydrogen-like ions, the direct population of the excited levels by recombination is added to the electron impact excitation.

Total line radiation is again obtained from a van-Regemorter type formula, but now using the f and $\bar{g}$ values from the data set for two effective lines. Recombination radiation is calculated by multiplying the total recombination rate by the ionisation energy. This is an underestimate, particularly for the dielectronic recombination. However, recombination radiation is usually a very small contribution to the radiated power. Finally, bremsstrahlung is added according to the well-known formula.

The total radiation is calculated in units of 10^{-13} W/cm³ for the respective impurity density in the code. The integral over the plasma area, printed underneath the plots, is normalized to density one and given in 10^{-13} W/cm. For the line radiation,

only radiation above a given energy threshold ('ELI') can be considered (filter cut-off in soft X-ray measurements). If this threshold is positive, continuum radiation is added, and in the case of a negative value it is not.

The input data, required from the file, are described in the comments to the individual read statements in Listings 1 and 4. One of the nickel data sets presently used for JET is shown in Listing 5.

9. CONTROLS DURING THE RUN AND GRAPHIC OUTPUT

At the beginning of the run, the code requests the file name of the data set containing plasma and atomic data information. The output is determined by an integer output control variable. If it is zero, the data will be written into a file by the subroutine DATO. However, at present, this option is only used to produce a selected list of densities and rate coefficients, and the structure of a more elaborate output must be implemented, if required. Input of 1 or 2 for the control variable will result in plots after each time cycle, the emission shells of which are not normalized or normalized, respectively. -1 or -2(=default) tell the code to skip these plots and go to the last distribution at time T. It is also possible to display the first plot after a specified time interval, entered as the next variable, which is normally zero and is set equal to T if it is greater. Finally, an optional energy limit for total radiation calculations can be put in, which is otherwise taken to be zero.

The program will come up with the display of emission shells, of the total density profile (index NP) and of the radial distribution of radiation losses (index NO) after the requested time interval. Then, input of a control character determines the next step. If there is no input ('return') the program will proceed to the next time cycle or, if it was the last

cycle, to the display of intensities and total radiation versus simulated time (note that the index of the total radiation is now NP). If again there is no input, it will alternate between radial plot and time plot. 'I' causes the intensity plot to be repeated with the optional choice of lower and upper limits (z1,z2) for the ionisation stages and a different full-scale value (ymax). The latter data remain valid until changed in a subsequent input step. 'D' displays ground state densities and the radial variation of ΔZ_{eff} (index NO) for a concentration $n_I(0)$.VGR on axis:

$$Z_{eff} = 1 + \Delta Z_{eff} = 1 + \Sigma\, n_Z Z(Z - 1)/n_e \qquad (49)$$

The same options apply as before. 'L' results in a logarithmic plot of the intensities (2 decades, 0 corresponds to 0.01, 0.5 to 0.1, and 1 to 1). The 'L' option is essentially used for showing the exponential decay of impurity line radiation after the source is switched off. 'C' entered during the run causes the program to go to the last plot. If it was already displaying the time traces, the program is re-entered at the beginning for further calculations, using the last result as the initial values. This option is useful to calculate a corona equilibrium distribution after the radial distribution of the total impurity density is established. Finally, 'E' at any time ends the run of the program. The variable t tells the program to proceed to another simulated time. If the time plot is repeated, t will become the starting value. If there is no input, t is set to zero.

The legend of the plots is mostly self-evident. The first three lines underneath all plots are identical, except for the displayed control character and plot description at the beginning of the first line. tØ is the time when the source is switched off, a is the radius of the plasma boundary, including scrape-off width DB, nØ is the neutral hydrogen density for CX, ion the ionisation length, c the particle confinement time.

In the fourth line, a selection of four I* values, as well as the relevant excitation energies, are displayed under the intensity plots. It starts at the ion index z1 (normally 1, note that I*(1) contains the particle balance). By repeating the plot and changing z1, all I* results can be obtained. Under the plot of ground state densities, the n_e and T_e fall-off lengths are given. Furthermore, the initial value of Δt (DET), the neoclassical multiplier 'QF' and the multiplier for dielectronic recombination (drc*) is printed.

Several examples of the use of STRAHL on the JET NORD-500 are given in Figs.2-12. Complete listings of STRAHL, PLAD, SAL, INTP, RDPAR and CRT011 are appended to this report.

REFERENCES

/1/ K Lackner et al., Z. Naturforsch. 27a(1982) 931.

/2/ W Engelhardt and W Feneberg, J. Nucl. Mater. 76-77(1978) 518.

/3/ P H Rutherford et al., Proc. of the Int. Symposium on Plasma Wall Interaction, Jülich, FRG, 1976.

/4/ W Lotz, IPP 1/62 (1967) and IPP 1/76 (1968).

/5/ S von Goeler et al., Nucl. Fusion 15(1975) 301.

/6/ H R Griem, Plasma Spectroscopy, McGraw-Hill 1964.

/7/ A Burgess, Astrophys.J. 141(1968) 1588.

/8/ A L Merts et al., LA-6220-MS(1976).

/9/ D E Post et al., At. Data Nucl. Data Tables 20(1977) 397.

/10/ M E Puiatti et al., Plasma Phys, 23(1981) 1075).

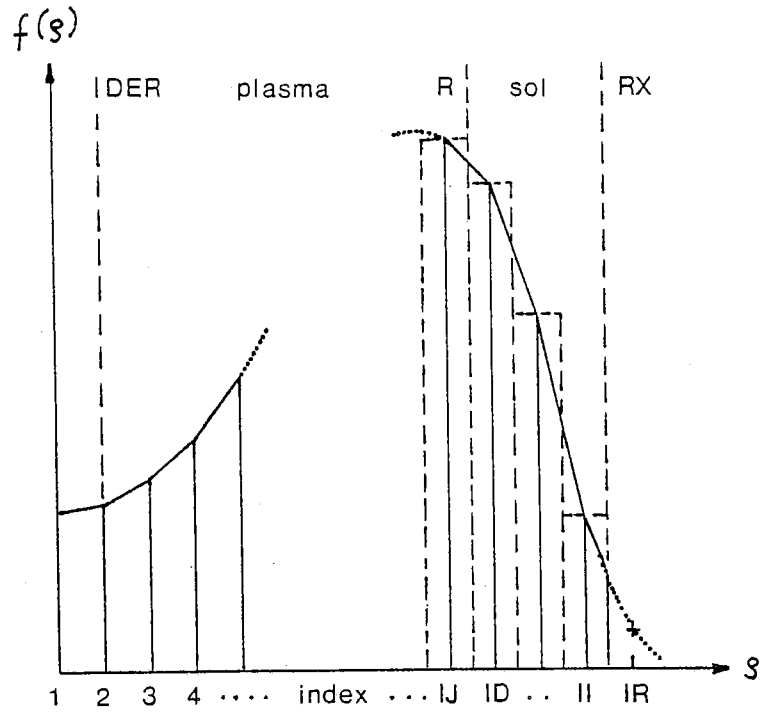
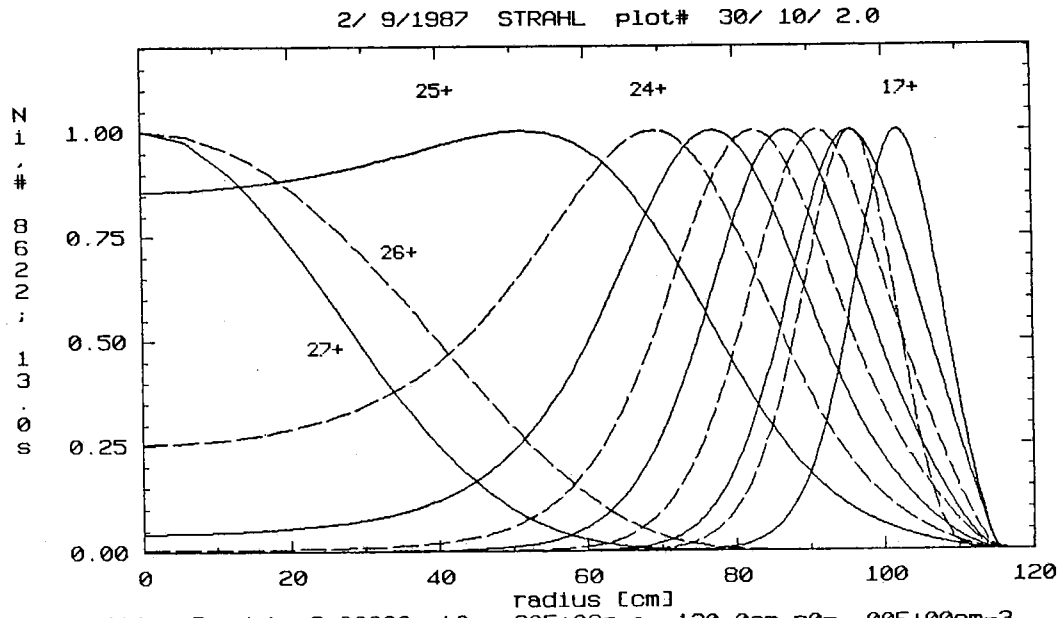
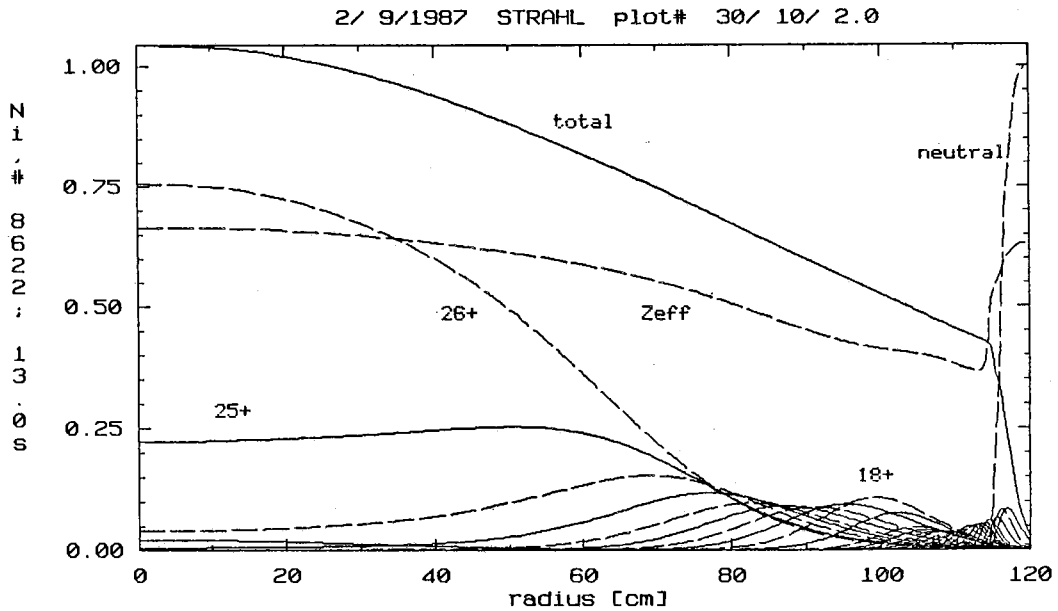


Fig. 1: Definition of indices along the ρ-coordinate, and explanation of the integration methods.



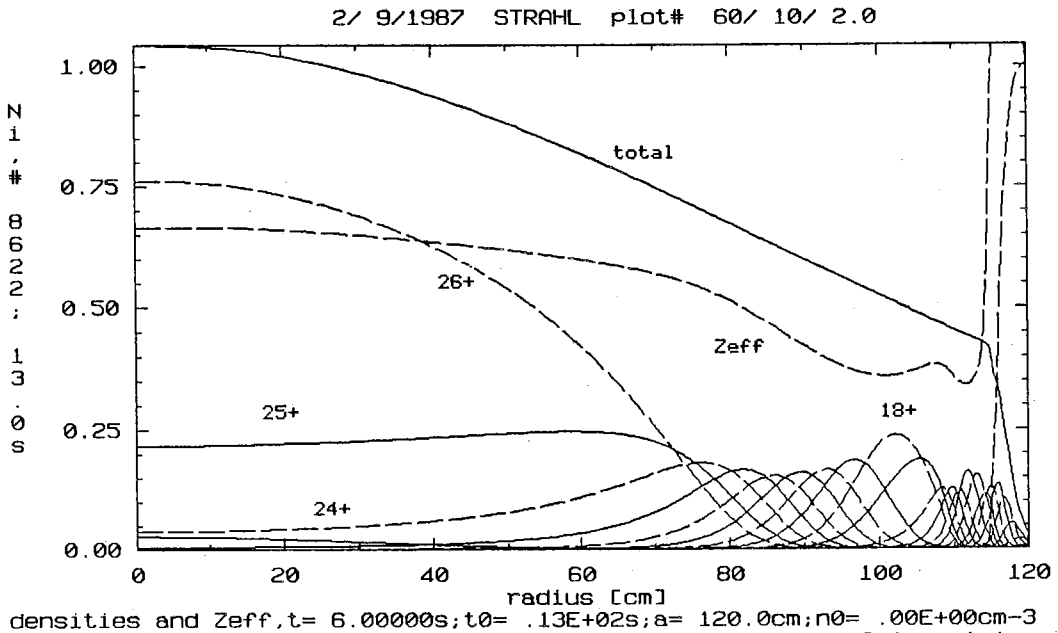
I intensities+Prad,t= 3.00000s;t0= .20E+02s;a= 120.0cm;n0= .00E+00cm-3
Te(0)= 3200.eV;ne(0)= .32E+14cm-3;D(0)=10000.cm2/s;VD(0)=-1.0;ion= 1.1cm
DB= 5.cm;c= 4.1ms;RCL= .00;Prad= .98E+05*E-13W/cm;n= 1.04;Elim= 0.eV
I*= 18- 42.: 1.730 19- 820.: .727 20- 0.: 3.336 21- 65.: 3.146

Fig. 2: Example of normalized nickel emission shells. Usual transport.



