INT 179/92 July 1992

SEMAL Documentation

O. Sauter

SEMAL DOCUMENTATION

Olivier Sauter

Centre de Recherches en Physique des Plasmas
Association Euratom - Confédération Suisse
Ecole Polytechnique Fédérale de Lausanne
Av. des Bains - CH 1007 Lausanne - Switzerland

TABLE OF CONTENTS

SEMAL DOCUMENTATION

1. General description

The code **SEMAL** solves the electromagnetic wave equations to all orders in Larmor radii. These equations are derived from the linear Vlasov-Maxwell equations. Their derivation as well as the final equations are described in Refs. 1 and 2. One assumes a slab geometry with inhomogeneity along x and equilibrium magnetic field $\mathbf{B}_0$ along z, Maxwellian equilibrium distribution functions and k_y=0 in the perturbed current. The geometry is shown in Fig.1. The resulting equation for the electric field **E** is the following [Eq.(4.10), Ref.1]:

$$\nabla \wedge \nabla \wedge \mathbf{E}(x) - \frac{\omega^2}{c^2} \left(\underline{\underline{\varepsilon}}\,\mathbf{E}\right)(x) = 0 , \qquad (4.10)$$

with

$$\nabla = \left(\frac{d}{dx}, i\,k_y, i\,k_z\right) ,$$

$$\left(\underline{\underline{\varepsilon}}\,\mathbf{E}\right)(x) = \underline{\underline{I}}\,\mathbf{E}(x) + \sum_{\sigma} \int dx'' \, \frac{\omega_{p\sigma}^2(x'')}{\sqrt{2\pi}\,\omega^2 \rho_\sigma} \exp\left[- \frac{(x-x'')^2}{2\,\rho_\sigma^2} \right] \underline{\underline{D}}\,\mathbf{E}(x)$$

$$+ \sum_{\sigma,n} \int dx'' \; dx' \int_0^\pi d\theta \, \frac{\omega_{p\sigma}^2(x'')}{2\pi^2 \omega^2 \rho_\sigma^2(x'')} \exp\left\{ - \frac{\left(x'' - \frac{x+x'}{2}\right)^2 (1 - \cos\theta)}{\rho_\sigma^2(x'') \sin^2\theta} \right\}$$

$$\times \exp\left\{ - \frac{(x-x')^2 (1+\cos\theta)}{4\,\rho_\sigma^2(x'') \sin^2\theta} \right\} \left[\frac{\omega}{|k_{//}| v_{T\sigma//}} Z_{n\sigma} - (1+\xi_{n\sigma} Z_{n\sigma})\left(1 - \frac{T_{\sigma\perp}}{T_{\sigma//}}\right) \right] \underline{\underline{M}}\,\mathbf{E}(x')$$

where

$$\underline{\underline{M}} = \begin{pmatrix} n \sin n\theta & \begin{array}{c} -i \dfrac{\sin n\theta}{\sin^2\theta} \dfrac{(x'-x'')}{\rho_\sigma^2} \times \\ \times [(x-x'')\cos\theta - (x'-x'')] \end{array} & \begin{array}{c} i \dfrac{\sin n\theta}{\sin^2\theta} \Xi_{n\sigma} \times \\ \times \dfrac{[(x-x'')\cos\theta - (x'-x'')]}{\rho_\sigma} \end{array} \\ M^*_{xy}(x',x) & \dfrac{\cos n\theta}{\sin\theta} \dfrac{(x-x'')(x'-x'')}{\rho_\sigma^2} & -\dfrac{\cos n\theta}{\sin\theta} \Xi_{n\sigma} \dfrac{(x-x'')}{\rho_\sigma} \\ M^*_{xz}(x',x) & M^*_{yz}(x',x) & \dfrac{\cos n\theta}{\sin\theta} \Xi_{n\sigma}^2 \end{pmatrix},$$

$$\underline{\underline{D}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -1 + \dfrac{(x-x'')^2}{\rho_\sigma^2} & -\Xi_0 \dfrac{(x-x'')}{\rho_\sigma} \\ 0 & D_{yz} & \Xi_0^2 \end{pmatrix}.$$

with

$$\omega_{c\sigma} = \frac{q_\sigma B_0}{m_\sigma}, \quad \omega_{p\sigma}^2 = \frac{n_\sigma q_\sigma^2}{\varepsilon_0 m_\sigma}, \quad v_{T\sigma(\perp,//)}^2 = \frac{2T_{\sigma(\perp,//)}}{m_\sigma}, \quad \rho_\sigma = \frac{v_{T\sigma\perp}}{\sqrt{2}|\omega_{c\sigma}|}.$$

$$\Xi_{n\sigma} = \frac{\omega - n\omega_{c\sigma}}{k_{//}\omega_{c\sigma}} \frac{1}{\rho_\sigma},$$

$$Z_{n\sigma} = Z_{n\sigma}(\xi_{n\sigma}) \; ; \; \xi_{n\sigma} = \frac{\omega - n\omega_{c\sigma}}{|k_{//}| v_{T\sigma//}},$$

and where $Z_{n\sigma}$ is the plasma dispersion function [2]. Note that one can easily relate the terms D_{yz}, D_{zz} and M_{yz}, M_{zz} with the following relation:

$$\sum_n \int dx' \int_0^{\pi} d\theta \exp\{-\frac{(x-x')^2(1+\cos\theta)}{4\rho_\sigma^2(x'')\sin^2\theta}\} \exp\{-\frac{(x''-\frac{x+x'}{2})^2(1-\cos\theta)}{\rho_\sigma^2(x'')\sin^2\theta}\} \frac{\cos n\theta}{\sin\theta}$$

$$= \pi\sqrt{2\pi}\,\rho_\sigma \exp\{-\frac{(x-x'')^2}{2\rho_\sigma^2}\} .$$

Therefore, ε_{yz} and ε_{zz} can also be written as :

$$(\varepsilon_{yz}E_z) = \sum_{\sigma,n}\int dx''\cdots\frac{\omega}{k_{//}v_{T\sigma//}}\frac{2\,\omega_{c\sigma}}{v_{T\sigma\perp}}\sqrt{\frac{T_{\sigma//}}{T_{\sigma\perp}}}\,(1+\xi_{n\sigma}Z_{n\sigma})\frac{\cos n\theta}{\sin\theta}(x-x'')E_z(x'),$$

and

$$(\varepsilon_{zz}E_z) = E_z(x)+\sum_{\sigma,n}\int dx'\cdots\frac{\omega}{|k_{//}|v_{T\sigma//}}\frac{2T_{\sigma//}}{T_{\sigma\perp}}\,\xi_{n\sigma}\,(1+\xi_{n\sigma}Z_{n\sigma})\frac{\cos n\theta}{\sin\theta}E_z(x'),$$

The code **SEMAL** solves Eq.(4.10) with this latter form for ε_{yz} and ε_{zz} and with, in the present version, $T_{\sigma\perp} = T_{\sigma//} = T_\sigma$. Thus, only D_{yy} remains in the matrix $\underline{\underline{D}}$ and this term, for historical reasons, is refered to as the "special yy term" or "B_{yy}" in the code.

The boundary conditions are described in Sec.5.2 of Ref.1. They assume either a sheet antenna in the vacuum surrounding the plasma, as in Fig.1, or impose the value of the tangential field (E_y, E_z) on both sides of the plasma.

One of the main diagnostic of the code is the power deposition profile, obtained following the procedure described in Ref. 3. With the same assumptions as for Eq.(4.10), one finds a positive-definite formula for the profile valid to all orders in Larmor radii:

$$P_L(x) = \sum_{n,\sigma}\frac{2q_\sigma^2}{\pi^{5/2}m_\sigma}\int dx''\int_0^{\frac{\pi}{2}} d\theta\,\frac{n_\sigma(x'')}{|k_{//}|v_{T\sigma//}(x'')}\,\frac{\omega_{c\sigma}^4(x'')}{v_{T\sigma\perp}^4(x'')}\,\frac{|x-x''|^3}{\cos^4\theta}$$

$$\times\, e^{-\xi_{n\sigma}^2}\exp\{-\frac{(x-x'')^2}{2\rho_\sigma^2\cos^2\theta}\}\,\frac{T_{\sigma\perp}}{T_{\sigma//}}\,[1-\frac{n\omega_{c\sigma}}{\omega}(1-\frac{T_{\sigma//}}{T_{\sigma\perp}})] \qquad (4.19)$$

$$\times\left|\int_0^{\pi} d\theta'\,[E_x\sin\theta'\sin n\theta' + i\cos n\theta'\,(\cos\theta' E_y + \frac{(\omega-n\omega_{c\sigma})\cos\theta}{k_{//}\,\omega_{c\sigma}|x-x''|}E_{//})]\right|^2 ,$$

with $\mathbf{E} = \mathbf{E}[\,x'' - \frac{|x-x''|}{\cos\theta}\cos\theta'\,]$.

The code ISMENE [4] solving Eq.(4.10) expanded up to second order in Larmor radius, kindly provided by Dr. K. Appert, has been incorporated in the code SEMAL. The formula for the power deposition profile valid up to second order in Larmor radius [5] has also been included in the code.

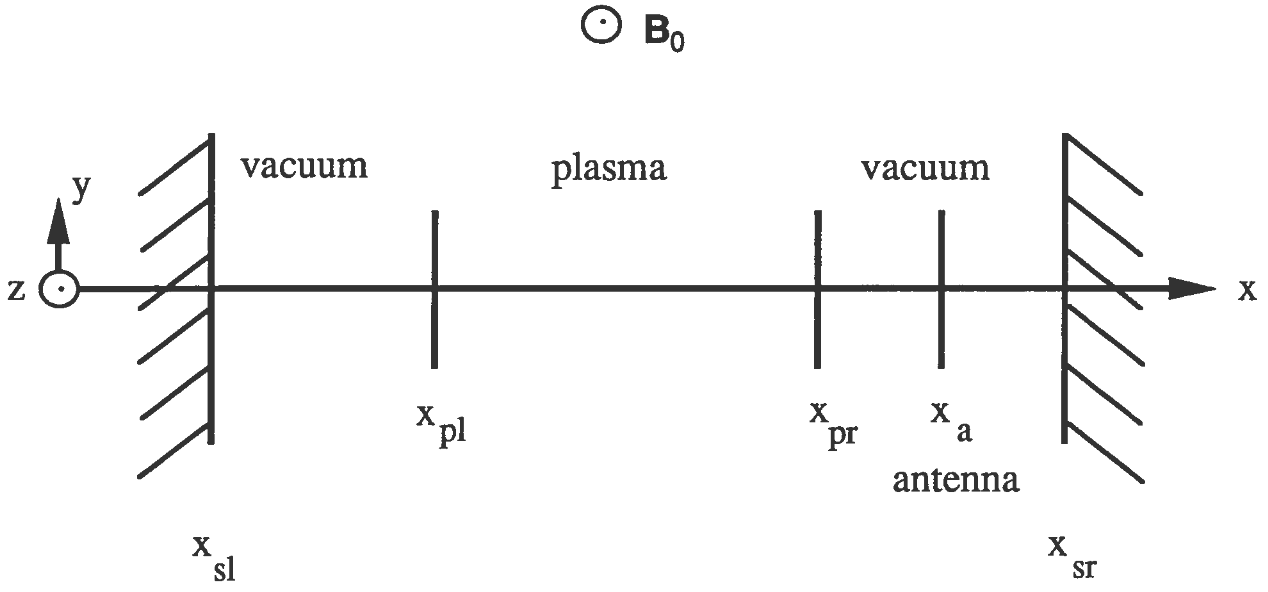


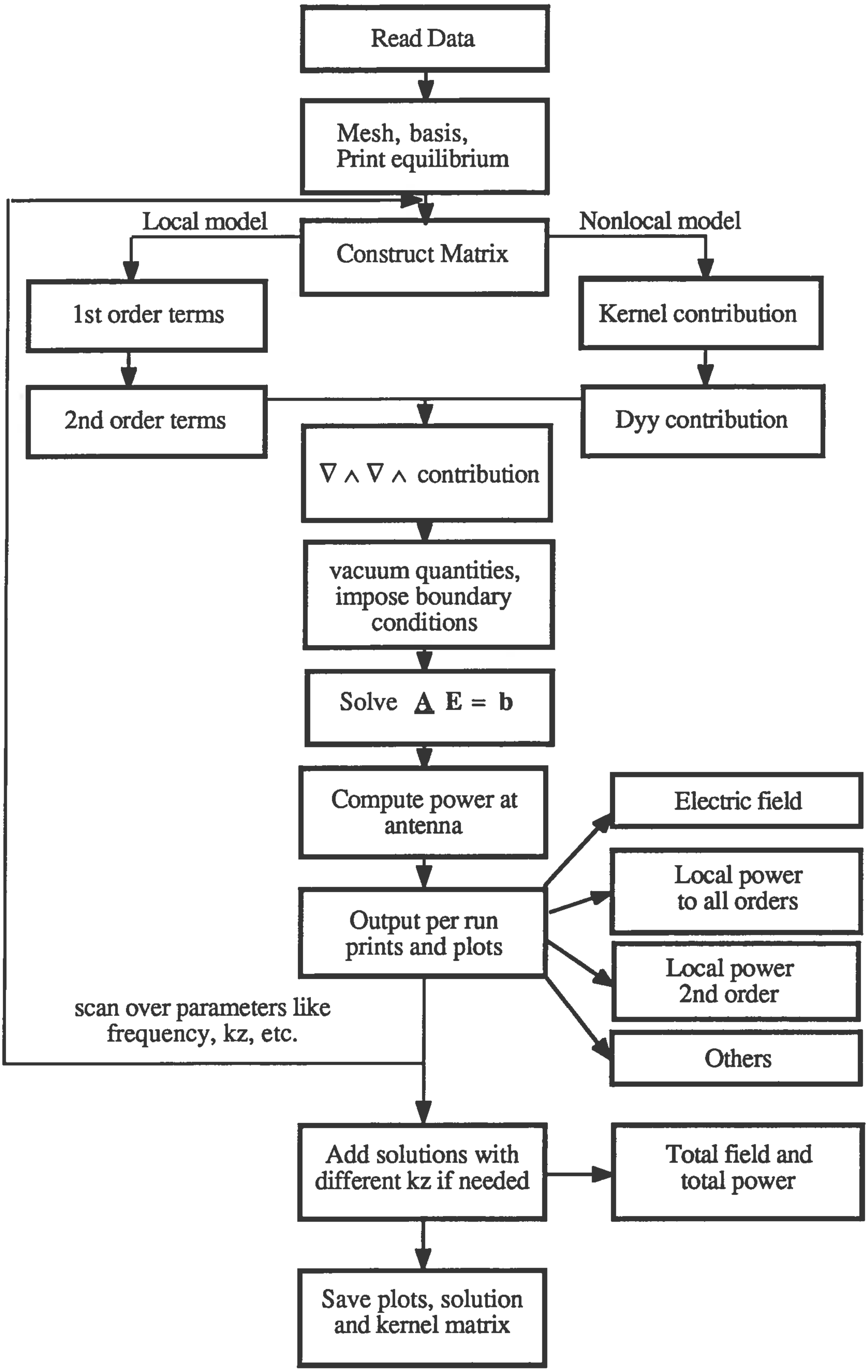
Fig.1: Configuration of the slab model. The plasma is situated between two vacuum regions which are limited by perfectly-conducting walls. An idealized antenna is located in vacuum on the right-hand side of the plasma. The current flows in the (y, z) plane. The equilibrium magnetic field $B_0(x)=B_0R_0/(R_0\pm x)$ is parallel to the z-axis.