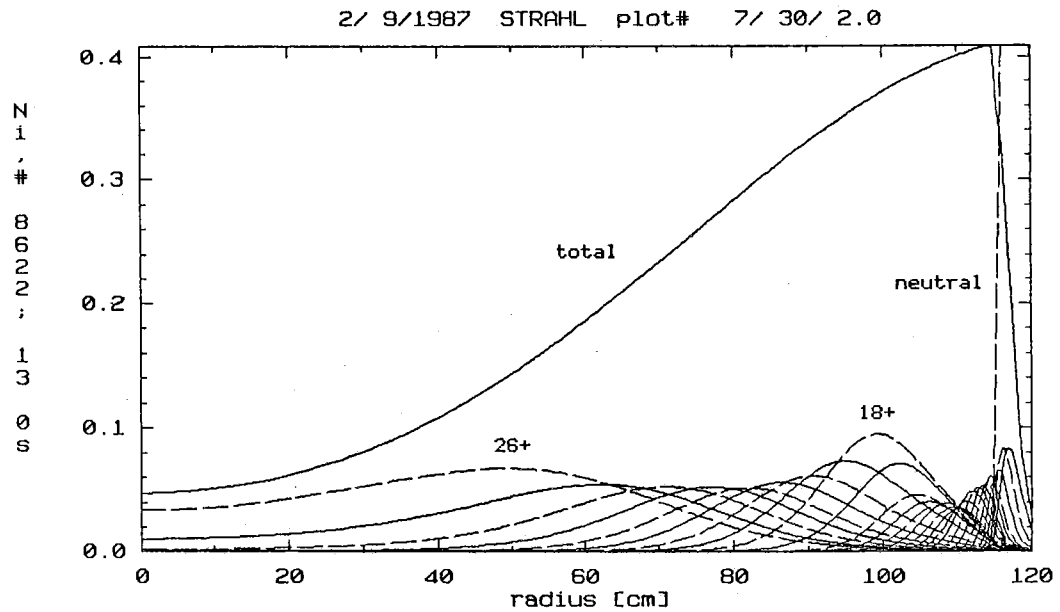
D densities and Zeff,t= 3.00000s;t0= .20E+02s;a= 120.0cm;n0= .00E+00cm-3
Te(0)= 3200.eV;ne(0)= .32E+14cm-3;D(0)=10000.cm2/s;VD(0)=-1.0;ion= 1.1cm
DB= 5.cm;c= 4.1ms;RCL= .00;Prad= .98E+05*E-13W/cm;n= 1.04;Elim= 0.eV
s:1.0/4.0cm;DET= .25E-03s;QF= .0;drc* 1.0;Ti(0)= 2000.eV;v0= -.2E+06cm/s

Fig. 3: Steady-state ground state distributions and Zeff. Usual transport



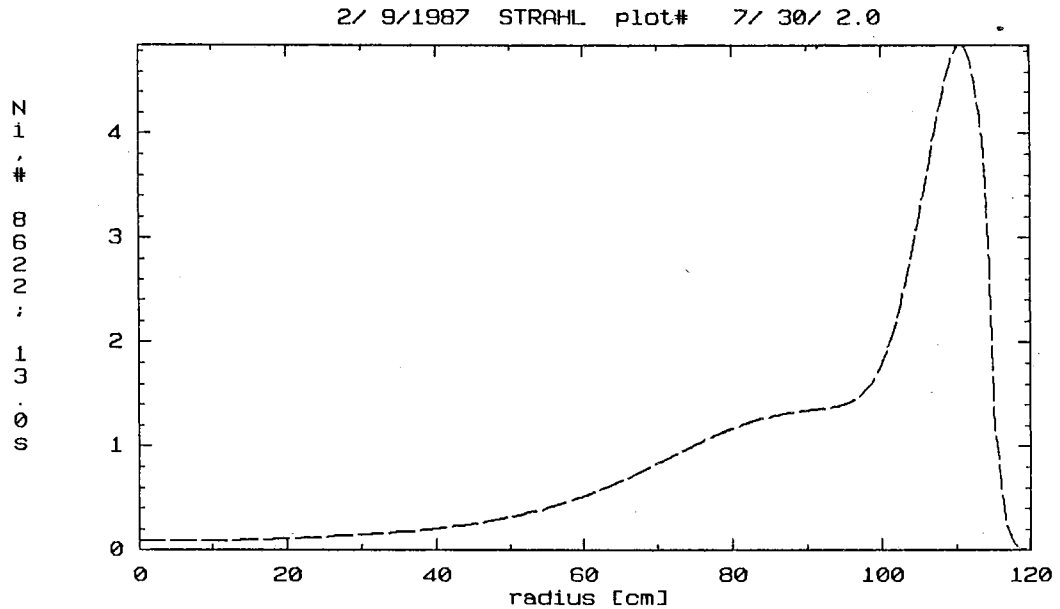
D densities and Zeff,t= 6.00000s;t0= .13E+02s;a= 120.0cm;n0= .00E+00cm-3
Te(0)= 3200.eV;ne(0)= .32E+14cm-3;D(0)= 0.cm2/s;VD(0)=-1.0;ion= 1.1cm
DB= 5.cm;c=*****ms;RCL= .00;Prad= .90E+05*E-13W/cm;n= 1.04;Elim= 0.eV
s:1.0/4.0cm;DET= .25E-03s;QF= .0;drc* 1.0;Ti(0)= 2000.eV;v0= -.2E+06cm/s

Fig. 4: Corona ground state distributions and Zeff.



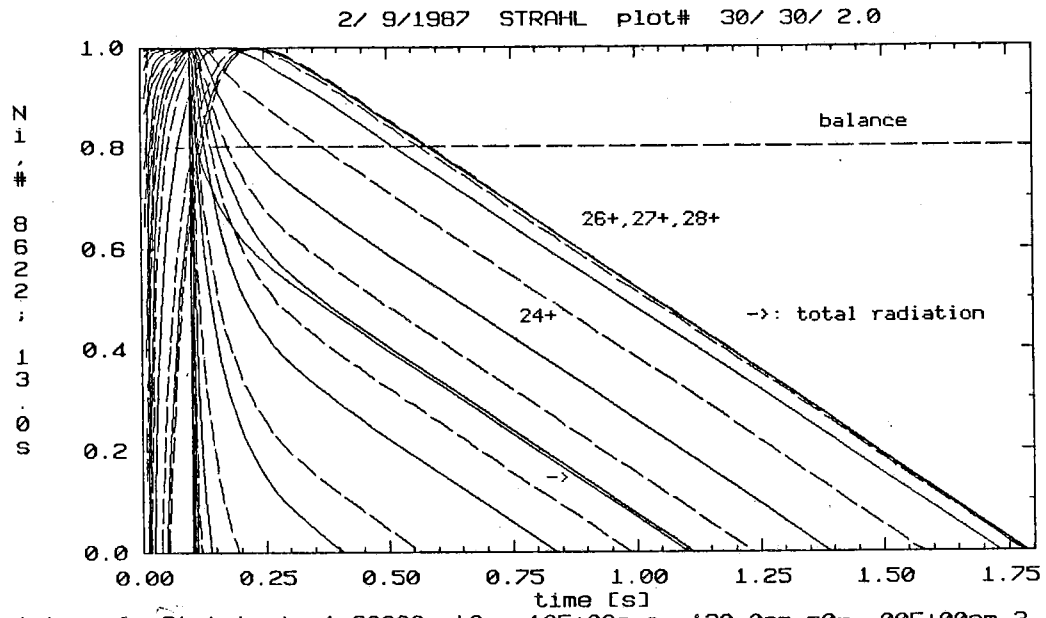
D densities and Zeff,t= .10000s;t0= .10E+00s;a= 120.0cm;n0= .00E+00cm-3
Te(0)= 3200.eV;ne(0)= .32E+14cm-3;D(0)=10000.cm2/s;VD(0)=-1.0;ion= 1.1cm
DB= 5.cm;c= 1.6ms;RCL= .00;Prad= .14E+07*E-13W/cm;n= .05;Elim= 0.eV
s:1.0/4.0cm;DET= .17E-03s;QF= .0;drc* 1.0;Ti(0)= 2000.eV;v0= -.2E+06cm/s

Fig. 5: Nickel ground state distributions 0.1 s after start of injection.



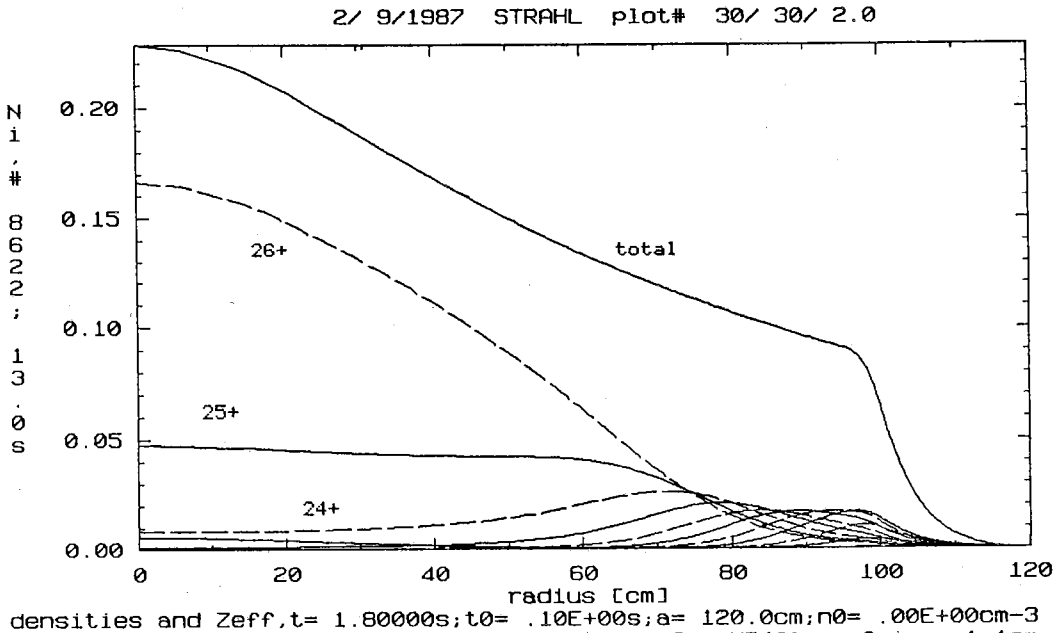
I intensities+Prad,t= .10000s;t0= .10E+00s;a= 120.0cm;n0= .00E+00cm-3
Te(0)= 3200.eV;ne(0)= .32E+14cm-3;D(0)=10000.cm2/s;VD(0)=-1.0;ion= 1.1cm
DB= 5.cm;c= 1.6ms;RCL= .00;Prad= .14E+07*E-13W/cm;n= .05;Elim= 0.eV
I*= 31- 10.: 4.296 32- 15.: .000 33- 20.: .000 34- 30.: .000

Fig. 6: Radiation profile for the case of Fig. 5. Units are W/cm3 for density as calculated multiplied by 1.E13.



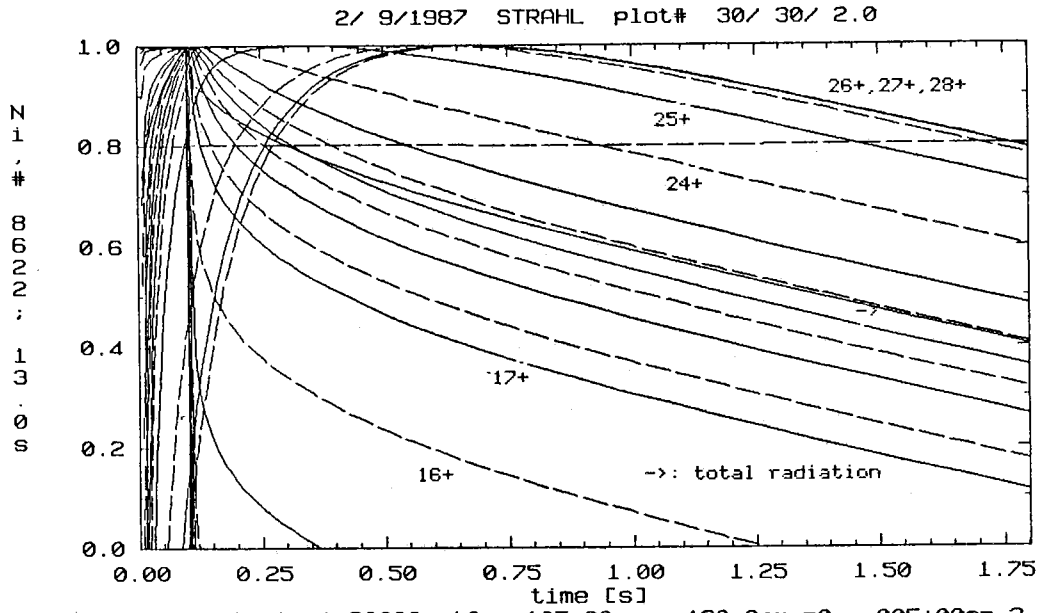
L integrals+Ptot to t= 1.80000s;t0= .10E+00s;a= 120.0cm;n0= .00E+00cm-3
Te(0)= 3200.eV;ne(0)= .32E+14cm-3;D(0)=10000.cm2/s;VD(0)=-1.0;ion= 1.1cm
DB= 5.cm;c=349.4ms;RCL= .00;Prad= .34E+05*E-13W/cm;n= .00;Elim= 0.eV
I*= 1- 0.: 1.002 2- 0.: .000 3- 0.: .000 4- 0.: .000

Fig. 7: Decay of nickel radiation after the end of injection on a log. scale. The slope is equal to the particle confinement time.



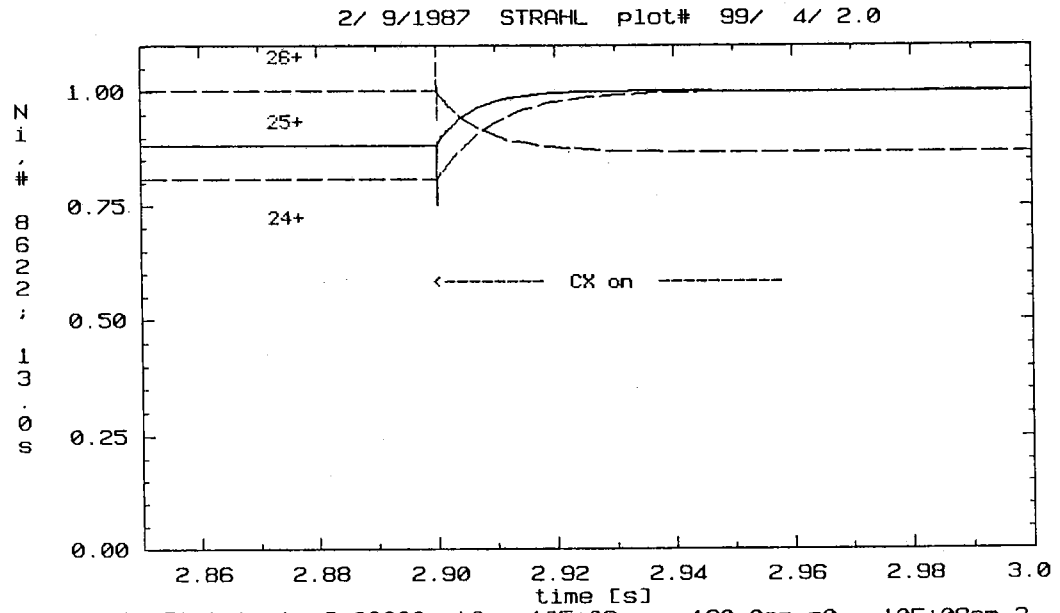
D densities and Zeff,t= 1.80000s;t0= .10E+00s;a= 120.0cm;n0= .00E+00cm-3
Te(0)= 3200.eV;ne(0)= .32E+14cm-3;D(0)= 1000.cm2/s;VD(0)= .0;ion= 1.1cm
DB= 5.cm;c=950.8ms;RCL= .00;Prad= .46E+05*E-13W/cm;n= .23;Elim= 0.eV
s:1.0/4.0cm;DET= .17E-03s;QF= .0;drc* 1.0;Ti(0)= 2000.eV;v0= -.2E+06cm/s

Fig. 8: Decaying density distribution. D(0)=0.1 m2/s, D(a)=1.0 m2/s. High inward drift (25 m/s) at the edge (outside 95 cm).