2. General structure of the code

The structure of the code **SEMAL** is shown in Table 1. The local model refers to the code ISMENE, while the nonlocal option means that the full second-order integro-differential equation is solved. The data are read in the form of a namelist named NEWRUN. Naglib subroutines, for some interpolations, and Linpack subroutines, for solving the full complex linear system, are used in the code apart from standard Fortran 77 subroutines. But subroutines from Numerical Recipes [6] or Dr. K. Appert can be used by setting NLINPK = .F. and/or NLNAG = .F..

At the beginning of the code, a few blocks describe all the subroutines and the common blocks and their variables used in the code. Therefore, only the subroutines and variables which need additional description will be mentioned in this user manual, in particular those which are in the namelist.

Table 1. Structure of SEMAL


Read Data
Mesh, basis, Print equilibrium
Local model
Construct Matrix
Nonlocal model
1st order terms
Kernel contribution
2nd order terms
Dyy contribution
∇ ∧ ∇ ∧ contribution
vacuum quantities, impose boundary conditions
Solve A E = b
Compute power at antenna
Electric field
Output per run prints and plots
Local power to all orders
Local power 2nd order
scan over parameters like frequency, kz, etc.
Others
Add solutions with different kz if needed
Total field and total power
Save plots, solution and kernel matrix

3. Namelist

The input file (unit 5) should have the following structure:

First, two lines for labelling
the run
$NEWRUN
  BNOT= 2.71, *etc. (all input variables which you want to change)*
$END

The variables in the namelist are listed, with a small comment, in Appendix A. The last column refers to the subsection of this documentation where the variable is explained. The third column gives the type of the variable (Integer, Real, Complex, Logical), specifies whether or not it is an array (A) and gives the block number of the common block in which they are defined.

We give here two examples of Namelist with the resulting antenna power and value of $\underline{E}(x_{pr})$, which gives a check of the code as well.

3.1 Empty Namelist

You can run the code without specifying any parameters. Thus all the parameters will have their default values, specified in the subroutine PRESET. It is the case of an Alfvén mode corresponding to Fig. 7.5 of Ref. 1 with $\omega/\omega_{ci} = 0.298$, but with very few NX points. The input data should have the following form :

Test job nr.1:
Empty namelist using default values as parameters
$NEWRUN
$END

As a result, on the output file, you will obtain a few tables describing the equilibrium parameters, the power emitted by the antenna and the fields at a few x points.

The power emitted by the antenna should be equal to :

$$P_a = (2.183\ \text{E-07};\ 1.994\ \text{E-04})\ [\text{W}/\text{m}^2]$$

The value of the fields, in V/m, at x = 0.08 [m] are:

E_x	E_y	E_z
(4.097 E-02; 1.758 E-02)	(4.752 E-05; 1.392 E-02)	(7.498 E-05; 2.515 E-06)

Note that in this case, only the local model has been used, as NLKREL and NLKRIO are .FALSE.. If you want to run the same case using the nonlocal model, you have to specify the following parameters :

NLKREL = .T., NLKRIO = .T., METHOD = 3, LMESHB = .T.,

The last parameter specifies a special packing very near the edge of the plasma (see Sec. 4.5). It is needed here because we have taken homogeneous profiles. Therefore, there is a large step at the edge. In this case, the values of P_a and **E**(0.08) are :

P_a = (1.802 E-06; 1.721 E-04)

$E_x(0.08)$	$E_y(0.08)$	$E_z(0.08)$
(-9.918 E-01; 2.099 E-01)	(-4.805 E-04; 4.281 E-03)	(9.548 E-05; 1.785 E-05)

3.2 D-T JET plasma

As a second example, illustrating what are the minimal parameters needed to describe a new experiment, we describe a D-T plasma with 50-50 deuterium and tritium corresponding to the case of Fig. 7.9 of Ref. 1 with n_α/n_e = 0%. The input data are of the following form:

```
Test Job nr. 2 : D-T plasma
Equilibrium mesh
$NEWRUN
BNOT = 2.71, NRSPEC = 3,
AMASS(2) = 2.0, 3.0, ACHARG(2) = 1.0, 1.0
CENDEN(2) = 0.5 E+20, 0.5 E+20
CENTMP(1) = 5000., 5000., 5000.,
EQUDEN = 0.95, 0.95, 0.95
EQUTMP = 0.95, 0.95, 0.95
RMAJOR = 2.96, XSHLOW = - 1.30, XSHUP = 1.30, XANTEN = 1.26
XPLEFT = - 0.25, XPRIGT = 1.25, EDGPRO = 1.25,
FNOT = 0.0, DLOMCI = 1.0,
ANTKY = 0.0, ANTKZ = 5.0,
NLKREL = .T., NLKRIO = .T., METHOD = 3,
NX = 100, NXPP = 100,
NLOTP2(8) = .T., .T., NQX = 50
NQRSPC = 3,
NQL1 = 0, 2, 3,
NQL2 = 0, 2, 4,
$END
```

In this case : P_a = (3.615 E-03, 7.434 E-03) and the value of **E** (x = 1.25) is :

E_x	E_y	E_z
(7.220 E-01, 5.257 E-01)	(-2.038 E-01, 2.740 E-01)	(-5.529 E-03, 6.869 E-03)

The output of the power absorbed by each species using the expanded formula has also been activated (NLOTP2(9) = .T.). It gives a total absorbed power of 1.23026E-03 W/m^2, where 4.1267E-04 W/m^2 are absorbed by the electrons and 8.1759E-04 W/m^2 by the deuterium. Using the nonlocal formula (NLOTP2(8) = .T.), it gives a total power of 1.79046E-03 W/m^2, 5.57707E-04 W/m^2 absorbed by the electrons, 1.16567E-03 W/m^2 by the deuterium and 6.70840E-05 W/m^2 by the tritium.

Note that the time used to compute the power using the nonlocal formula can be reduced by adding the following parameters in the namelist NEWRUN :

```
NINTQX = 0, 1, 2,
QXLEFT(1, 2) = - 0.15, QXRIGT(1, 2) = 0.15,
QXLEFT(1, 3) = - 0.15, 0.90, QXRIGT(1, 3) = 0.15, 1.10,
```

To reduce the time even more, one can compute the power absorbed by the electrons and the deuterium using the expanded formula and by tritium using the full (nonlocal) formula :

```
NINTQX = - 1, - 1, 2,
QXLEFT(1, 3) = - 0.15, 0.90, QXRIGT(1, 3) = 0.15, 1.10,
NLPOMX = .T.
NSEXPI = 1, 2, NSEXPT = 2,
NQL1TO = 0, 2, 3,
NQL2TO = 0, 2, 4,
```

Note that the full formula has to be used for the tritium because it is the third and fourth harmonic of the tritium which contribute.

4. Specification of equilibrium profiles

4.1 Standard parameters

These parameters have all to be specified when starting with a new scenario : ACHARG, AMASS, BNOT, CENDEN, CENTMP, EQUDEN, EQUTMP, EXPDEN, EXPTMP, EDGPRO, RMAJOR, NLHFS, XPLEFT, XPRIGT, NLTOKA, RMINOR. The parameter NRSPEC specifies the total number of species (electron + ions).

The first two specify the charge and mass of each species. ACHARG(1) and AMASS (1) are already set to the correct value for the electron and should not be changed. BNOT, CENDEN and CENTMP are the value of the "toroidal" magnetic field (parallel to O_z), the density (per species) and temperature (per species) at x = 0.0, respectively. The electron density is computed using the neutral condition. The density and temperature profiles are of the following type :

$$p(x) = p_o \cdot \left(1 - s\left(\frac{x}{x_{edg}}\right)^2\right)^\gamma$$

where p_0= CENDEN or CENTMP, s = EQUDEN or EQUTMP, γ = EXPDEN or EXPTMP for the density or the temperature respectively. The coordinate of the edge of the profile is x_{edg} = EDGPRO. Thus, s is the step-value of the profile at $x = x_{edg}$ if the plasma is connected to vacuum at this point (i.e. if XPRIGT = EDGPRO). One sees that at $x = x_{edg}$:

$$p(x_{edg}) = p_0\,(1 - s)^\gamma \underset{\uparrow \text{ if } \gamma=1}{=} p_0\,(1 - s)$$

A typical value of γ is one or two, and of s is 0.95 or 0.99, which gives a step-value between 5% and 0.01% of the central value.

The value of RMAJOR, major axis of the tokamak, gives the toroidal magnetic field profile :

$$B_0(x) = \text{BNOT} \cdot \frac{1}{\left(1 \pm \frac{x}{\text{RMAJOR}}\right)}$$

where the minus sign is taken if NLHFS = .T. and the plus sign otherwise. This depends whether the antenna, always located at the right-hand side of the plasma (XANTEN > XPRIGT), should be on the high- or low-field side respectively.

The domain of x is specified by the left, XPLEFT, and right, XPRIGT, boundaries, at which the boundary conditions due to the plasma-vacuum interface are imposed. If NLTOKA = .T. then XPLEFT = - RMINOR and XPRIGT = + RMINOR, where RMINOR is the minor radius of the tokamak.

4.2 Frascati FTU type of profile

These profiles are used if NLCESA = .TRUE.. They are defined as follows, according to Dr. R. Cesario of ENEA/Frascati - Italy :

$$n(x) = \begin{cases} (n_0 - n_L) \cdot \left(1 - \left(\frac{x}{x_L}\right)^2\right) + n_L & 0 < x < x_L \\ n_L \cdot \exp\left\{ - \frac{(x - x_L)}{x_{pr}\, s_\lambda} \right\} & x_L < x < x_{pr} \end{cases}$$

$$T(x) = \begin{cases} (T_0 - T_L) \left(1 - \left(\frac{x}{x_L}\right)^2\right)^{2.5} + T_L & 0 < x < x_L \\ T_L \cdot \exp\left\{ - \frac{(x - x_L)}{x_{pr} \cdot \delta_s} \right\} & x_L < x < x_{pr} \end{cases}$$

where

$n_e = n_i = n(x)$; $T_e = T_i = T(x)$
n_0 = CENDEN
n_L = DENLIM, density at limiter
x_L = RLIMIT, limiter radius
x_{pr} = XPRIGT, right edge of plasma
s_λ = SLAMBDA, e-fold of density in scrape-off layer
(Note that s_λ is normalized by $\frac{0.35}{\text{XPRIGT}}$ in AUXVAL.)
T_o = CENTMP
T_L = TEMLIM, temperature at limiter
δ_s = DELTAS, e-fold of temperature in scrape-off layer

In general, one solves in a very small interval in the scrape-off layer as the ion Bernstein wavelength is very short.

The magnetic field is defined in the same way as in Sec. 4.1.

4.3 General Atomics DIII-D type of profile

These profiles are used if NLCHIU = .TRUE.. They are defined as follows, according to Dr. S.C. Chiu of GA/San Diego, U.S.A. :

$$n(x) = \begin{cases} (n_0 - n_L) \cdot \left(1 - \left(\frac{x}{x_L}\right)^2\right) + n_L & 0 < x < x_L \\[2ex] n_L - (n_L - n_2) \cdot \frac{x - x_2}{WN_2} & x_L < x < x_2 + WN2 = x_2 \\[2ex] n_2 \cdot \exp\left\{- \frac{(x - x_2)}{s_\lambda}\right\} & x_2 < x < x_3 = x_2 + WDGAP \\[2ex] n_3 - (n_3 - n_{shield}) \cdot \frac{x - x_3}{x_4 - x_3} & x_3 < x < x_4 = x_{shield} = x_3 + WDSHI \end{cases}$$

$$T(x) = \begin{cases} (T_0 - T_L) \cdot \left(1 - \left(\frac{x}{x_L}\right)^2\right) + T_L & 0 < x < x_L \\[2ex] T_L \cdot \exp\left\{- \frac{(x - x_L)}{\delta_s}\right\} & x_L < x < x_2 = x_L + WT2 \\[2ex] T(x_2) \cdot \exp\left\{- \frac{x - x_2}{\delta_{s2}}\right\} & x_2 < x \end{cases}$$

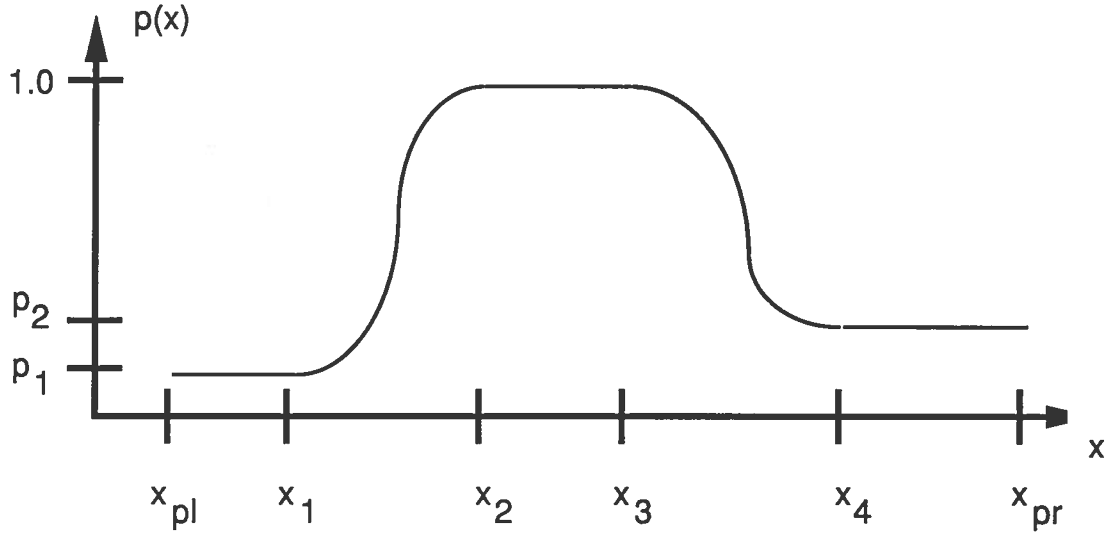
where

$n_e = n_i = n(x)$; $T_e = T_i = T(x)$
n_0 = CENDEN
n_L = DENLIM, density at limiter
x_L = RLIMIT, limiter radius
n_2 = DENWN2, density at x_2 = RLIMIT + WN2
s_λ = SLAMDA, e-fold of density in scrape-off layer
$n_3 = \text{DENWN2} \cdot \exp\left\{- \frac{\text{WDGAP}}{\text{SLAMDA}}\right\}$
n_{shield} = DENSHI, density at Faraday shield
T_0 = CENTMP
T_L = TEMLIM, temperature at limiter
δ_s = DELTAS, first e-fold of temperature in scrape-off layer
δ_{s2} = DELTS2, seconde-fold of temperature in scrape-off layer

The magnetic field is defined in the same way as in Sec. 4.1.