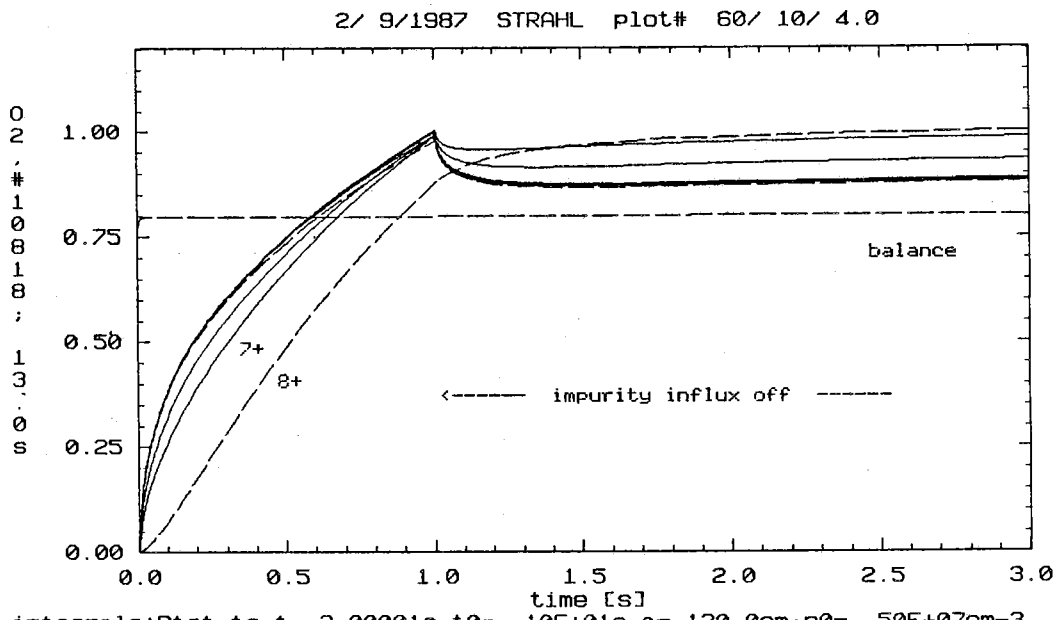
L integrals+Ptot to t= 1.80000s;t0= .10E+00s;a= 120.0cm;n0= .00E+00cm-3
Te(0)= 3200.eV;ne(0)= .32E+14cm-3;D(0)= 1000.cm2/s;VD(0)= .0;ion= 1.1cm
DB= 5.cm;c=950.8ms;RCL= .00;Prad= .46E+05*E-13W/cm;n= .23;Elim= 0.eV
I*= 1- 0.: 1.001 2- 0.: .000 3- 0.: .000 4- 0.: .000

Fig. 9: Decay of intensities (log. scale) with transport parameters of Fig. 8. Confinement is three times higher than normal.



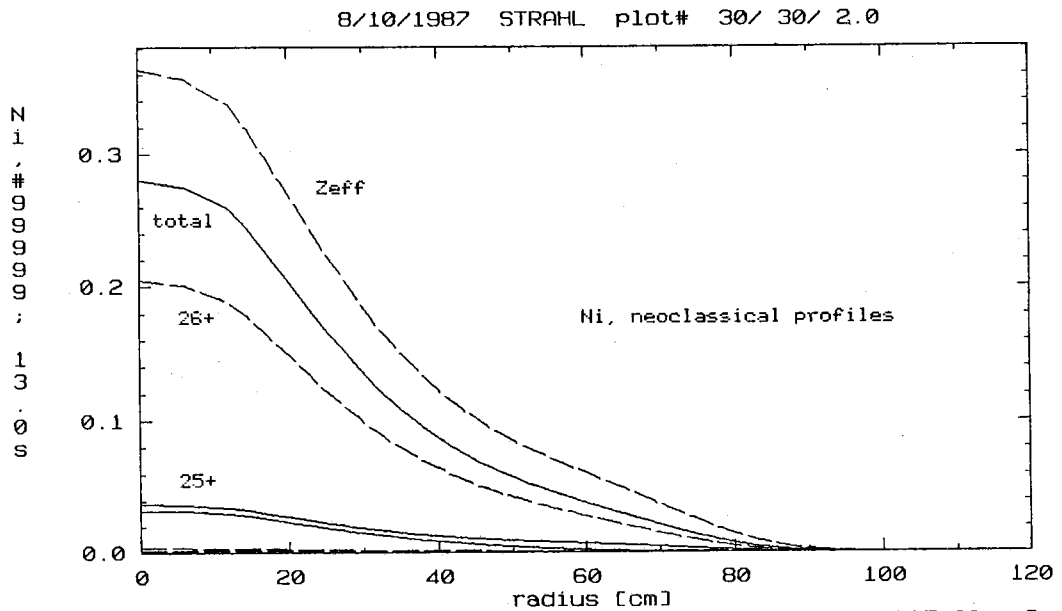
I integrals+Ptot to t= 3.00000s;t0= .10E+02s;a= 120.0cm;n0= .10E+08cm-3
Te(0)= 3200.eV;ne(0)= .20E+14cm-3;D(0)=10000.cm2/s;VD(0)=-1.0;ion= 1.7cm
DB= 5.cm;c= 6.1ms;RCL= .00;Prad= .70E+05*E-13W/cm;n= .97;Elim= 0.eV
I*= 25- 107.: 9.343 26- 75.:19.557 27-7800.: 2.043 28-8000.: .032

Fig. 10: Change of steady-state line intensities by charge-exchange recombination. More cycles for better time resolution.



I integrals+Ptot to t= 3.00001s;t0= .10E+01s;a= 120.0cm;n0= .50E+07cm-3
Te(0)= 4260.eV;ne(0)= .61E+14cm-3;D(0)=10000.cm2/s;VD(0)=-1.0;ion= 2.3cm
DB= 7.cm;c= 9.1ms;RCL=1.00;Prad= .53E+04*E-13W/cm;n= 60.01;Elim= 0.eV
I*= 1- 0.: .999 2- 26.: .017 3- 14.: .032 4- 22.: .054

Fig. 11: Oxygen line intensities, 100% recycling. The oxygen source is switched off at 1.0 s.



D densities and Zeff,t=99.99980s;t0= .30E+02s;a= 120.0cm;n0= .00E+00cm-3
Te(0)= 5000.eV;ne(0)= .40E+14cm-3;D(0)= 38.cm2/s;VD(0)= .0;ion= 1.6cm
DB= 7.cm;c=*****ms;RCL= .00;Prad= .11E+05*E-13W/cm;n= .28;Elim= 0.eV
s:2.6/4.0cm;DET= .22E-01s;QF=1.0;drc* 1.5;Ti(0)= 3500.eV;v0= -.2E+06cm/s

Fig. 12: Neoclassical calculation for JET. Note the long timescales. Impurity influx switched off at t=30 s. VGR=0.002.

Listing 1: Important variables in program STRAHL / file READ statements

variable		meaning
R	real	raius of main plasma (cm, index 1 to IJ of RR)
DB	real	width of scrape-off-layer (cm, index ID to II)
RX	real	sum of R and DB (cm, between II and IR)
T,TOO	real	actual time, time of last run if reentry (s)
TO,T1	real	impurity influx off, CX recombination on (s)
DET	real	delta-t (s) (multiplied by TINC after each cycle)
DETA	real	delta-t for (optional) second time interval
DER	real	delta-r (cm, centre to first grid point)
DRO	real	delta-rho (cm**k)
T2R	real	second derivative of Ti at r=0 (ev/cm**2)
V	real	(thermal) escape velocity (cm/s)
RR	real(101)	radius (cm)
RO	real(101)	rho = radius**k (cm**k)
TE	real(101)	electron temperature (eV)
TI	real(101)	ion temperature (eV)
TDR	real(101)	derivative w:r.t. r of ion temperature (eV/cm)
RNE	real(101)	electron density (cm**-3)
DK	real(101)	anomalous spreading coefficient (cm**2/s)
VD	real(101)	anomalous drift (cm/s)
DV	real(101)	parallel losses (1/s)
F	real(101)	sum of anomalous and neoclassical spreading terms (cm**2/s)
E	real(101)	sum of anomalous and neoclassical drift (cm/s)
S	real(29,101)	ionisation rate coefficient (cm**3/s)
AL	real(29,101)	recombination rate coefficient (cm**3/s)
RN	real(31,101)	actual densities (cm-3); index "1"=neutral to "LZ"=hydrogen-like,"NN"=fully stripped. The proton density is in RN(NP,I), its derivative (cm-4) in RN(NO,I) (index=1-2.....LZ-NN-NP-NO).
RA	real(31,101)	densities of previous time step. Indices as RN. For output RA(2...NN,I) is overwritten by line radiation. RA(NP,I) contains the total density, RA(NO,I) total radiation, RA(1,I) norm. neutrals.
SUM	real(31,100)	line-of-sight integrals in SUM(2...NN,M). Total radiation in SUM(NP,M), time in SUM(NO,M), particle balance in SUM(1,M).

*** see text for units of line and total radiation ***

```
C-----------------------------------------------------------------------
      READ(2,1) IEL,K,IR
1     FORMAT(A2,F4.0,I4)
C
C  IMPURITY ELEMENT "IEL", COORDINATE R**"K", NO. OF GRIDPOINTS "IR"
C
C  NUMBER OF TIME STEPS "NTE", SIMULATED TIME "RTL" (S), STEPS BETWEEN
C  PLOTS "ITZ", XMIN (FOR PLOT), POSITION OF IMPURITY SOURCE "RL",
C  PLASMARADIUS "R", SCRAPE-OFF WIDTH "DB",RECOMBIN. FUDGE FACT."ALM"
C  NEUTRAL DENSITY FOR CX-REKOMBINATION "ADH", CX TIME "T1"
C
C  D-MULTIPLIER "FAKT", NEOCL.-FACT."QF", NORMALIZED DRIFT VELOCITY "VD",
C  IMPURITY LEVEL (FOR PROTON DENSITY/NEOCL. TRANSP.) "VGR", IMPURITY
C  PULSE LENGTH "TO",-VELOCITY "VO"(NEGATIVE), "TAU" PARALLEL, TE FALL-
C  OFF LENGTH "RLT", RECYCLING FACTOR "RCL"
C
      READ(2,*) NTE,RTL,ITZ,XMIN,RL,R,DB,ALM,ADH,T1
      READ(2,*) FAKT,QF,VD(1),VGR,TO,VO,TAU,RLT,RCL
```

Listing 2: coefficients for the iteration scheme / solution of equations

```
      DO 700 LV=2,NN
C
      NZ=LV
-----------------------------------------------------------------------
C
C COEFFICIENTS AT R=0
C
      C(1)=-DET*F(1)*PRO(1)
      B(1)=1.-C(1)+DET/2.*
     >(Q(1)*(ZZ*(RN(NO,2)+RA(NO,2))/DER-(1.+.5*ZZ)*T2R/TI(1)*
     >(RN(NP,1)+RA(NP,1)))+VD*2.*DK(1)/RX**2)
      D(1)=RA(NZ,1)*(2.-B(1))-RA(NZ,2)*C(1)
C
      BET(1)=DET/2*RNE(1)*(S(NZ,1)+AL(NZ-1,1))
      B(1)=B(1)+BET(1)
C
      GAM(1)=-DET/2*RNE(1)*(RA(NZ,1)*(S(NZ,1)+AL(NZ-1,1))-
     >(RA(NZ-1,1)+RN(NZ-1,1))*S(NZ-1,1)-
     >(RA(NZ+1,1)+RN(NZ+1,1))*AL(NZ,1))
      D(1)=D(1)+GAM(1)
C
C  COEFFICIENTS AT R=R OR AT R+DB
C
      A(IR)=1.-.5*(E(IR-1)+E(IR))/V-GRO*(F(IR-1)+F(IR))
      B(IR)=1.-.5*(E(IR-1)+E(IR))/V+GRO*(F(IR-1)+F(IR))
      D(IR)=-RA(NZ,IR-1)*A(IR)-RA(NZ,IR)*B(IR)
C
C  NORMAL COEFFICIENTS
C
      DO 400 I=2,II
      VDA=DET*PRO(I)**2
      VDR=4.*VDA*F(I)
      VV=DET/2.*(DV(I)+PRO(I)*(E(I+1)-E(I-1))+E(I)/RR(I))
      A(I)=VDA*(F(I)*(DRO/RO(I)-2.)+.5*(F(I+1)-F(I-1)-E(I)/PRO(I)))
      B(I)=1.+VDR+VV
      C(I)=-VDR-A(I)
      D(I)=-RA(NZ,I-1)*A(I)+RA(NZ,I)*(2.-B(I))-RA(NZ,I+1)*C(I)
C
      BET(I)=DET/2*RNE(I)*(S(NZ,I)+AL(NZ-1,I))
      B(I)=B(I)+BET(I)
C
      GAM(I)=-DET/2*RNE(I)*(RA(NZ,I)*(S(NZ,I)+AL(NZ-1,I))-
     >(RA(NZ-1,I)+RN(NZ-1,I))*S(NZ-1,I)-
     >(RA(NZ+1,I)+RN(NZ+1,I))*AL(NZ,I))
400   D(I)=D(I)+GAM(I)
C
       BET(1)=B(1)
       GAM(1)=D(1)/B(1)
C
       DO 500 I=2,IR
       BET(I)=B(I)-(A(I)*C(I-1))/BET(I-1)
500    GAM(I)=(D(I)-A(I)*GAM(I-1))/BET(I)
C
       RN(NZ,IR)=GAM(IR)
C
       DO 600 I=2,IR
       J=IR+1-I
600    RN(NZ,J)=GAM(J)-(C(J)*RN(NZ,J+1))/BET(J)
```

Listing 3: coefficients for alternating ion. and recomb. cycles

```
      DO 700 LV=2,NN
C
C  ALTERNATING UP AND DOWN
C
      NZ=LV
      IAW=MOD(N,2)+1
      IF(IAW .EQ. 1) NZ=NN-LV+2
C
-----------------------------------------------------------------------
C  COEFFICIENTS AT R=0
C
      C(1)=-DET*F(1)*PRO(1)
      B(1)=1.-C(1)+DET/2.*
     > (Q(1)*(ZZ*(RN(NO,2)+RA(NO,2))/DER-(1.+.5*ZZ)*T2R/TI(1)*
     > (RN(NP,1)+RA(NP,1)))+VD(1)*2.*DK(1)/RX**2)
      D(1)=RA(NZ,1)*(2.-B(1))-RA(NZ,2)*C(1)
C
      GO TO (732,731) IAW
C
731   B(1)=B(1)+DET*RNE(1)*S(NZ,1)
      D(1)=D(1)-DET*RNE(1)*(RA(NZ,1)*AL(NZ-1,1)-RA(NZ+1,1)*AL(NZ,1)-
     > RN(NZ-1,1)*S(NZ-1,1))
      GO TO 740
C
732   B(1)=B(1)+DET*RNE(1)*AL(NZ-1,1)
      D(1)=D(1)-DET*RNE(1)*(RA(NZ,1)*S(NZ,1)-RA(NZ-1,1)*S(NZ-1,1)-
     > RN(NZ+1,1)*AL(NZ,1))
C
C  COEFFICIENTS AT R=R OR AT R+DB
C
740   A(IR)=1.-.5*(E(IR-1)+E(IR))/V-GRO*(F(IR-1)+F(IR))
      B(IR)=1.-.5*(E(IR-1)+E(IR))/V+GRO*(F(IR-1)+F(IR))
      D(IR)=-RA(NZ,IR-1)*A(IR)-RA(NZ,IR)*B(IR)
C
C  NORMAL COEFFICIENTS
C
      DO 400 I=2,II
      VDA=DET*PRO(I)**2
      VDR=4.*VDA*F(I)
      VV=DET/2.*(DV(I)+PRO(I)*(E(I+1)-E(I-1))+E(I)/RR(I))
      A(I)=VDA*(F(I)*(DRO/RO(I)-2.)+.5*(F(I+1)-F(I-1)-E(I)/PRO(I)))
      B(I)=1.+VDR+VV
      C(I)=-VDR-A(I)
      D(I)=-RA(NZ,I-1)*A(I)+RA(NZ,I)*(2.-B(I))-RA(NZ,I+1)*C(I)
C
      GO TO (721,711) IAW
C
711   B(I)=B(I)+DET*RNE(I)*S(NZ,I)
      D(I)=D(I)-DET*RNE(I)*(RA(NZ,I)*AL(NZ-1,I)-RA(NZ+1,I)*AL(NZ,I)-
     > RN(NZ-1,I)*S(NZ-1,I))
      GO TO 400
C
721   B(I)=B(I)+DET*RNE(I)*AL(NZ-1,I)
      D(I)=D(I)-DET*RNE(I)*(RA(NZ,I)*S(NZ,I)-RA(NZ-1,I)*S(NZ-1,I)-
     > RN(NZ+1,I)*AL(NZ,I))
C
400   CONTINUE
C
```