4.4 Cubic profiles

These profiles are used if NLCUB = .T.. They were defined at the very beginning, but they are probably not very useful. They are defined in five sections :



The first, third and fifth sections are constant and the other two are cubic polynomials with zero slopes at both ends. The six coefficients are given by :

$[x_1, x_2, x_3, x_4, p_1, p_2]$ = [ADN(1), ADN(2), ADN(3), ADN(4), ADN(5), ADN(6)]
and
$[x_1, x_2, x_3, x_4, p_1, p_2]$ = [ATP(1,j),ATP(2,j),ATP(3,j),ATP(4,j),ATP(5,j),ATP(6,j)],
j = 1, NRSPEC (species)

for the density and temperature respectively. The profiles p(x) are then scaled by CENDEN and CENTMP, the central values of the density and temperatures, respectively. The electron density is calculated using the neutrality condition.

The magnetic field is defined in the same way with :

$[x_1, x_2, x_3, x_4, p_1, p_2]$ = [ABT(1), ABT(2), ABT(3), ABT(4), ABT(5), ABT(6)]

4.5 Mesh packing

The different subroutines constructing the meshes are called in subroutine AUXVAL. The standard one is MESH (Sec. 4.5.1 and 4.5.2) where the meshes are constructed either uniformly or with some packing. A special mesh is used when the x" domain (Eq.4.10) does not coincide with the x domain (MESHSP, Sec. 6.2). Finally, the x" mesh is in general (if LADAPP = .T.) adapted to the x mesh in subroutine MESHPP (Sec. 4.5.3).

4.5.1 *Standard uniform and nonuniform mesh*

The main parameters controlling the x mesh packing are the following :

NX, NMESHX, NPOIDX, SOLPDX, AXPLAC(10), AXWIDT(10)

NX is the number of intervals. If (NMESHX = 0 or NPOIDX = 0 or SOLPDX = 1.0) then the mesh is equidistant. Otherwise, the mesh is packed around the NPOIDX points x=AXPLAC(i), with a characteristic width of AXWIDT(i), i = 1, NPOIDX. NPOIDX should be smaller or equal to 10. The packing is based on Lorentzians of the type :

$$f(x) = \frac{w_0}{w_0^2 + (x - x_0)^2}$$

where x_0 is the centre of the peak and w_0 its half-width at half height (see Fig. 2). The packed mesh is given by (see Fig. 2):

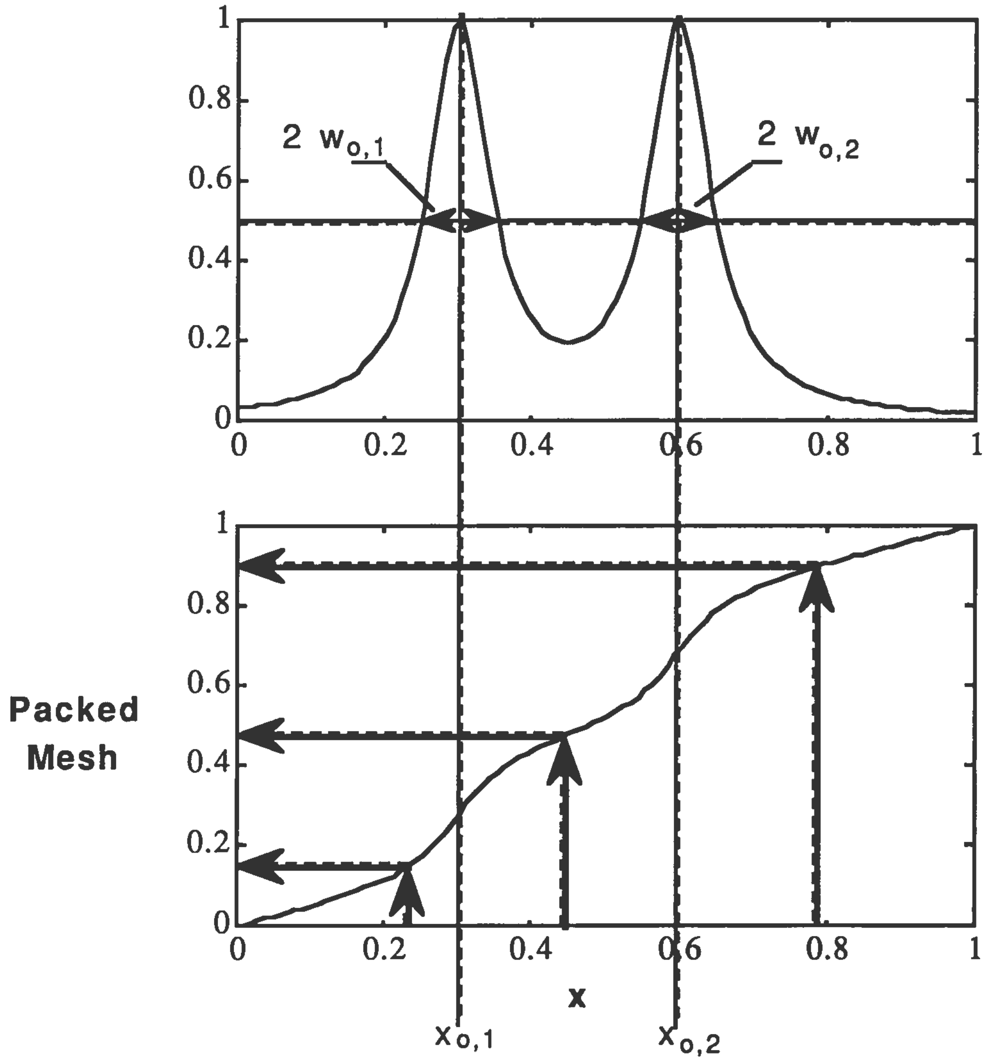
$$x_{packed} = p\,x + (1 - p)\frac{\int_0^x \sum_{i=1}^{N} \frac{w_{0,i}}{w_{0,i}^2 + (x - x_{0,i})^2}\,dx}{\int_0^1 \sum_{i=1}^{N} \frac{w_{0,i}}{w_{0,i}^2 + (x - x_{0,i})^2}\,dx}$$

$$= p\,x + (1 - p)\frac{\sum_{i=1}^{N} \mathrm{Atan}\left(\frac{x - x_{0,i}}{w_{0,i}}\right) + \mathrm{Atan}\left(\frac{x_{0,i}}{w_{0,i}}\right)}{\sum_{i=1}^{N} \mathrm{Atan}\left(\frac{1 - x_{0,i}}{w_{0,i}}\right) + \mathrm{Atan}\left(\frac{x_{0,i}}{w_{0,i}}\right)}$$

The corresponding namelist variables are:

$x_{o,i}$	<------>	AXPLAC(i), i=1,...,NPOIDX
$w_{o,i}$	<------>	AXWIDT(i), i=1,...,NPOIDX
p	<------>	SOLPDX
N	<------>	NPOIDX (≤ 10)

Figure 2 : "Standard" mesh packing



For the x" mesh, we have the same possibility with NMESHP, NPOIDP, SOLPDP, APPLAC and APWIDT. However, we have found out afterwards that we obtain a much better converged solution if we take the same mesh for x" as for x. This is why we have taken these parameters out of the namelist and leave the variable LADAPP equal to .TRUE.. In this way, the x" mesh is adapted to the x mesh after the x packing has been performed. If NXPP = NX/2, then NXPP is set to NX/2 and the x" mesh matches every second point of the x mesh. If NXPP = NX, then NXPP = NX and x" ≡ x. In this case, we also specify NGXPP = 1 (in subroutine AUXVAL). The adaptation of the x" mesh to the x mesh is done in the subroutine MESHPP. For the x" mesh, one should always specify :

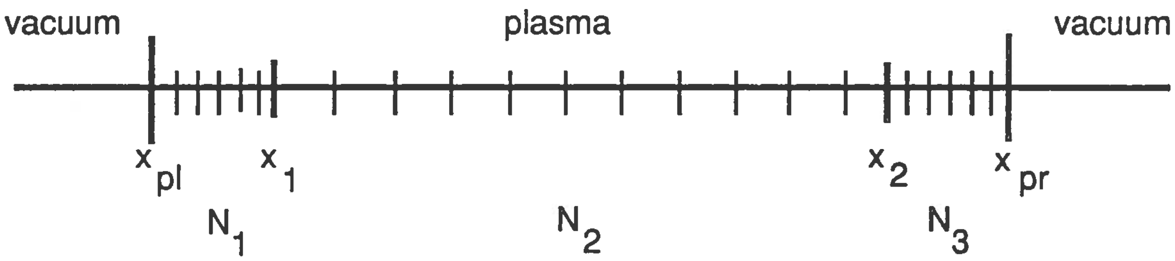
NXPP = NX and NGXPP = 1.

NGXPP might be set equal to 12, specifying two integration points at $^1/_4$ and $^3/_4$, but it doubles the time and is in general not necessary. Note that if one wants to modify the integration points for the x" integral, two characteristics should be kept in mind :

a) The x" integration points should not coincide with the x nodes.
b) The x" integration points should be as uniformly distributed with respect to the x nodes as possible.

4.5.2 *Edge packing*

This packing (applied if LMESHB = .TRUE.) is in general useful if the waves studied have a large electrostatic component or if there is a large step-value of the density at the plasma-vacuum interface (for example with homogeneous profiles) with respect to the central density. In these cases there might be a Debye-shielding effect which results in a large drop of the field amplitude on a scale-length comparable to the electron Larmor radius or the electron or ion Debye length. Therefore, in order to resolve this drop, a special packing very near the edge is necessary :



Two equal intervals $\Delta x = x_1 - x_{pl} = x_{pr} - x_2$, near each edge of the plasma, are uniformaly divided into (NX/NXBOUN) sub-intervals. In the rest of the x domain, $[x_1, x_2]$, the x mesh is constructed as described in Sec. 4.5.1. The interval Δx is defined by :

$$\Delta x = \Delta l \cdot \delta$$

where

Δl = DEBX,
$\delta =$ λ_{De}, electron Debye length, if NBOPAK = 1
λ_{Di}, first ion species Debye, if NBOPAK = 2
$\lambda_{Dtot} = 1 / (1/\lambda_{0i} + 1/\lambda_{De})$, if NBOPAK = 3
ρ_e, electron Larmor radius, if NBOPAK = 4
ρ_i, Larmor of first ion species, if NBOPAK = 5
$\rho_{tot} = 1 / (1/\rho_e + 1/\rho_i)$, if NBOPAK = 6

5. Necessary administrative parameters

5.1 Model, type of basis functions

The general type of the model is specified by the variable MODEL where:

if MODEL = 1 or 2, hot model is chosen,
if MODEL = 3 or 4, cold model is chosen,

and

if MODEL = 1 or 3, E_z component is taken into account,
if MODEL = 2 or 4, E_z is set to zero.

Then, the type of hot model (local or nonlocal) is specified by NLKREL and NLKRIO. if NLKREL and/or NLKRIO are .TRUE. then the respective contributions to the perturbed current are computed using the kernel formula (Eq. 4.10). Otherwise, we use the expanded formula, as in the code ISMENE [4] that is the local model. Note that if NLKRIO = .FALSE., then NLKREL is enforced to be .FALSE. in subroutine AUXVAL, as the local model can also be used for the electrons, since they have a smaller Larmor radius than the ions. In general, one always specifies either both NLKREL and NLKRIO to .TRUE. or both to .FALSE.. Note also that the variable NLGRAD, specifying whether the gradient of the equilibrium quantities are taken into account in the local model, is always set to .TRUE. (See Ref. 7).

The type of basis functions used is specified by the values of METHOD and NBASIS:

METHOD	NBASIS	Type of basis functions used
1	1	linears for E_x, E_y and E_z
1	2	Hermite cubics for E_x, E_y and E_z
3	1	piece-wise constant for E_x, linears for E_y and E_z
5	2	Quadratic for E_x and cubics for E_y and E_z

Note that only the third option (METHOD = 3 and NBASIS = 1) is compatible with the computation of the full integro-differential equation (nonlocal model). Thus, the other options are useful only when the ISMENE part of SEMAL is used (NLKRIO = .FALSE.).

There are still a few options which might be changed :

NLDIAG : (default .F.) If .T. then only the diagnostics are used in this run, without calculating a new solution. One uses an old solutions read on file "fort.9" in subroutine OREASOL.

NLBAND : (default .F.) If .T. then the matrix is considered as a band matrix. If only the local model is used, then NLBAND is always .TRUE.. If the nonlocal model is used (NLKRIO = .T.), then formally the matrix is full. However, if NLBAND = .T., one looks at the actual maximum bandwidth and solve the system using a band matrix solver. This saves time and is particularly useful when linpack subroutines are not available.

NLINPK : (default .T.) If .T. then Linpack subroutines are used in SOLVE to solve the linear system consisting of either a general full complex matrix or a complex band matrix. If .F. then home-made or Numerical Recipes "subroutines are used instead.

NLNAG: (default .T.) If .T. then NAGLIB subroutines are used in CBSPL1 and CBSPL2 for the cubic spline interpolation. Otherwise, Numerical Recipes subroutines are used.

5.2 Antenna, boundary conditions

The code can solve Eq. (4.10) successively for different antenna parameters. The total number of runs per job is specified by NRUN, and the different runs are indexed by NSTEP = 1, ..., NRUN. The variables which may change from one set of parameters to another (or from one run to another) have a dimension MDDATA.

The frequency can be determined in two ways :

1) If DLOMCI = 0.0 then :

$$\text{OMEGA} = 2\,\pi\,\text{FNOT}$$

That is, the antenna circular frequency ω, is determined by FNOT [Hz].

2) If DLOMCI ≠ 0.0 then :

$$\text{OMEGA} = (1 + \text{DLOMCI})\ \omega_{ci},$$

That is, ω is determined by its shift with the fundamental cyclotron frequency of the first ion species (AMASS(2), ACHARG(2)).

Thus, if one wants to specify ω with FNOT, one should not forget to set DLOMCI = 0 as well.