Listing 4: File READ statements in subroutines PLAD and SAL

```
      SUBROUTINE PLAD(MS,IR,ID,IEL,NAME,R,DB,K,TAU,RLT,FAKT,QF,T2R,
     > RO,RR,PRO,RNE,TE,TI,TDR,Q,DK,DV,RN,PN1,PN2,PT1,PT2,DER,DRO,VD)
C
C  PLAD CALCULATES PLASMA PARAMETERS AND TRANSPORT COEFFICIENTS FOR
C  THE DIFFUSION PROGRAM "STRAHL"
C-----------------------------------------------------------------------
C  AXIS DENSITY "RE0", ELECTON TEMPERATURE "TE0", ION TEMPERATURE "TI0"
C  RELATIVE EDGE DENSITY "PN1", EDGE TEMPERATURE "PT1", COEFFICIENTS "PN2,
C  PN3,PT2,PT3,PT4" (EXP. FIT POSSIBLE), PULSE NUMBER "NUM", TIME "ZEIT"
C
C  FIT COEFFICIENTS FOR RADIAL VARATION OF ANOMALOUS D "PD1,PD2,PD3,PD4"
C  FIT COEFFICIENTS FOR DRIFT "PV1,PV2,PV3,PV4" (NOT ALL USED)
C
      READ(2,*) RE0,TE0,TI0,PN1,PN2,PN3,PT1,PT2,PT3,PT4,NUM,ZEIT
      READ(2,*) PD1,PD2,PD3,PD4,PV1,PV2,PV3,PV4
C-----------------------------------------------------------------------
C  FACTOR FOR NEOCLASSICAL TRANSPORT COEFFICIENTS (CGS); BTOR=2 T,
C  HYDROGEN, WITHOUT TI**3/2 BUT TRANSFORMATION EV**1/2 -> ERG**1/2
C
      Q0=QF*2.84E-11
C
C-----------------------------------------------------------------------
      SUBROUTINE SAL(S,SINT,RION,AL,ALM,ADH,IR,LZ,RNE,TE,RR,PRO,DEE,
     >DE1,F1,DE2,F2,V0,X1,MS)
C
C  SAL CALCULATES IONISATION AND RECOMBINATION COEFFICIENTS S/ALPHA
C  INCLUDING CX RECOMBINATION. IT PROVIDES DATA FOR LINE EXCITATION
C  AND TOTAL RADIATION LOSSES IN THE DIFFUSON PROGRAM "STRAHL".
C
C  SAL READS PARAMETERS FROM THE ATOMIC PHYSICS DATA SETS. ADDITIONALLY,
C  IT REQUIRES IR, TE(R), NE(R), PRO(R), V0 FOR IONISATION LENGTH.
C  IT RETURNS LZ, S/ALPHA, DEE FOR LINE CALCULATIONS, RION, X1 FOR
C  RECOMBINATION, DE1,F1 AND DE2,F2 FOR TOTAL LINE RADIATION LOSSES.
C
C  F1,F2,G1 AND G2 ARE THE OSCILLATOR STRENGTHS AND GAUNT FACTORS
C  OF THE IONS (1 = ATOM ETC.). FOR RADIATION CALCULATIONS IN "STRAHL"
C  THEY ARE MULTIPLIED AND RETUREND IN F1/F2.
C
C  TO ACCOUNT FOR FORBIDDEN LINES THE GAUNT FACTORS ARE MULTIPLIED
C  BY RESPECTIVE WEIGHT FACTORS.
C  !!  NEGATIVE GAUNT FACTORS SIGNAL DELTA-N # 0 TRANSITIONS  !!
C-----------------------------------------------------------------------
C  CX-RECOMBINATION (ROUTINE CALLED SEPARATELY), FIXED N0 PROFILE.
C  VALUE FROM 40 KEV/AMU O+8 RECOMBINATION, SIMILAR FOR 1/2,1/3 ENERGY
C-----------------------------------------------------------------------
C  READ ATOMIC DATA (S1=WAVELENGTH OF DIAGNOSTIC LINE -  NOT USED)
C
      DO 50 I=1,LG
      READ(2,*,END=55)LZ,S1,DEE(I),DE1(I),F1(I),G1(I),DE2(I),F2(I),
     >G2(I)
      READ(2,*)A1(I),A2(I),B1(I),B2(I),C1(I),C2(I),X1(I),X2(I),Z1(I),
     >Z2(I),QZ(I),ZE(I)
50      CONTINUE
C
C-----------------------------------------------------------------------
```

Listing 5: Example of data set for nickel

```
Ni 2.0 101
```

```
  30,   3.0,   10,    0.,135.,  115.,    5.,    1.0,  0.0E8,   10.
1.0,    0.,    -1.,    .001,    10.,  -1.6E5,   .1,    4.0,  0.000
3.20E13,3200.,2000.,.01,2.01,0.56,100.0,1.99,1.43,0.,8622,13.0
  0.0,    0.0,   0.,    0.,        -1.,   0.,    0.,    0.
 1,   0.0,    0.0,   0.0,   0.0,   0.0,   0.0,   0.0,   0.0
 4.0,  2.4,   .4,   .94,   .6,  .17, 8.68,  10.,    2.,  8.,   3.,  17.
 2,    0.0,    0.0,   10.,    2.,    .8,   15.,    1.,   -.25
 0.,    3.,   0.,   .9,    0.,  .25, 18.2,  18.2,   0.,  9.,   3.,  16.
 3,    0.0,    0.0,   15.,    2.,    .8,   25.,    1.,   -.25
 3.8,  4.5,   .7,   .2,    .5,  .6,  35.2,  122.,   8.,  6.,   3.,  15.
 4,    0.0,    0.0,   20.,    2.,    .8,   30.,    1.,   -.25
 4.4,  4.5,   .4,   0.,    .6,  0.,  54.9,  146.,   7.,  6.,   3.,  14.
 5,    0.0,    0.0,   30.,    2.,    .8,   45.,    1.,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  75.5,  171.,   6.,  6.,   3.,  13.
 6,    0.0,    0.0,   40.,    2.,    .8,   60.,    1.,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  108.,  196.,   5.,  6.,   3.,  12.
 7,    0.0,    0.0,   50.,    2.,    .8,   70.,    1.,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  133.,  221.,   4.,  6.,   3.,  11.
 8,  292.0,   40.0,   60.,    2.,    .8,   100.,    1.,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  162.,  246.,   3.,  6.,   3.,  10.
 9,    0.0,    0.0,   70.,    2.,    .8,   121.,   1.1,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  193.,  271.,   2.,  6.,   3.,   9.
10,    0.0,    0.0,   85.,    2.,    .8,   137.,   1.3,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  225.,  296.,   1.,  6.,   3.,   8.
11,    0.0,    0.0,   84.,   2.3,    .8,   177.,   1.1,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  321.,  363.,   6.,  2.,   3.,   7.
12,    0.0,    0.0,   78.,   2.3,    .8,   165.,   0.8,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  352.,  393.,   5.,  2.,   3.,   6.
13,    0.0,    0.0,   56.,   1.1,    .8,   172.,   0.4,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  384.,  423.,   4.,  2.,   3.,   5.
14,    0.0,    0.0,   69.,   1.7,    .8,   188.,   0.3,   -.25
 4.5,  4.5,    0.,  0.,    0.,  0.,  430.,  458.,   3.,  2.,   3.,   4.
15,    0.0,    0.0,   62.,   2.0,    .8,   200.,   0.4,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  464.,  494.,   2.,  2.,   3.,   3.
16,    0.0,    0.0,   58.,   1.1,    .8,   250.,   0.4,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  499.,  531.,   1.,  2.,   3.,   2.
17,  249.2,   49.7,   50.,   .77,    .8,   288.,   0.45,  -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  571.,  1458.,  2.,  6.,   3.,   1.
18,  292.0,   42.5,   41.,   .37,    .8,   302.,   0.3,   -.25
 7.0,  4.5,   0.,   0.,    0.,  0.,  608.,  1500.,  1.,  6.,   3.,   0.
19,  15.00,   820.,   997.,   3.5,   -.25,  1233.,   1.1,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  1546.,  1694.,  6.,  2.,   2.,   7.
20,   0.0,    0.0,   149.,   .06,    .8,   1042.,   1.3,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  1648.,  1775.,  5.,  2.,   2.,   6.
21,  190.,   65.30,   129.,   .09,    .8,   1033.,   1.6,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  1756.,  1858.,  4.,  2.,   2.,   5.
22,  118.,   107.,   112.,   .11,    .8,   1100.,   1.6,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  1894.,  1938,   3.,  2.,   2.,   4.
23,  159.,   78.03,   119.,   .16,    .8,   1127.,   1.4,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  2011.,  2056.,  2.,  2.,   2.,   3.
24,  118.,   107.,   108.,   .21,    .8,   1200.,   1.0,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  2131.,  2178.,  1.,  2.,   2.,   2.
25,  118.,   107.,   105.,   .15,    .8,   1327.,   .87,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  2295.,  9914.,  2.,  2.,   2.,   1.
26,  165.,    75.,   67.7,   .07,    .8,   1367.,   .56,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,  2399.,10020.,  1.,  2.,   2.,   0.
27,  1.00,   7800.,   7804.,   .77,   -.80,   9170.,   .24,   -.25
 4.5,  4.5,   0.,   0.,    0.,  0.,10290.,10288.,  2.,  0.,   1.,   1.
28,   0.0,   8000.,   8049.,   .42,   -.42,   9578.,   .15,   -.35
 4.5,  4.5,   0.,   0.,    0.,  0.,10775.,10775.,  1.,  0.,   1.,   0.
```

```
      PROGRAM STRAHL
C
C  DIFFUSION PROGRAM FOR ANOMALOUS AND NEOCLASSICAL TRANSPORT
C
C  THE DENSITIES OF THE ACTUAL AND THE PREVIOUS TIME STEP ARE STORED IN
C  RN AND RA, RESPECTIVELY. IF THERE ARE LZ IONISATION STAGES THERE ARE
C  LZ+1=NN SPECIES. BECAUSE OF THE SYSTEM, THE INDEX LZ+2=NP MUST ALSO
C  EXIST, BUT IS ALWAYS MULTIPLIED BY 0. THE PROTON DENSITY, IE NE-Z*NIMP,
C  IS STORED IN RN(NP,X); THE SPACIAL DERIVATIVE OF THE PROTON DENSITY
C  IS STORED IN LZ+3=NO. BOTH ARE NEEDED FOR NEOCLASSICAL COEFFICIENTS.
C
C  FOR OUTPUT THE LINE EMISSION (USING THE EXCITATION ENERGIES FROM THE
C  DATA SET) IS STORED IN RA(X,X).  LINES SHOULD BE NORMALIZED.
C
C  IN RA(NP,X) THE SUM OF ALL IONISATION STAGES IS KEPT. RA(NO,X)
C  SERVES FOR THE CALCULATION OF THE RADIATION LOSSES.
C
C  FOR TIME PLOT LINE INTEGRALS ARE STORED IN SUM(X,M), M REFERRING TO
C  THE RESPECTIVE TIME IN SUM(NO,M). TOTAL RADIATION IS IN SUM(NP,M)
C
      PARAMETER (PI=3.14159265, TINC=1.2, MS=31)
      DIMENSION RN(31,101),RA(31,101),S(29,101),A(101),B(101),C(101),
     > D(101),RNE(101),DK(101),BET(101),GAM(101),RR(101),RO(101),
     > PRO(101),TE(101),TI(101),TDR(101),DEE(29),DE1(29),DE2(29),
     > X1(29),F1(29),F2(29),X(201),Y(201),Q(101),AL(29,101),SINT(101),
     > SUM(31,100),DV(101),YM(30),E(101),F(101),ICL(7),VD(101)
      CHARACTER IDA*18,HEAD*20,NAME*16,IEL*2,INPAR(5)*12
      CHARACTER*1 ANT,IY,IEND
      REAL K
C
      DATA RN/3131*0./,RA/3131*0./,S/2929*0./,AL/2929*0./,
     > SUM/3100*0./,DV/101*0./,YM/30*1.E-30/,E/101*0./,SMAX/1.E-30/,
     > M/0/,T00/0./,T01/0./,IZU/1/,TVE/0./,DVE/0./,DVA/0./,
     > TV2/0./,DV2/0./,DA2/0./,REJ/0./,SOLD/0./,GERC/0./
      DATA IY/'C'/,IEND/'E'/,HEAD/'  plot#              '/,ANT/' '/
C
C  ARRAY FIRST INDEX FOR SUBROUTINES = MS
C
      IZO=MS
C
C  DATE
C
2000  CALL CLOCK(ICL)
      WRITE(IDA,2) ICL(5),ICL(6),ICL(7)
2     FORMAT(I2,'/',I2,'/',I4)
      IDA(13:18)='STRAHL'
C
      WRITE(1,1308) T00
1308  FORMAT(' Program STRAHL (27.08.1987 K. Behringer), start time='
     > ,F8.4,'s')
2020  ND=4
      CALL RDPAR('file=?output control(default=-2)?go to t=?(0)'
     > //',E-limit(eV)=?(0)',INPAR,ND,1,1)
      IF(ND .EQ. 0) GO TO 2020
      READ(INPAR(2),'(I12)',ERR=2020) IPL
      READ(INPAR(1),'(A12)',ERR=2020) NAME
      READ(INPAR(3),'(F12.0)',ERR=2020) T01
      READ(INPAR(4),'(F12.0)',ERR=2020) ELI
      IF(INPAR(2)(12:12) .EQ. ' ') IPL=-2
      IF(IPL .LT. 0 .AND. T01 .GT. 0.) IPL=ABS(IPL)
```

```
C
C  IPL=1:PLOT, IPL=2:NORMALIZED PLOT, -1/-2:LAST PLOT ONLY, IPL=0:PRINT
C
      OPEN(UNIT=2,FILE=NAME,STATUS='OLD',ACCESS='SEQUENTIAL',ERR=2020)
C
      READ(2,1) IEL,K,IR
1     FORMAT(A2,F4.0,I4)
C
C  IMPURITY ELEMENT "IEL", COORDINATE R**"K", NO. OF GRIDPOINTS "IR"
C
C  NUMBER OF TIME STEPS "NTE", SIMULATED TIME "RTL" (S), STEPS BETWEEN
C  PLOTS "ITZ", XMIN (FOR PLOT), POSITION OF IMPURITY SOURCE "RL",
C  PLASMARADIUS "R", SCRAPE-OFF WIDTH "DB",RECOMBIN. FUDGE FACT."ALM"
C  NEUTRAL DENSITY FOR CX-RECOMBINATION "ADH", CX TIME "T1"
C
C  D-MULTIPLIER "FAKT", NEOCL.-FACT."QF", NORMALIZED DRIFT VELOCITY "VD",
C  IMPURITY LEVEL (FOR PROTON DENSITY/NEOCL. TRANSP.) "VGR", IMPURITY
C  PULSE LENGTH "T0",-VELOCITY "V0"(NEGATIVE), "TAU" PARALLEL, TE FALL-
C  OFF LENGTH "RLT", RECYCLING FACTOR "RCL"
C
      READ(2,*) NTE,RTL,ITZ,XMIN,RL,R,DB,ALM,ADH,T1
      READ(2,*) FAKT,QF,VD(1),VGR,T0,V0,TAU,RLT,RCL
C
C  PARAMETERS; GENERALLY: LENGTHS IN CM, TIME IN S.
C  CALCULATE DELTA-T FROM SIMULATED TIME AND NUMBER OF STEPS:
C
      DET=RTL/((TINC**NTE-1.)*FLOAT(ITZ)/(TINC-1.)-1.)
      DETA=DET
C
C  IF EVENT, DELTA-T IS RESET. DISTRIBUTE NTE INTO TIME INTERVALS
C
      FF=T0
      IF(T1 .NE. 0 .AND. T1 .LT. T0) FF=T1
      IF(FF .GE. RTL) GO TO 7654
      N=INT(0.5*(FLOAT(NTE)+1.0-ALOG((RTL-FF)/FF)/ALOG(TINC)))
      IF(N .LT. 1) N=1
      DET=FF/((TINC**N-1.)*FLOAT(ITZ)/(TINC-1.)-1.)
      N=NTE-N
      DETA=(RTL-FF)/((TINC**N-1.)*FLOAT(ITZ)/(TINC-1.))
      PRINT *, N,DET,DETA
7654  NTE=NTE*ITZ-1
      IF(IR .GT. 101) IR=101
      II=IR-1
      RX=R+DB
C
C  INITIAL VALUES AND VALUES FOR PROGRAM REENTRY
C
      T=T00
      T0=T0+T00
      T1=T1+T00
      S(1,1)=0.
C
C  CALCULATION OF PLASMA DATA IN ROUTINE "PLAD"
C
      CALL PLAD(MS,IR,ID,IEL,NAME,R,DB,K,TAU,RLT,FAKT,QF,T2R,RO,
     > RR,PRO,RNE,TE,TI,TDR,Q,DK,DV,RN,PN1,PN2,PT1,PT2,DER,DRO,VD)
C
      IJ=ID-1
      V=DK(IR)
C
```

```
C  CALCULATION OF ION.- UND REC.-COEFFICIENTS IN "SAL"
C
       I=MS-2
       CALL SAL(S,SINT,RION,AL,ALM,ADH,IR,LZ,RNE,TE,RR,PRO,DEE,DE1,
     > F1,DE2,F2,V0,X1,I)
C
       CLOSE(UNIT=2)
C
       NN=LZ+1
       NP=LZ+2
       NO=LZ+3
C
C  PROTON DENSITY AND DERIVATIVE INTO CORRECT ARRAYS
C
       DO 100 I=1,IR
       RN(NP,I)=RN(MS-1,I)
100    RN(NO,I)=RN(MS,I)
C
C  CALCULATION OF THE SOURCE AND NORMALIZATION TO N=1
C
       I1=INT(RL**K/DRO+1.50001)
       IF(I1 .GT. IR) I1=IR
       GEZA=0.
       DO 375 I=2,I1
375    GEZA=GEZA+SINT(I)*RNE(I)*S(1,I)*RR(I)/PRO(I)
       RN0=DK(IJ)*2.*RX/(RION*GEZA)
C
C  GEZA=NUMBER OF PARTICLES ENTERING PLASMA PER SECOND AND CM LENGTH
C
       GEZA=PI*RN0*GEZA
C
       GRO=K/RX*(FLOAT(IR)-1.5)/V
       GF=0.5*PI*RX*V
C
C  OUTPUT OF PLASMA DATA IN ROUTINE "DATO" (NOT USED AT PRESENT)
C
       IF(IPL .NE. 0) GO TO 151
C
       OPEN(UNIT=4,FILE='SAL:DATA',STATUS='UNKNOWN',ACCESS='SEQUENTIAL')
C
*      CALL DATO(T,LZ,IR,IDA,NAME,RR,RNE,TE,TI,TDR,DK,Q,E,RN,RA,
*    > SUM,S,AL,MS,GEZA,TSU,ALM)
C
C ****************************TIME LOOP**********************************
C
151    DO 300 N=1,NTE
C
C  EVENT AFTER NEXT TIMESTEP? (PULSE SWITCHED OFF, CX)
C
       FF=T+DET+DETA
       IF(T .LT. T0 .AND. FF .GT. T0) DET=T0-T
       IF(T .LT. T1 .AND. FF .GT. T1) DET=T1-T
C
       I=MS-2
       IF(T .EQ. T1) CALL
     > SAL(S,SINT,RION,AL,ALM,ADH,IR,LZ,RNE,TE,RR,PRO,DEE,DE1,F1,
     > DE2,F2,V0,X1,I)
C
       IF(T .EQ. T0) DET=DETA
       IF(T .EQ. T1 .AND. N .NE. 1) DET=DETA
```

```
      T=T+DET
      CON=1.+REJ
      IF(T .GT. T0) CON=REJ
      IF(N .EQ. NTE) IPL=IABS(IPL)
C
C  STORE LAST TIME STEP
C
      DO 425 I=1,IR
      DO 450 J=1,NO
450   RA(J,I)=RN(J,I)
425   CONTINUE
C
C  LOOP OF IONISATION STAGES, 1. NEUTRALS
C
C  ITERATION AT FIXED TIME (NO LONGER USED, 3.11.81)
C
C370  DO 1000 L=1,2
C
      DO 350 I=1,I1
350   RN(1,I)=RN0*SINT(I)*CON
C