5.2.1 *Standard boundary conditions (antenna)*

These boundary conditions correspond to the set-up shown in Fig.1 and have been taken directly from the code ISMENE [4]. The location of the left and right walls are defined by XSHLOW and XSHUP, respectively. The solution in vacuum is calculated analytically as explained in Ref. 8 and summarized in Ref. 1 (Sec. 5.2.1). The antenna current has the following form :

$$\mathbf{j_a} = (J_y\,\mathbf{e}_y + J_z\mathbf{e}_z)\,\delta(x - x_a) + j_x\,H(x - x_a)\,\mathbf{e}_x$$

where $j_x = -\,i\,(k_yJ_y + k_zJ_z)$, assuming $\nabla\cdot\mathbf{j} = 0$ for simplicity,
$\mathbf{j_a} = \mathbf{j_a}\exp\{i(k_yy + k_zz - \omega t)\}$

$\delta(x - x_a)$ is the Dirac distribution and $H(x - x_a)$ the Heaviside function. The different parameters are specified as follows for the run numbered NSTEP :

x_a = XANTEN
k_y = ANTKY (NSTEP), (= WKY in the code)
k_z = ANTKZ (NSTEP), (=WKZ in the code)
$J_y = \beta\,k_z$, with β = ANTAMP (NSTEP), (= AMPLI in the code)
$J_z = -\,(1 - s)\,\beta\,k_y$, with s = ANTDIR (NSTEP), (= SANT in the code)

Note that one should also have NBNDRI = 2, which is the default value.

5.2.2 *Free waves condition*

If NBNDRI = 1, then the boundary conditions correspond to imposing free waves coming in and out at x = XPLEFT and XPRIGT. These conditions remain from the code ISMENE, but were never actually used in SEMAL. However, they might be of some interest for special studies, which is why we have left this option in the code SEMAL.

5.2.3 *Waveguide boundary conditions*

In this case, we do not assume any vacuum surrounding the plasma, but we impose the value of E_y and E_z directly at x = XPLEFT and XPRIGT. That is, we impose the tangential electric field at the plasma boundary. Assuming some value of (E_y, E_z) at the right-hand side, it simulates a waveguide which touches the plasma and has this (E_y, E_z) field just outside the waveguide mouth. Setting (E_y, E_z) = 0 at the left-hand side simulates a perfectly-conducting shell. In general, we have added some artificial damping (Sec. 7) near XPLEFT, simulating that the wave is totally absorbed at the left-hand side.

Note that by setting XPPRGT different from XPRIGT, we can impose a vacuum in between the plasma and the waveguide, as there are no particles left for x>XPPRGT+2ρ_i.

In connection with this type of boundary conditions, we calculate (if NLOTP2(10) = .T.) the following impedances at x_{pr} = XPRIGT :

$$Z_{yy} = \mu_0 E_y/B_y \text{ and } Z_{yz} = \mu_0 E_y/B_z$$
$$Z_{zy} = \mu_0 E_z/B_y \text{ and } Z_{zz} = \mu_0 E_z/B_z$$

Solving once with ($E_y(x_{pr})$ = 1., $E_z(x_{pr})$ = 0.) and the second time with (E_y = 0., E_z = 1.) gives all the components of the plasma impedance matrix.

The values of E_y, E_z for the run NSTEP are defined by :

E_y(XPLEFT) = EYBOUN (1, NSTEP), (≡ EYBRUN(1) in the code)
E_z(XPLEFT) = EZBOUN (1, NSTEP), (≡ EZBRUN(2) in the code)
E_y(XPRIGT) = EYBOUN (2, NSTEP), (≡ EYBRUN(1) in the code)
E_z(XPRIGT) = EZBOUN (2, NSTEP), (≡ EZBRUN(2) in the code)

6 Guiding center domain different from x domain

The density and temperature profiles are always given in terms of the guiding center coordinates, that is in terms of the x" variable. The x" domain, [x''_{left}, x''_{right}], is specified by [XPPLFT, XPPRGT], whereas the x domain is given by [XPLEFT, XPRIGT]. In general on takes the same domain for both as, if the density and temperature are very small near the edge with respect to their central values, the difference between the particle and guiding center profiles are negligible. However, there might be cases where one wants the x-edge to be really in vacuum, which is true only if XPRIGT > XPPRGT + 2ρ_i (or XPLEFT < XPPLFT - 2ρ_i), where ρ_i is the ion Larmor radius at XPPRGT or XPPLFT, respectively. Note that the plasma - vacuum interfaces are assumed to be located at x = XPRIGT and XPLEFT.

6.1 x" specifications

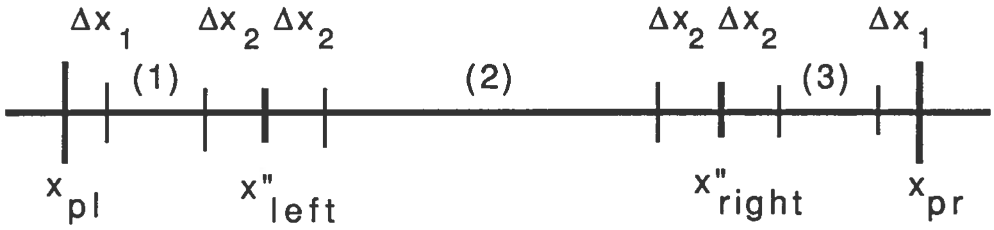
When XPPLFT ≠ XPLEFT and/or XPPRGT ≠ XPRIGT, the following parameters have to be specified :

LPPDIF = .T.,
XPPLFT [m], left x" boundary (left edge of guiding center profiles),
XPPRGT [m], right x" boundary (right edge of guiding center profiles).

Note that the profiles are still defined in the same way as described in Sec.4. Note also that in the local model, the profiles are given in terms of the particles, and this section is meaningless in this case.

6.2 Special packing

If LPPDIF = .T., the mesh is distributed in a special way, in order to resolve the density profile near x''_{left} and x''_{right} :



If LMESHB = .T., then NX/NXBOUN intervals are packed near x_{pl} and x_{pr} as described in Sec. 4.5.2 with :

$$\Delta x_1 = \Delta l\ \delta, \qquad \Delta l = \text{DEBX},$$

and where δ depends on the value of NBOPAK. Note that here the Debye length and the Larmor radius are calculated at $x = x''_{right}$.

In region (1) and (3), if they exist (as one can have $x_{pl} = x''_{left}$ or $x_{pr} = x''_{right}$), the mesh is equidistant, whereas in region (2) one may pack the mesh as described in Sec. 4.5.1. The number of intervals in each region (1), (2) and (3) are proportional to their relative widths.

Finally, there are always NX/NXPPBO intervals packed in each Δx_2 portion with :

$$\Delta x_2 = \Delta l\ \delta, \qquad \Delta l = \text{DEBXPP},$$

and where δ also depends on the value of NBOPAK and is the same as for Δx_1.

Afterwards, if LADAPP = .T. (as should always be the case), the x" mesh is adapted to the x mesh, but, of course, taken into account only the part inside [XPPLFT, XPPRGT]. This is done in subroutine MESHPP.

7. Artificial damping

There are different types of artificial damping, depending on which model (local or nonlocal) is used.

In the case of the local model, we have kept the same damping possibilities as in ISMENE : phenomenological or collision damping. The first one is activated if NLALDM = .T. and the second one if NLCOLL = .T.. The first one introduces an imaginary part in α_{xx}, $\alpha_{xx} = \alpha_{xx} - i{\cdot}DEL$, for the electrons only. The second one adds a term in γ_{zz} :

$$\gamma_{zz} = \gamma_{zz} + \frac{\omega_{pe}^2}{\omega^2} \frac{i\nu_{ei}}{\omega + i\nu_{ei}}$$

where

$\nu_{ei} = 2.9 \cdot 10^{-12}\, n_e\, \lambda\, Te^{-3/2} \cdot$ ANOMAL,
λ is the Coulomb logarithm.

In the case of the cold model, the phenomenological damping, i·DEL, is added to γ_{xx}, if NLALDM=.T..