      DO 380 I=1,IR
380   F(I)=DK(I)+Q(I)*(RN(NP,I)+RA(NP,I))
C
C  NOW OTHER IONISATION STAGES
C
      DO 700 LV=2,NN
C
C  ALTERNATING UP AND DOWN
C
      NZ=LV
      IAW=MOD(N,2)+1
      IF(IAW .EQ. 1) NZ=NN-LV+2
C
      ZZ=NZ-1
      DO 510 I=2,IR
510   E(I)=Q(I)*(ZZ*(RN(NO,I)+RA(NO,I))-TDR(I)/TI(I)*(1.+0.5*ZZ)*
     > (RN(NP,I)+RA(NP,I)))+VD(I)*2.*DK(I)/RX**2*RR(I)
C
      DO 520 I=ID,IR
      F(I)=F(IJ)
520   E(I)=E(IJ)
C
C  COEFFICIENTS AT R=0
C
      C(1)=-DET*F(1)*PRO(1)
      B(1)=1.-C(1)+DET/2.*
     > (Q(1)*(ZZ*(RN(NO,2)+RA(NO,2))/DER-(1.+.5*ZZ)*T2R/TI(1)*
     > (RN(NP,1)+RA(NP,1)))+VD(1)*2.*DK(1)/RX**2)
      D(1)=RA(NZ,1)*(2.-B(1))-RA(NZ,2)*C(1)
C
      GO TO (732,731) IAW
C
731   B(1)=B(1)+DET*RNE(1)*S(NZ,1)
      D(1)=D(1)-DET*RNE(1)*(RA(NZ,1)*AL(NZ-1,1)-RA(NZ+1,1)*AL(NZ,1)-
     > RN(NZ-1,1)*S(NZ-1,1))
      GO TO 740
C
732   B(1)=B(1)+DET*RNE(1)*AL(NZ-1,1)
      D(1)=D(1)-DET*RNE(1)*(RA(NZ,1)*S(NZ,1)-RA(NZ-1,1)*S(NZ-1,1)-
```

```
     > RN(NZ+1,1)*AL(NZ,1))
C
C  COEFFICIENTS AT R=R OR AT R+DB
C
740     A(IR)=1.-.5*(E(IR-1)+E(IR))/V-GRO*(F(IR-1)+F(IR))
        B(IR)=1.-.5*(E(IR-1)+E(IR))/V+GRO*(F(IR-1)+F(IR))
        D(IR)=-RA(NZ,IR-1)*A(IR)-RA(NZ,IR)*B(IR)
C
C  NORMAL COEFFICIENTS
C
        DO 400 I=2,II
        VDA=DET*PRO(I)**2
        VDR=4.*VDA*F(I)
        VV=DET/2.*(DV(I)+PRO(I)*(E(I+1)-E(I-1))+E(I)/RR(I))
        A(I)=VDA*(F(I)*(DRO/RO(I)-2.)+.5*(F(I+1)-F(I-1)-E(I)/PRO(I)))
        B(I)=1.+VDR+VV
        C(I)=-VDR-A(I)
        D(I)=-RA(NZ,I-1)*A(I)+RA(NZ,I)*(2.-B(I))-RA(NZ,I+1)*C(I)
C
        GO TO (721,711) IAW
C
711     B(I)=B(I)+DET*RNE(I)*S(NZ,I)
        D(I)=D(I)-DET*RNE(I)*(RA(NZ,I)*AL(NZ-1,I)-RA(NZ+1,I)*AL(NZ,I)-
     > RN(NZ-1,I)*S(NZ-1,I))
        GO TO 400
C
721     B(I)=B(I)+DET*RNE(I)*AL(NZ-1,I)
        D(I)=D(I)-DET*RNE(I)*(RA(NZ,I)*S(NZ,I)-RA(NZ-1,I)*S(NZ-1,I)-
     > RN(NZ+1,I)*AL(NZ,I))
C
400     CONTINUE
C
        BET(1)=B(1)
        GAM(1)=D(1)/B(1)
C
        DO 500 I=2,IR
        BET(I)=B(I)-(A(I)*C(I-1))/BET(I-1)
500     GAM(I)=(D(I)-A(I)*GAM(I-1))/BET(I)
C
        RN(NZ,IR)=GAM(IR)
C
        DO 600 I=2,IR
        J=IR+1-I
600     RN(NZ,J)=GAM(J)-(C(J)*RN(NZ,J+1))/BET(J)
C
C  END OF IONISATION STAGES
C
700     CONTINUE
        IF(QF .EQ. 0.) GO TO 355
C
C  CALCULATION OF PROTON DENSITY (ONLY IF NEOCL. COEFF. REQUESTED)
C
        DO 410 I=1,IR
        RN(NP,I)=RNE(I)-VGR*RN(2,I)*RNE(1)
        DO 420 NZ=3,NN
        ZZ=FLOAT(NZ-1)*VGR*RNE(1)
420     RN(NP,I)=RN(NP,I)-ZZ*RN(NZ,I)
410     IF(RN(NP,I) .LT. 0.) RN(NP,I)=0.
C
C  DERIVATIVE WITH RESPECT TO R
```

```
C
      DO 430 I=2,II
430   RN(NO,I)=PRO(I)*(RN(NP,I+1)-RN(NP,I-1))
C
C  TIME INTEGRATED INFLUX
C
355   GERC=GERC+DET*GEZA*CON
C
C  PARTICLES LOST AT PERIPHERY (NOT PARALLEL) / SECOND
C  PARTIAL BALANCE POSSIBLE IN TS2, TV2, DS2, DV2, DA2
C
      TSU=0.
      DO 575 NZ=2,NN
      TSU=TSU+GF*(RN(NZ,IR)+RA(NZ,IR)+RN(NZ,II)+RA(NZ,II))
*     IF(NZ .EQ. IGR) TS2=TSU
575   CONTINUE
C
C  RECYCLING WITHOUT PARALLEL LOSSES
C
*     REJ=RCL*TSU/GEZA
C
C  END OF ITERATION
C
C1000  CONTINUE
C
C  TIME INTEGRATED LOSSES AT PERIPHERY
C
      TVE=TVE+DET*TSU
*     TV2=TV2+DET*TS2
C
C  PARALLEL LOSSES / SECOND (*PI LATER)
C
      DSU=0.
      DO 580 NZ=2,NN
      DO 585 I=ID,II
585   DSU=DSU+RN(NZ,I)*DV(I)*RR(I)/PRO(I)
*     IF(NZ .EQ. IGR) DS2=DSU
580   CONTINUE
C
C  RECYCLING OF TOTAL LOSSES
C
      REJ=RCL*(TSU+(DVA+DSU)*PI/2.)/GEZA
C
C  TIME INTEGRATED, NO BACKSTREAMING FROM DIVERTOR
C
      DVE=DVE+(DSU+DVA)*DET/2.*PI
*     DV2=DV2+(DS2+DA2)*DET/2.*PI
C
C  BYPASS (TO BE MODIFIED)
C
C     DVE=DVE*(1.-10.*DET)+(1.-5.*DET)*(DSU+DVA)*DET/2.*PI
      DVA=DSU
*     DA2=DS2
C
C  OUTPUT AFTER IONISATION (ODD) TIMESTEP
C
590   IAW=MOD(N+1,ITZ)
      IF(IAW .NE. 0 .AND. T .NE. T0 .AND. T .NE. T1) GO TO 300
C
C  INCREASE TIMESTEP FOR NEXT RUN
```

```
C
      IF(IAW .EQ. 0) DET=TINC*DET
C
C  PREPARATION OF RA(NP), C AND RA(NO) FOR SUMS
C
      DO 475 I=1,IR
      RA(NP,I)=0.
      C(I)=0.
475   RA(NO,I)=0.
C
C  INTEGRATION OF PARTICLES IN PLASMA FOR BALANCE
C
      M=M+1
      SUM(NO,M)=T
      DO 625 NZ=2,NN
      SUM(NZ,M)=0.125*(RN(NZ,IR)*RR(IR)/PRO(IR)
     > -RN(NZ,II)*RR(II)/PRO(II))
C
C  IN THE TOTAL VOLUME (INCLUDING SCRAPE-OFF)
C
      DO 675 I=2,II
C
C  OR ONLY IN THE CONFINED PLASMA (CHANGE INDICES ABOVE)
C
C     DO 675 I=2,IJ
675   SUM(NZ,M)=SUM(NZ,M)+RN(NZ,I)*RR(I)/PRO(I)
      SUM(NZ,M)=PI*SUM(NZ,M)
C
625   SUM(1,M)=SUM(1,M)+SUM(NZ,M)
C
C  PARTICLE CONFINEMENT TIME (MS); KEEP OLD SUM
C
      EIN=500.*(SUM(1,M)+SOLD)/(GEZA*CON-(SUM(1,M)-SOLD)
     > /(T-SUM(NO,M-1)))
      SOLD=SUM(1,M)
C
C  SUM(1,TIME) IS USED FOR THE INTEGRAL BALANCE (OUGHT TO BE 1.0)
C
      SUM(1,M)=(SUM(1,M)+TVE+DVE)/GERC
C
C  EXCITATION, INTEGRATION UND NORMALIZATION (NOT ATOMS)
C
      RMAX=1.E-30
      DMAX=1.E-30
C ---------------------------------------------------------------------
      DO 1100 NZ=2,NN
      FF=1.E-30
      DO 1200 I=1,IR
      RA(NZ,I)=RN(NZ,I)/SQRT(TE(I))*
     > 1.34E-12*RNE(I)*EXP(-DEE(NZ)/TE(I))
C
C  SUM OF ALL IONSISATION STAGES, ZEFF; NORMALIZE DENSITY PLOT
C
      RA(NP,I)=RA(NP,I)+RN(NZ,I)
      IF(RA(NP,I).GT.DMAX) DMAX=RA(NP,I)
      C(I)=C(I)+VGR*RN(NZ,I)*FLOAT((NZ-1)*(NZ-2))*RNE(1)/RNE(I)
C
C  CALCULATION OF RADIATION LOSSES; 1. LINE RADIATION
C
      S1=F1(NZ)*EXP(-DE1(NZ)/TE(I))
```

```
      S2=F2(NZ)*EXP(-DE2(NZ)/TE(I))
      S3=2.6E-11*RNE(I)*RN(NZ,I)/SQRT(TE(I))
C
C  2. RECOMBINATION CONTINUA
C
      S4=1.6E-6*X1(NZ)*RNE(I)*RN(NZ+1,I)*AL(NZ,I)
C
C  3. BREMSSTRAHLUNG
C
      S4=S4+1.69E-19*RNE(I)*SQRT(TE(I))*RN(NZ,I)*FLOAT((NZ-1)*(NZ-1))
C
C  ONLY RADIATION ABOVE ENERGY LIMIT; FIND MAXIMUM RADIATION
C
      IF(DE1(NZ) .GT. ABS(ELI)) RA(NO,I)=RA(NO,I)+S3*S1
      IF(DE2(NZ) .GT. ABS(ELI)) RA(NO,I)=RA(NO,I)+S3*S2
      IF(ELI .GE. 0) RA(NO,I)=RA(NO,I)+S4
      IF(RA(NO,I).GT.RMAX) RMAX=RA(NO,I)
C
      IF(NZ .NE. LZ) GO TO 1200
      RA(1,I)=RN(1,I)/(RN0*SINT(I1)*CON+1.E-30)
C
C  RECOMBINATION INTO EXCITED HYDROGEN-LIKE LEVELS
C
      XFH=(X1(LZ)-DEE(LZ))/TE(I)
      RKK=5.2E-14*FLOAT(LZ-1)*SQRT(XFH)*(XFH**2+2.335*XFH+.251)/
     > (XFH**2+3.331*XFH+1.682)
      RA(LZ,I)=RA(LZ,I)+RKK*RN(NN,I)*RNE(I)*1.E-6*DEE(LZ)
C
1200  IF(RA(NZ,I) .GT. FF) FF=RA(NZ,I)
C
C  CALCULATE LINE-OF-SIGHT INTEGRALS
C
      SUM(NZ,M)=DER/2.*(RA(NZ,1)+RA(NZ,2))-.25*RA(NZ,2)/PRO(2)
     > +0.0625*(RA(NZ,IR)/PRO(IR)-RA(NZ,II)/PRO(II))
C
      DO 1050 I=2,II
1050  SUM(NZ,M)=SUM(NZ,M)+.5*RA(NZ,I)/PRO(I)
C
C  YM = MAX. OF EACH LINE INTEGRAL; SMAX = ABSOLUTE MAXIMUM
C
      IF(SUM(NZ,M).GT.YM(NZ)) YM(NZ)=SUM(NZ,M)
      IF(YM(NZ) .GT. SMAX) SMAX=YM(NZ)
C
      IF(IPL .EQ. 1) GO TO 1100
C
      DO 1300 I=1,IR
1300  RA(NZ,I)=RA(NZ,I)/FF
C
      IF(RMAX .LT. 1.2) RMAX=1.2
1100  CONTINUE
C-----------------------------------------------------------------------
C
C  TOTAL RADIATION LOSSES INTO SUM(NP,M)
C
      SUM(NP,M)=0.125*(RA(NO,IR)*RR(IR)/PRO(IR)
     > -RA(NO,II)*RR(II)/PRO(II))
      DO 1400 I=2,II
1400  SUM(NP,M)=SUM(NP,M)+RA(NO,I)*RR(I)/PRO(I)
C
      SUM(NP,M)=PI*SUM(NP,M)
```

```
*       SUM(2,M)=(TV2+DV2)/(TVE+DVE)
        IF(SUM(NP,M) .GT. SMAX) SMAX=SUM(NP,M)
        YM(NP)=SMAX
C
        WRITE(HEAD(9:),9) M,ITZ,K
9       FORMAT(I3,'/',I3,'/',F4.1)
        IF(IPL .NE. 0) GO TO 750
C
C  OUTPUT OF CALCULATION IN ROUTINE "DATO"
C
        IF(N .NE. NTE) GO TO 300
        CALL DATO(T,LZ,IR,IDA,NAME,RR,RNE,TE,TI,TDR,DK,Q,E,RN,RA,
     > SUM,S,AL,MS,GEZA,TSU,ALM)
        GO TO 300
C
C  PLOT OUTPUT+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
750     IF(IPL .LT. 0 .AND. N .NE. NTE) GO TO 300
        IF(T .LT. T01 .AND. N .NE. NTE) GO TO 300
        MR=IR
        X(1)=0.
        Y(1)=0.
        IF(IPL .GT. 0) GO TO 760
C
C  PLOTPARAMETERS FOR TIME PLOT FRAME, YMAX ADJUSTABLE BY "SYM"
C
        MR=M
        XMIN=T01
        XMAX=T
        YMIN=0.
        YMAX=1.
        IF(SYM .GT. 0) YMAX=SYM
        IO=NP
        CALL CRT011(0,IDA//HEAD,'time [s]',NAME,XMIN,XMAX,YMIN,YMAX,
     > 1,X,Y)
        GO TO 745
C
C  PLOTPARAMETERS FOR RADIAL PLOT FRAME; "D"= PLOT DENSITIES
C
760     XMIN=0.
        XMAX=RX
        YMIN=0.
        IF(ANT .NE. 'D') YMAX=RMAX
        IF(ANT .EQ. 'D') YMAX=DMAX
        IF(SYM .GT. 0.) YMAX=SYM
        IO=NO
        CALL CRT011(0,IDA//HEAD,'radius [cm]',NAME,XMIN,XMAX,YMIN,YMAX,
     > 1,X,Y)
C
C  PLOT SIGNALS INTO FRAME; TIME PLOT; LOG PLOT (2 DECADES) POSSIBLE
C
745     DO 800 I=2,IO,2
        IF(IPL .GT. 0) GO TO 855
        DO 860 J=1,MR
        X(J)=SUM(NO,J)
        Y(J)=SUM(I,J)/YM(I)
        IF(IPL .EQ. -1) Y(J)=SUM(I,J)/SMAX
        IF(Y(J) .LE. 0) Y(J)=1.E-30
860     IF(ANT .EQ. 'L') Y(J)=0.5*ALOG10(Y(J))+1.
        GO TO 720
```

```
C
C  RADIAL PLOT; LOG FOR INTENSITIES; TOTAL DENSITY = RA(NO,J) ZEFF=C(J)
C
855     DO 850 J=1,IR
        X(J)=RR(J)
        Y(J)=RA(I,J)
        IF(ANT .EQ. 'D' .AND. I .NE. 1 .AND. I .LT. NP) Y(J)=RN(I,J)
        IF(ANT .EQ. 'D' .AND. I .EQ. NO) Y(J)=C(J)
        IF(Y(J) .LE. 0) Y(J)=1.E-30
850     IF(ANT .EQ. 'L') Y(J)=0.5*ALOG10(Y(J))+1.
720     JO=MR
        IF(I .LT. IZU .OR. I .GT. IZO) GO TO 800
        CALL INTP(X,Y,JO,101)
        CALL CRT011(2,' ',' ',' ',XMIN,XMAX,YMIN,YMAX,JO,X,Y)
800     CONTINUE
C
C  DASHED LINES FOR ODD IONISATION STAGES; TIME PLOT, RADIAL PLOT
C
        DO 900 I=1,IO,2
        IF(IPL .GT. 0) GO TO 955
        DO 960 J=1,MR
        X(J)=SUM(NO,J)
        IF(I .EQ. 1) GO TO 965
        Y(J)=SUM(I,J)/YM(I)
        IF(IPL .EQ. -1) Y(J)=SUM(I,J)/SMAX
        IF(Y(J) .LE. 0.) Y(J)=1.E-30
        IF(ANT .EQ. 'L') Y(J)=0.5*ALOG10(Y(J))+1.
        GO TO 960
965     Y(J)=.8*SUM(1,J)
960     CONTINUE
        GO TO 770
C
955     DO 950 J=1,IR
        X(J)=RR(J)
        Y(J)=RA(I,J)
        IF(ANT .EQ. 'D' .AND. I .NE. 1 .AND. I .LT. NP) Y(J)=RN(I,J)
        IF(ANT .EQ. 'D' .AND. I .EQ. NO) Y(J)=C(J)
        IF(Y(J) .LE. 0.) Y(J)=1.E-30
950     IF(ANT .EQ. 'L') Y(J)=0.5*ALOG10(Y(J))+1.
770     JO=MR
        IF(I .LT. IZU .OR. I .GT. IZO) GO TO 900
        CALL INTP(X,Y,JO,101)
        CALL CRT011(42,' ',' ',' ',XMIN,XMAX,YMIN,YMAX,JO,X,Y)
900     CONTINUE
C
C  WRITE INFORMATION UNDER PLOTS; NO ONLY IF CX WAS SWITCHED ON
C
        FF=ADH
        IF(T .LT. T1) FF=0.
        IF(IPL .GT. 0 .AND. ANT .NE. 'D') WRITE(1,44) ANT,T,T0,RX,FF
44      FORMAT(1X,A1,' intensities+Prad,t=',F8.5,'s;t0=',
     > E8.2,'s;a=',F6.1,'cm;n0=',E8.2,'cm-3')
C
        IF(IPL .GT. 0 .AND. ANT .EQ. 'D') WRITE(1,45) ANT,T,T0,RX,FF
45      FORMAT(1X,A1,' densities and Zeff,t=',F8.5,'s;t0=',
     > E8.2,'s;a=',F6.1,'cm;n0=',E8.2,'cm-3')
C
        IF(IPL .LT. 0) WRITE(1,46) ANT,T,T0,RX,FF
46      FORMAT(1X,A1,' integrals+Ptot to t=',F8.5,'s;t0=',
     > E8.2,'s;a=',F6.1,'cm;n0=',E8.2,'cm-3')
```

```
C
      FF=SUM(NP,M)/(RA(NP,1)+1.E-30)
      WRITE(1,47) TE(1),RNE(1),F(1),VD(1),RION,DB,EIN,
     > RCL,FF,RA(NP,1),ELI
47    FORMAT(1X,'Te(0)=',F6.0,'eV;ne(0)=',E8.2,'cm-3;D(0)=',F6.0,
     > 'cm2/s;V0(0)=',F4.1,';ion=',F4.1,'cm',/,1X,'DB=',F3.0,
     > 'cm;c=',F5.1,'ms;RCL=',F4.2,';Prad=',
     > E8.2,'*E-13W/cm;n=',F6.2,';Elim=',F5.0,'eV')
C
C  NORMALIZED LINE-OF-SIGHT LINE INTEGRALS FOR IMPURITY MEASUREMENTS
C
      IF(ANT .EQ. 'D') GO TO 4949
      JU=IZU
      JO=JU+3
      DO 4848 J=JU,JO
      Y(J)=SUM(J,M)*2./(RA(NP,1)+1.E-30)
4848  IF(J .EQ. 1) Y(1)=SUM(1,M)
      WRITE(1,48) (J,DEE(J),Y(J),J=JU,JO)
48    FORMAT(' I*=',4(2X,I2,'-',F5.0,':',F6.3))
      GO TO 5050
C
4949  WRITE(1,49) TAU,RLT,DETA,QF,ALM,TI(1),V0
49    FORMAT(1X,'s:',F3.1,'/',F3.1,'cm;DET=',E8.2,'s;QF=',F3.1,
     > ';drc*',F4.1,';Ti(0)=',F6.0,'eV;v0=',E8.1,'cm/s')
c
5050  CALL WALPHA
      ND=5
      CALL RDPAR('Intens.,Dens.,Log.,Cont.,End, ?;t=?z1,z2=?,ymax=?',
     > INPAR,ND,1,24)
      ANT=INPAR(1)(1:1)
      READ(INPAR(2),'(F12.0)') T01
      IF(INPAR(3)(12:12) .NE. ' ') READ(INPAR(3),'(I12)') IZU
      IF(INPAR(4)(12:12) .NE. ' ') READ(INPAR(4),'(I12)') IZO
      IF(INPAR(5)(12:12) .NE. ' ') READ(INPAR(5),'(F12.0)') SYM
C
C  T01: GOTO TIME AND START OF TIME-PLOT; IZU,IZO: LOWER AND UPPER
C  IONISATION STAGES FOR PLOT AND IZU FOR I* PRINT; SYM: YMAX;
C  ANT: I=REPEAT INTENSITIES, D=DENSITIES, L=LOG, C=CONTINUE, E=END
C
      IF(ANT .EQ. IEND) GO TO 301
      IF(ANT .NE. IY .AND. ANT .NE. ' ') GO TO 750
      IF(ANT .EQ. IY .AND. N .NE. NTE) IPL=-IPL
      IF(ANT .EQ. IY .AND. N .EQ. NTE) GO TO 300
      IF(N .NE. NTE) GO TO 300
      IPL=-IPL
      GO TO 750
C
C*****************************END OF TIME LOOP*****************************
C
300   CONTINUE
      T00=T
C
C  IF ANT=CONTINUE, REENTER PROGRAM
C
      IF(ANT .EQ. IY) GO TO 2020
301   STOP
      END
```

```
      SUBROUTINE PLAD(MS,IR,ID,IEL,NAME,R,DB,K,TAU,RLT,FAKT,QF,T2R,
     > RO,RR,PRO,RNE,TE,TI,TDR,Q,DK,DV,RN,PN1,PN2,PT1,PT2,DER,DRO,VD)
C
C  PLAD CALCULATES PLASMA PARAMETERS AND TRANSPORT COEFFICIENTS FOR
C  THE DIFFUSION PROGRAM "STRAHL"
C
      DIMENSION RO(1),RR(1),PRO(1),RNE(1),TE(1),TI(1),
     > TDR(1),Q(1),DK(1),DV(1),RN(MS,1),VD(1)
      CHARACTER NAME*16,IEL*2
      REAL K
C
      NO=MS
      NP=MS-1
C
C  AXIS DENSITY "RE0", ELECTON TEMPERATURE "TE0", ION TEMPERATURE "TI0"
C  RELATIVE EDGE DENSITY "PN1", EDGE TEMPERATURE "PT1", COEFFICIENTS "PN2,
C  PN3,PT2,PT3,PT4" (EXP. FIT POSSIBLE), PULSE NUMBER "NUM", TIME "ZEIT"
C
C  FIT COEFFICIENTS FOR RADIAL VARATION OF ANOMALOUS D "PD1,PD2,PD3,PD4"
C  FIT COEFFICIENTS FOR DRIFT "PV1,PV2,PV3,PV4" (NOT ALL USED)
C
      READ(2,*) RE0,TE0,TI0,PN1,PN2,PN3,PT1,PT2,PT3,PT4,NUM,ZEIT
      READ(2,*) PD1,PD2,PD3,PD4,PV1,PV2,PV3,PV4
C
C  EXPONENTIAL FIT FOR TEMPERATURE?
C
      IFL=0
      IF(PT1.LT.0.0.OR.PT2.LT.0.0.OR.PT3.LT.0.0.OR.PT4.LT.0.0) IFL=1
C
      RX=R+DB
      DRO=(RX**K)/(FLOAT(IR)-1.5)
      DER=DRO**(1./K)
      VDN=VD(1)
      IF(PV3 .NE. 0.) VD(1)=PV1
      PRO(1)=2./DER**2
C
C  FACTOR FOR NEOCLASSICAL TRANSPORT COEFFICIENTS (CGS); BTOR=2 T,
C  HYDROGEN, WITHOUT TI**3/2 BUT TRANSFORMATION EV**1/2 -> ERG**1/2
C
      Q0=QF*2.84E-11
      Q(1)=Q0/SQRT(TI0)
C
      DO 100 I=1,IR
      RO(I)=(I-1)*DRO
      RR(I)=RO(I)**(1./K)
      IF(RR(I) .LE. R) ID=I+1
C
C  RADIAL DISTRIBUTIONS UP TO PLASMA RADIUS R
C
      IF(RR(I) .GT. R) GO TO 120
      PRBN=1.-(RR(I)/R)**PN2
C
C  SPECIAL FIT FOR HOLLOW ELECTRON DENSITY PROFILES (H-MODE)
C
      IF(RE0 .LT. 0) PRBN=PRBN*(1.-0.6*EXP(-(RR(I)/90.)**2))
C
C  PARABULA OR EXPONENTIAL FIT FOR TEMPERATURE
C
      IF(IFL.EQ.0) PRBT=1.-(RR(I)/R)**PT2
      IF(IFL.EQ.1) PRBT=EXP(PT1*RR(I)**2/R**2+
```

```
     > PT2*RR(I)**4/R**4+PT3*RR(I)**6/R**6+PT4*RR(I)**8/R**8)
C
       RNE(I)=ABS(RE0)*((1.-PN1)*PRBN**PN3+PN1)
C
C  DERIVATIVE OF ELECTRON = PROTON DENSITY
C
       RN(NO,I)=ABS(RE0)*(PN1-1.)*PN2*PN3/R*PRBN**(PN3-1.)*
     > (RR(I)/R)**(PN2-1.)
C
C  TEMPERATURE PROFILES (TE PROP. TI ASSUMED), HOLD AT TE=5 EV
C
       IF(IFL.EQ.0) TE(I)=(TE0-PT1)*PRBT**PT3+PT1
       IF(IFL.EQ.1) TE(I)=TE0*PRBT
       IF(TE(I) .LT. 5.) TE(I)=5.
       TI(I)=TI0/TE0*TE(I)
       RN(NP,I)=RNE(I)
C
C  DERIVATIVE OF ION TEMPERATURE ASSUMING TI PROP. TE
C
       IF(IFL.EQ.0) TDR(I)=TI0/TE0*(PT1-TE0)*PT2*PT3/R*PRBT**(PT3-1.)
     > *(RR(I)/R)**(PT2-1.)
       IF(IFL.EQ.1)
     > TDR(I)=TI(I)*(2.*PT1*RR(I)/R**2+4.*PT2*RR(I)**3/R**4
     > +6.*PT3*RR(I)**5/R**6+8.*PT4*RR(I)**7/R**8)
C
C  ANOMALOUS DIFFUSION COEFFICIENT IN UNITS OF 1 SQUARE M /S
C
C      DK(I)=FAKT*3.91E19*TE(I)/(312.5*RNE(I)*TE(I)+1.25E17)
       DK(I)= FAKT*10000.
       IF(PD1 .NE. 0.) DK(I)=DK(I)/(PD1+1.)*(1.+PD1*(RR(I)/R)**PD2)
C
       IF(I .EQ. 1) GO TO 100
C
C  Q**2 PROFILE; Q=1 ON AXIS, Q=4 ON BOUNDARY ASSUMED
C
       PRBN=1.-(RR(I)/R)**2
       Q(I)=Q0*16.*((RR(I)/R)**2/(1.-PRBN**4))**2/SQRT(TI(I))
C
C  DRIFT VELOCITY FIT IN UNITS OF 2*D*R/A**2
C
120    VD(I)=PV1
       IF(RR(I).GE.PV3) VD(I)=VDN*(1.-PV4*(RR(I)-PV3)**2/(RX-PV3)**2)
       IF(RR(I).LT.PV2 .OR. RR(I).GT.PV3) GO TO 125
       DPV=(PV3-PV2)**2
       C=-(VDN-PV1)/DPV**2
       B=(VDN-PV1)/DPV-C*DPV
       VD(I)=PV1+B*(RR(I)-PV2)**2+C*(RR(I)-PV2)**4
C
125    PRO(I)=.5*K*RO(I)/(RR(I)*DRO)
100    CONTINUE
       T2R=(TE(2)-TE(1))*2./DER**2
       IJ=ID-1
C
C  SCRAPE-OFF LAYER
C
       DO 75 I=ID,IR
       TE(I)=(TE(IJ)-5.)*EXP((RR(IJ)-RR(I))/RLT) + 5.
       TI(I)=TI(IJ)
       TDR(I)=0.
       Q(I)=Q(IJ)
```

```
      DK(I)=DK(IJ)
C     DV(I)=1262.*SQRT(TI(I)/LZ)
C
C  PARALLEL LOSSES CHARACTERISED BY TAU-PARALLEL IN MS
C
      DV(I)=1000./TAU
C     RNE(I)=RNE(IJ)*EXP((RR(IJ)-RR(I))/(.42*TI(I)**.25))
      RNE(I)=RNE(IJ)*EXP((RR(IJ)-RR(I))/SQRT(DK(I)/DV(I)))
      RN(NP,I)=RNE(I)
C     RN(NO,I)=-RNE(I)/(.42*TI(I)**.25)
75    RN(NO,I)=-RNE(I)/SQRT(DK(I)/DV(I))
C
C
C  REPLACE TAU BY FALL-OFF LENGTH
C
      TAU=SQRT(DK(ID)/DV(ID))
C
      WRITE(NAME,9797) IEL,NUM,ZEIT
9797  FORMAT(A2,',#',I5,'; ',F4.1,'s')
C
      RETURN
C
      END
```

```
      SUBROUTINE SAL(S,SINT,RION,AL,ALM,ADH,IR,LZ,RNE,TE,RR,PRO,DEE,
     >DE1,F1,DE2,F2,V0,X1,MS)
C
C  SAL CALCULATES IONISATION AND RECOMBINATION COEFFICIENTS S/ALPHA
C  INCLUDING CX RECOMBINATION. IT PROVIDES DATA FOR LINE EXCITATION
C  AND TOTAL RADIATION LOSSES IN THE DIFFUSON PROGRAM "STRAHL".
C
C  SAL READS PARAMETERS FROM THE ATOMIC PHYSICS DATA SETS. ADDITIONALLY,
C  IT REQUIRES IR, TE(R), NE(R), PRO(R), V0 FOR IONISATION LENGTH.
C  IT RETURNS LZ, S/ALPHA, DEE FOR LINE CALCULATIONS, RION, X1 FOR
C  RECOMBINATION, DE1,F1 AND DE2,F2 FOR TOTAL LINE RADIATION LOSSES.
C
C  F1,F2,G1 AND G2 ARE THE OSCILLATOR STRENGTHS AND GAUNT FACTORS
C  OF THE IONS (1 = ATOM ETC.). FOR RADIATION CALCULATIONS IN "STRAHL"
C  THEY ARE MULTIPLIED AND RETUREND IN F1/F2.
C
C  TO ACCOUNT FOR FORBIDDEN LINES THE GAUNT FACTORS ARE MULTIPLIED
C  BY RESPECTIVE WEIGHT FACTORS.
C  !!  NEGATIVE GAUNT FACTORS SIGNAL DELTA-N # 0 TRANSITIONS  !!
C
      DIMENSION S(MS,1),SINT(1),AL(MS,1),RNE(1),TE(1),PRO(1),
     >DEE(1),DE1(1),F1(1),DE2(1),F2(1),X1(1),RR(1)
      DIMENSION G1(30),G2(30),A1(30),A2(30),X2(30),B1(30),B2(30),
     >C1(30),C2(30),Z1(30),Z2(30),QZ(30),ZE(30)
C
C  CX-RECOMBINATION (ROUTINE CALLED SEPARATELY), FIXED N0 PROFILE.
C  VALUE FROM 40 KEV/AMU O+8 RECOMBINATION, SIMILAR FOR 1/2,1/3 ENERGY
C
      IF(S(1,1) .EQ. 0) GO TO 920
C
      DO 930 M=2,LZ
      FF=FLOAT(M)
      DO 940 I=1,IR
940   AL(M,I)=AL(M,I)+ADH/RNE(I)*1.1E-7*FF**1.07*
     >(0.5+(RR(I)/RR(IR)-0.3)**2)
930   CONTINUE
C
      RETURN
C
920   II=IR-1
      LG=MS-1
C
C  READ ATOMIC DATA (S1=WAVELENGTH OF DIAGNOSTIC LINE -  NOT USED)
C
      DO 50 I=1,LG
      READ(2,*,END=55)LZ,S1,DEE(I),DE1(I),F1(I),G1(I),DE2(I),F2(I),
     >G2(I)
      READ(2,*)A1(I),A2(I),B1(I),B2(I),C1(I),C2(I),X1(I),X2(I),Z1(I),
     >Z2(I),QZ(I),ZE(I)
50     CONTINUE
C
C  ZEROS FOR FULLY STRIPPED IONS
C
55    NN=LZ+1
      DEE(NN)=0.
      DE1(NN)=0.
      DE2(NN)=0.
      F1(NN)=0.
      F2(NN)=0.
C
```

```
C  CALCULATION OF IONISATION RATE COEFFICIENTS "S" AFTER "LOTZ"
C  COEFFICIENT "1" IS USED FOR IONISATION OF SPECIES "1"->"2" ETC.
C
      DO 150 M=1,LZ
      S(M,IR)=0
      DO 250 I=1,II
      Y1=X1(M)/TE(I)
      Y2=X2(M)/TE(I)
      S5=0.
      S6=0.
C
C  FIRST SUBSHELL
C
      IF(Y1 .GT. 100.) GO TO 245
      S1=EXP(-Y1)*(ALOG(1.+1./Y1)-.4/(1.+Y1)**2)
      S2=ALOG(1.+1./(Y1+C1(M)))-.4/(1.+C1(M)+Y1)**2
      S2=EXP(-Y1-C1(M))*S2
      S5=A1(M)*Z1(M)*(S1/Y1-B1(M)*EXP(C1(M))*S2/(Y1+C1(M)))
C
C  SECOND SUBSHELL
C
245   IF(Y2 .GT. 100.) GO TO 250
      S3=EXP(-Y2)*(ALOG(1.+1./Y2)-.4/(1.+Y2)**2)
      S4=ALOG(1.+1./(Y2+C2(M)))-.4/(1.+C2(M)+Y2)**2
      S4=EXP(-Y2-C2(M))*S4
      S6=A2(M)*Z2(M)*(S3/Y2-B2(M)*EXP(C2(M))*S4/(Y2+C2(M)))
250   S(M,I)=6.7E-7*(S5+S6)/TE(I)**1.5
150   CONTINUE
C
C  CALCULATION OF RECOMBINATION RATE COEFFICIENTS "AL"
C  COEFFICIENT "1" IS USED FOR RECOMBINATION OF SPECIES "2"->"1",
C  IE THE INDEX IS THE CHARGE OF THE RECOMBINING ION.
C  RECOMBINATION TO NEUTRAL ATOMS SET TO ZERO (23.1.86).
C
C  1. RADIATIVE RECOMBINATION
C
      DO 225 M=2,LZ
      FF=FLOAT(M)
      S2=FF**2*(2.*QZ(M)**2-ZE(M))/QZ(M)**3
      S3=126.*FF**(12./17.)/13.6**(1./17.)
C
      DO 275 I=1,IR
      Y1=X1(M)/TE(I)
      S1=SQRT(13.6/TE(I))*S2*Y1*(ALOG(1.+1./Y1)-.4/(1.+Y1)**2)
C
C  AFTER VALENCE SHELL(QUANT. NUMBER QZ, 2*QZ**2-ZE EMPTY PLACES, ZE>=0)
C  SUM UP ALL HIGHER (HYDROGEN-LIKE) SHELLS BETWEEN QZ+1 AND COLL. LIMIT
C
      JU=INT(QZ(M)+1.01)
C
C  UPPER "N" LIMIT: COLLISION LIMIT; STOP AT QZ+20
C
      JO=INT(S3*TE(I)**(1./17.)/RNE(I)**(2./17.)+1.)
      J=JU+20
      IF(JO .GT. J) JO=J
C
      S4=13.6*FF**2/TE(I)
      S5=2.*FF**4*(13.6/TE(I))**1.5
      DO 325 J=JU,JO
      ZZ=FLOAT(J)
```

```
      Y2=S4/ZZ**2
325   S1=S1+S5/ZZ**3*(ALOG(1.+1./Y2)-.4/(1.+Y2)**2)
C
275   AL(M,I)=2.6E-14*S1
225   CONTINUE
C
C  2. DIELECTRONIC RECOMBINATION (NOT FOR HYDROGEN-LIKE IONS)
C
      JO=LZ-1
      JU=LZ-2
C
C  RELEVANT ARE THE EXCITED STATES OF THE RECOMBINING IONS.
C  IF "2"->"1": LEVELS OF "2" (ADDRESS = INDEX + 1).
C
C  BYPASS DIELECTRONIC RECOMBINATION
C
      IF(ALM .EQ. 0) GO TO 550
C
      DO 500 M=2,JO
      J=M+1
      FF=FLOAT(M)
C
      Y1=DE1(J)/(13.6*(FF+1.))
      Y2=DE2(J)/(13.6*(FF+1.))
      S4=1.+.105*Y1+.015*Y1**2
C
C  DIFFERENTLY FOR DELTA-N # 0, NOT FOR RECOMB. TO HE- AND LI-LIKE IONS
C
      IF(G1(J) .LT. 0. .AND. M .LT. JU) S4=2.+.42*Y1+.06*Y1**2
C
      S5=1.+.105*Y2+.015*Y2**2
C
      IF(G2(J) .LT. 0. .AND. M .LT. JU) S5=2.+.42*Y2+.06*Y2**2
C
      S1=F1(J)*SQRT(DE1(J))/S4
      S2=F2(J)*SQRT(DE2(J))/S5
C
      S3=6.5E-10*(FF+1.)**2*SQRT(FF/(FF**2+13.4))
      DO 600 I=1,IR
C
C  DENSITY DEPENDENCE (CAN BE BYPASSED BY NEGATIVE MULTIPLIER)
C
      S6=1.
      S7=1.
      IF(ALM .LT. 0) GO TO 650
C
      ZZ=(1.51E17/RNE(I)*FF**6*SQRT(TE(I)))**(1./7.)
      S6=1./(1.+200./ZZ)
      S7=S6
C
C  DIFFERENT FORMULA FOR DELTA-N # 0
C
      IF(G1(J) .LT. 0.) S6=1./(1.+666.7/((FF+1.)*ZZ)**2)
C
      IF(G2(J) .LT. 0.) S7=1./(1.+666.7/((FF+1.)*ZZ)**2)
C
C  CORRECTION TO EXCITATION ENERGIES OF RECOMBINED IONS
C
650   ZZ=1.+0.015*FF**3/(FF+1.)**2
      Y1=DE1(J)/(TE(I)*ZZ)
```

```
      Y2=DE2(J)/(TE(I)*ZZ)
      S4=0.
      IF(Y1 .LE. 100.) S4=EXP(-Y1)
      S5=0.
      IF(Y2 .LE. 100.) S5=EXP(-Y2)
C
600   AL(M,I)=AL(M,I)+ABS(ALM)*S3/SQRT(TE(I)**3)*
     >(S1*S6*S4+S2*S7*S5)
500   CONTINUE
C
C  NOW F*G FOR TOTAL LINE RADIATION LOSSES
C
550   DO 800 M=1,LZ
      F1(M)=F1(M)*ABS(G1(M))
800   F2(M)=F2(M)*ABS(G2(M))
C
C  INTEGRATION OF NE*S FROM THE EDGE FOR THE CALCULATION OF NEUTRALS
C
      SINT(IR)=-0.0625*S(1,II)*RNE(II)/PRO(II)
      DO 200 I=3,IR
      J=IR+2-I
      SINT(J)=SINT(J+1)+.25*(S(1,J+1)*RNE(J+1)/PRO(J+1)+
     >S(1,J)*RNE(J)/PRO(J))
C
C  CALCULATE IONISATION LENGTH (FOR NORMALIZING THE NEUTRAL DENSITY)
C
200   IF(SINT(J).LT.ABS(V0)) M=J
C
      RION=-V0/(RNE(M)*S(1,M))
C
C  CALCULATE RELATIVE NEUTRAL IMPURITY DENSITY (SCALED LATER)
C
      SINT(1)=0.
      SINT(IR)=1.
      DO 750 I=2,II
750   SINT(I)=RR(IR)/RR(I)*EXP(SINT(I)/V0)
C
      RETURN
      END
```

```
      SUBROUTINE INTP(X,Y,N,L)
C
C  INTP DOUBLES THE NUMBER OF POINTS FOR PLOTTING BY INTERPOLATION
C  ZERO DERIVATIVE ON AXIS IS ASSUMED. ODD NUMBER OF POINTS PREFERRED
C
      DIMENSION X(L),Y(L)
C
      DO 100 I=N,2,-1
      J=2*I-1
      X(J)=X(I)
100   Y(J)=Y(I)
C
      N=2*N-1
C
      DO 200 I=2,N,2
      X0=-X(3)
      Y0=Y(3)
      IF(I .EQ. 2) GO TO 255
      J=I+3
C
      IF(J .LE. N) GO TO 350
      J=I-3
      GO TO 250
C
350   DY2=ABS((Y(I+3)-Y(I+1))/(X(I+3)-X(I+1)))
      DY1=ABS((Y(I-1)-Y(I-3))/(X(I-1)-X(I-3)))
      IF(DY2 .GT. DY1) J=I-3
250   X0=X(J)
      Y0=Y(J)
255   DY2=Y(I+1)-Y(I-1)
      DY1=Y(I-1)-Y0
      DX2=X(I+1)-X(I-1)
      DX1=X(I-1)-X0
      DX0=X(I+1)-X0
      C=(DY2/DX2-DY1/DX1)/DX0
      B=(DY1*(X(I+1)+X(I-1))/DX1-DY2*(X(I-1)+X0)/DX2)/DX0
      A=Y(I-1)-B*X(I-1)-C*X(I-1)**2
      X(I)=.5*(X(I-1)+X(I+1))
200   Y(I)=A+B*X(I)+C*X(I)**2
C
      RETURN
      END
```

```
C+=====================================================================
*
*
      SUBROUTINE RDPAR(TEXT,INPAR,ND,NLEFT,LINE)
*
* =====================================================================
*
* VSFORTRAN SUBROUTINE                                WENDT SEPT.1982
*
* ROUTINE WILL OUTPUT ON TERMINAL STRING 'TEXT', READ AND COUNT NUMBER
* OF INPUT PARAMETERS WHICH WILL BE STORED RIGHT JUSTIFIED, UNLESS
* OTHERWISE SPECIFIED (NLEFT).
* 1 TO ND ELEMENTS OF INPAR ARE CLEARED, IE DEFAULT VALUES = 0,
* OTHERWISE OVERWRITTEN WHEN INPUT FOUND.
* SET INPUT PARAMETER 'ND' NEGATIVE IF INPAR ARRAY ONLY TO BE CLEARED
* WHEN VALUE FOUND.
*
*                    NORD-SYSTEM  JULY 1983 K. BEHRINGER
* PARAMETERS:
* TEXT:        CHARACTER STRING , LENGTH DEFINED OUTSIDE ROUTINE.
*              TEXT OUTPUT ON TERMINAL
*
* INPAR:       CHARACTER ARRAY
*              INPUT PARAMETERS, MAXIMAL 80 CHARACTERS. ITEMS SHOULD BE
*              SEPARATED BY COMMA. WHEN INPUT IS READ, INPAR(I) IS
*              CLEARED AND INPUT STORED RIGHT JUSTIFIED. OTHERWISE
*              INPAR(I) IS RETURNED AS ENTERED.
*
* ND:  INPUT: MAXIMUM NUMBER OF INPUT ITEMS. 1 TO ND ELEMENTS OF ARRAY
*              ARE BLANKED, UNLESS ND IS NEGATIVE, IN WHICH CASE INPAR
*              IS BLANKED ONLY IF NEW INPUT FOUND.
*     OUTPUT: NUMBER OF ITEMS INPUTED,INCLUDING DEFAULTS BEFORE LAST
*              ENTRY
* NLEFT:       NUMBER OF ARRAY TO BE STORED LEFT JUSTIFIED.
*
* LINE:  INPUT: PERFORM LINE LINEFEEDS BEFORE EXECUTING
*
* =====================================================================

C
       CHARACTER*(*) TEXT
       CHARACTER*(*) INPAR(1)
       CHARACTER*1  BLANK
       CHARACTER*72 INPUT
       DATA BLANK/' '/
       N=LEN(TEXT)
C
C      PERFORM LINEFEEDS
C
       DO 103 I=1,LINE
103    WRITE(1) 12B
C
C      BELL, ERASE LINE AND OUTPUT TEXT
C
       WRITE(1) 7B,4B,(TEXT(I:I),I=1,N)
C
C  NLEFT WHEN USED AS INPUT PARAMETER GIVES ARRAY NUMBER OF INPAR WHICH
C  SHOULD BE LOADED LEFT JUSTIFIED. ALL OTHER PARAMETERS LOADED
C  RIGHT JUSTIFIED.
       IPZ=NLEFT
```

```
      NPAR=ABS(ND)
C
C  CHECK DEFINED LENGTH OF 'INPAR'. BLANK ND ITEMS OF ARRAY INPAR
C
      IF(ND.GE.0) THEN
              DO 1 IK=1,NPAR
  1           INPAR(IK)=BLANK
      ENDIF
C
      NCH=LEN(INPAR(1))
      N=72
C
C INPUT CHARACTER STRING LENGTH N ON SAME LINE USING ROUTINE TER
C
      CALL TER(INPUT,N)
C
C      IF N=0 (NO INPUT) 'INPAR' PARAMETERS WILL REMAIN UNCHANGED
C
           ND=0
           I = 1
      IF(N .EQ. 0) RETURN
      NA = 1
C
100   CONTINUE
C
C
C    SEARCH FOR COMMA AND COUNT ITEMS (INCLUDING DEFAULTS)
C    NP = POSITION IN CHARACTER STRING OF COMMA IF > 0
C
      NP=INDEX(   INPUT(NA:N),      ',' )
C
           IF(NP.EQ.0) THEN
                       NEND=N
           ELSE
                       NEND=NA+NP-2
           ENDIF
C
C     COPY TO INPAR(I), RIGHT JUSTIFIED
C     IF NO COMMA FOUND, INPUT IS COPIED TO END OF STRING
C     IF COMMA FOUND FOLLOWING PREVIOUS COMMA, ITEM COUNT RAISED BUT
C     INPAR(I) NOT OVERWRITTEN.
C
      IF(NP.EQ.1)GO TO 10
C
      M=NCH-(NEND-NA)
C
C  COMMA FOUND, BLANK INPAR, COPY PARAMETER EXCLUDING COMMA.
C
      INPAR(I)=BLANK
           IF(I.EQ.IPZ) THEN
                       INPAR(I)(1:)   =INPUT(NA:NEND)
*
* RIGHT JUSTIFIED WHEN INPUT PARAMETER 'ND'.GT.0
           ELSE
                       INPAR(I)(M:NCH)=INPUT(NA:NEND)
           ENDIF
C
  10  CONTINUE
C
C
```

```
C       ITEM COUNT
        ND=I
C
C
        I=I+1
C
C  CHECK THAT NUMBER OF ITEMS NOT GREATER THAN ARRAY
C
           IF(I.GT.NPAR)GO TO 3
        NA=NEND+2
        IF(NA.LE.N)GO TO 100
C
  3        CONTINUE
C
        RETURN
        END
*****************************************************************************
C
        SUBROUTINE TER(INPUT,K)
C
C       FORTRAN 77 ROUTINE FOR INPUT OF CHARACTER STRING WITH POSSIBILITY
C       OF CORRECTIONS. "DEL" OR "CNTRL A" DELETE LAST CHARACTER. IN THE
C       FIELD BLANKS ARE SUBSTITUTED, AT THE END THE FIELD IS SHORTENED.
C       ARROWS LEFT AND RIGHT CAN BE USED TO OVERWRITE PARTS OF FIELD.
C
        CHARACTER INPUT*(*),BL*1
        INTEGER ANT
        INTEGER DEL,EN,CLA,LP,RP
        DATA DEL/177B/,EN/15B/,CLA/1B/,LP/10B/,RP/30B/,BL/' '/
C
C       VARIABLE SIZE
C
        N=K
C
C       INPUT CHARACTER
C
50      I=1
        K=0
75      ANT=INCH(1)
C
C       STRIP PARITY BIT
        ANT=IAND(ANT,177B)
C
C       CHECK FOR RETURN
C
        IF(ANT .EQ. EN) GO TO 200
C
C       CHECK FOR DELETE OR CONTROL A
C
        IF(ANT .NE. DEL .AND. ANT .NE. CLA) GO TO 100
        IF(I .NE. 1) GO TO 85
        WRITE(1) 7B
        GO TO 75
85      WRITE(1) 10B,40B,10B
        I=I-1
        IF(I .EQ. K) K=K-1
        INPUT(I:I)=BL
        GO TO 75
C
C       CHECK FOR ARROW LEFT
```

```
C
100     IF(ANT .NE. LP) GO TO 125
        I=I-1
        IF(I .GT. 0) GO TO 75
        WRITE(1) 7B,30B
        I=1
        GO TO 75
C
C       CHECK FOR ARROW RIGHT
C
125     IF(ANT .NE. RP) GO TO 150
        IF(I .GT. K) INPUT(I:I)=BL
        I=I+1
        IF(I .GT. N) GO TO 200
        GO TO 75
C
C       DON'T ACCEPT CONTROL CHARACTERS
C
150     IF(ANT .LT. 40B) GO TO 75
C
C       PUT CHARACTER INTO STRING
C
        INPUT(I:I)=CHAR(ANT)
        IF(I .GT. K) K=I
        I=I+1
        IF(I .GT. N) GO TO 200
        GO TO 75
C
200     RETURN
C
C       WRITE(1,20) INPUT(1:K)
C20     FORMAT(/,1X,A,/)
C       WRITE(1,10) ANT
C10     FORMAT(1X,06)
        END
```

```
      SUBROUTINE CRT011(ICODE,TEXTW,TEXTX,TEXTY,XMIN,XMAX,YMIN,
     > YMAX,N,X,Y)
C
C     ROUTINE FOR PLOT OUTPUT (PLOT10) 19.5.83 KBE
C
      DIMENSION X(1),Y(1),LAB(72)
      CHARACTER*(*) TEXTW,TEXTX,TEXTY
      CHARACTER*13 ELA
      COMMON /FSIZE/ IXA,IXB,IYA,IYB
      DATA IXA,IXB,IYA,IYB/150,1000,230,730/
C
C     CHOICE OF ICODE:
C     L.S. DIGIT: 0    ERASE, NEW FRAME, LIN PLOT
C                 1           NEW FRAME, LIN PLOT
C                 2    KEEP FRAME, LIN PLOT
C     NEXT DIGIT: SOLID, DASHED LINE ETC., >5 MEANS 5 - VALUE
C     NEXT DIGIT: SYMBOL PLOTTING
C
C     EXTRACT DIGITS
C
      K=INT(ICODE/10)
      L1=ICODE-K*10
      L3=INT(K/10)
      L2=K-L3*10
      IF(L2 .GT. 5) L2=5-L2
C
C     WRITE EXPONENT SEPARATELY
C
      IP=0
      YM=YMAX
      YMI=YMIN
      DIV=-YMIN
      IF(YMAX .GT. DIV) DIV=YMAX
      IF(DIV .GT. .1 .AND. DIV .LE. 100) GO TO 50
C
      WRITE(ELA,5) DIV
5     FORMAT(E13.6)
      READ(ELA(11:),10) IP
      IF(IP .LE. 0) IP=IP-1
      WRITE(ELA(11:),10) IP
10    FORMAT(I3)
      DIV=10.**IP
      YM=YMAX/DIV
      YMI=YMIN/DIV
C
      DOFOR I=1,N
      Y(I)=Y(I)/DIV
      ENDDO
      ELA(7:10)=' x E'
C
50    IF(L1 .NE. 0) GO TO 100
C
C     INITIATE AND DRAW AXES
C
      CALL INITT(960,1)
      CALL BINITT
C
100   IF(L1 .GT. 1) GO TO 200
C
      WRITE(1) 33B,61B
```

```
      CALL SLIMX(IXA,IXB)
      CALL SLIMY(IYA,IYB)
      CALL DLIMX(XMIN,XMAX)
      CALL DLIMY(YMI,YM)
      CALL NPTS(N)
      CALL LINE(L2)
      CALL SYMBL(L3)
      CALL CHECK(X,Y)
      CALL DSPLAY(X,Y)
      CALL OAXIS
C
C     CONVERT AND WRITE LABELS
C
      K=LEN(TEXTW)
      DOFOR I=1,K
      LAB(I)=ICHAR(TEXTW(I:I))
      ENDDO
      K1=(IXA+IXB-K*13)/2
      K2=IYB+20
      CALL NOTATE(K1,K2,K,LAB)
C
      IF(IP .EQ. 0) GO TO 150
      DOFOR I=1,7
      K=I+6
      LAB(I)=ICHAR(ELA(K:K))
      ENDDO
      CALL NOTATE(0,K2,7,LAB)
C
150   K=LEN(TEXTX)
      DOFOR I=1,K
      LAB(I)=ICHAR(TEXTX(I:I))
      ENDDO
      K1=(IXA+IXB-K*13)/2
      K2=IYA-55
      CALL NOTATE(K1,K2,K,LAB)
C
      K=LEN(TEXTY)
      DOFOR I=1,K
      LAB(I)=ICHAR(TEXTY(I:I))
      ENDDO
      K1=IXA-120
      K2=(IYA+IYB+K*25)/2
      CALL MOVABS(K1,K2)
      CALL VLABEL(K,LAB)
C
      K2=IYA-55
      CALL MOVABS(0,K2)
      CALL ANMODE
C     CALL WALPHA
      GO TO 300
C
200   IF(L1 .GT. 2) GO TO 300
C
C     PLOT IN EXISTING FRAME
C
      WRITE(1) 33B,61B
      CALL SLIMX(IXA,IXB)
      CALL SLIMY(IYA,IYB)
      CALL DLIMX(XMIN,XMAX)
      CALL DLIMY(YMIN,YM)
```

```
      CALL NPTS(N)
      CALL LINE(L2)
      CALL SYMBL(L3)
      CALL CHECK(X,Y)
      CALL CPLOT(X,Y)
      K2=IYA-55
      CALL MOVABS(0,K2)
      CALL ANMODE
C     CALL WALPHA
C
300   IF(IP .EQ. 0.) GO TO 400
C
      DOFOR I=1,N
      Y(I)=Y(I)*DIV
      ENDDO
C
400   RETURN
      END
```