In the case of the nonlocal model, the artificial damping is added to the whole diagonal of the dielectric tensor (if NLEDMP = .T.). In other terms, we add one term in Eq. (4.10) :

$$\frac{c^2}{\omega^2} \nabla \wedge \nabla \wedge \mathbf{E}\,(x) - (\,\underline{\underline{\varepsilon}}\,\mathbf{E}\,)\,(x) - i\,DEL\,\underline{\underline{I}}{\cdot}\mathbf{E}\,(x) = 0$$

In all the cases mentioned above using the variable DEL, the amplitude of the damping at the position x is calculated in the routine DAMPIN and has the following form :

$$DEL = DELNOT + \sum_{k=1}^{N_k} DL(k) \exp\left\{ - \left(\frac{x - x_{damp}(k)}{x_{width}(k)} \right)^2 \right\}$$

with $N_k \equiv$ NBDAMX (≤ 10) ,
DL(k) ≡ DELROT(k),
$x_{damp}(k) \equiv$ DAMPX0(k),
$x_{width}(k) \equiv$ DAMWDT(k).

8. Print/plot output control

The output on unit 6 (standard output) and the plots (on unit 8, file "fort.8") are all chosen and controlled in routine OUTPUT. The output is divided in four categories :

1) Initial output (Sec.8.1) : Output which does not need the solution (mesh, equilibrium quantities, dispersion relation, etc.).

2) Periodic output (Secs.8.2 and 8.3) : Output called each time the solution for a new set of parameters is found (i.e. each NSTEP=1,NRUN).

3) Final output (Sec.8.4) : Output called when all the NRUN solutions have been computed.

4) Special output (Sec.8.5) : Output called individually from different and sometimes exotic places in the code.

Two variables are used in different subroutines called by OUTPUT :

MXSPAC : (default = 10) Specifies the number of indices to skip when printing out the values of an array. This variable is very useful to limit the output size.

NLRDEQ : (default = .F.) If .T., then the values depending on x are printed on a reduced x mesh consisting of 51 equidistant points. This variable is mainly used in Sec.8.1.

8.1 Initial output, prints and plots

All the following routines are called when "OUTPUT (1)" is called in the code and if the corresponding NLOTP1 (i) or NLPL1 (i) are .TRUE. :

Initial prints:

if (NLOTP1 (1)) call ORUNPA : Prints a short output of the main characteristics of the particles and antenna parameters.
if (NLOTP1 (2)) call OMESH : Prints xpp and x meshes.
if (NLOTP1 (3)) call OEQUIL : Prints equilibrium quantities, profiles for each species.

if (NLOTP1 (4) call OPATEN : Prints the different part of the dielectric tensor as defined in the local model and for each species.

if (NLOTP1 (5)) call OTOTEN : As for OPATEN, but prints the sum over all species of α_{ij}, β_{ij} and γ_{ij}.

if (NLOTP1 (6)) call ODIREL : Prints the three solutions k_x^2 of the dispersion relation as obtained with the local model. Prints also $1/\Delta x^2$ in order to check the validity of the mesh.

Initial plots :

if (NLOPL1 (1)) call PLOTS : initialize the plot facility.

Note also that :

if NLOPL1 (1) is .FALSE. then no plots are plotted

The plot routines in the code SEMAL do not actually plot figures on a specific device, but they write the different values to be plotted on unit 8 (file "fort.8"). Then we use a different code, called CREAPICT, which can be run on a different computer than the code SEMAL and which has to be adapted to the local plot facilities. In Lausanne, we run this code on SUN or CRAY, using the UNIRAS plot package.

if (NLOPL1 (2)) call PDIREL : Plot real and imaginary parts of the three roots k_x of the dispersion relation, obtained with the local model, versus x. We use an "asinh-scale" on the y-axis in order to be able to distinguish the three roots which have very different values.

8.2 Periodic prints

All the following subroutines are called when "OUTPUT (2)" is called in the code and if the corresponding NLOTP2 (i) are true :

if (NLOTP2 (1)) call OSHORT : Short table of antenna current amplitude, and direction, k_y, k_z, FNOT and power emitted by antenna.

if (NLOTP2 (2)) call OFULLE (1)
if (NLOTP2 (3)) call OFULLE (2)
if (NLOTP2 (4)) call OFULLE (3)
} Print value of E_x, E_y or E_z respectively As well as derivative, phase, etc.

if (NLOTP2 (5)) call OMAGFI : Print value of perturbed magnetic field, E_+ and E_- polarizations.

if (NLOTP2 (6)) call OWLENG : Wavelength diagnostic calculated from E_x, E_y and E_z.

if (NLOTP2 (7)) call OPOWER : Print power emitted by antenna.

if (NLOTP2 (8)): if (.NOT.NLPOMX) call OPOSAB(NOUPOS)
if (NLPOMX) call OPOSMX(NOUPOS)

Print absorbed power per species, computed using the full formula or both full and 2nd order formula respectively.(See below for more explanations)

NOUPOS= 1: Print only the integrated power
2: Print also the power density

if (NLOTP2 (9)) call OABGAU : Print absorbed power per species, computed using the 2nd order formula.

If (NLOTP2(10) call OSURIM : Print B_y, B_z and surface impedance at XPRIGT and a few points before. A surface reflection coefficient is calculated, using the value of WGIMPD for the impedance just outside the plasma.

The subroutine OPOSAB calculates, on the XQ and XQGAUS meshes, the power absorbed by each species using the full nonlocal formula (Eq. (4.19)). The subroutine OABGAU, calculates on the same mesh the power absorbed by the different species using the expanded formula [5]. The option NLPOMX enables us to compute the power absorbed by some species using the 2nd order formula and by other species using the complete formula. That is, the subroutine OPOSMX calls OPOSAB and OABGAU.

The main parameters controlling the species and harmonics calculated with the complete nonlocal formula are :

NQRSPC : Number of species for which the absorbed power is calculated.

NQL1 (jspec) : First harmonic for species jspec taken into account in OPOSAB.

NQL2 (jspec) : Last harmonic for species jspec in OPOSAB.

Moreover, for each species jspec, one can specify over which interval $P_L(x)$, Eq.(4.19), needs to be calculated. The intervals are :

[QXLEFT(i, jspec), QXRIGT(i, jspec)], i = 1, NINTQX(jspec)

The number of intervals specified per species is NINTQX(jspec). If NINTQX = 0, the whole x domain is used and if NINTQX < 0, the power is not computed for this particular species.

The 2nd order power absorption is calculated in the same way as in ISMENE, except that it is on the mesh XQ, independant from the XX mesh. Thus, it always computes harmonic numbers from n = - 2 to n = + 2. In order to specify which species have to be computed using the 2nd order formula, we have introduced two new parameters :

NSEXPT : Total number of species to be computed in OABGAU

NSEXPI(j), j = 1 NSEXPT : Species for which power absorption is computed using the expanded formula.

The absorbed power density is then added to the array QDENS(ix, jspec, nharm) which depends on the mesh point (ix), species (jspec) and harmonic number (nharm). In order to decide which harmonics need to be taken into account in QDENS, we have to specify the following parameters :

NQL1TO(j), j = 1, NQRSPC : First harmonic for species j for which the absorbed power has to be calculated, using either the expanded formula (OABGAU) or the full formula (OPOSAB).

NQL2TO(j) : Last harmonic calculated.

In this way, the contribution from the harmonic n between - 2 and + 2, which are included in [NQL1TO, NQL2TO] for the given species, are added to QDENS.

As a summary, let us give an example corresponding to the case discussed in Sec. 3.2 : D-T JET plasma. In this case, we have three species (e^-, D, T), which absorb some energy at the 0th harmonic for the e^-, the 2nd for the D and the 3rd and 4th for the T. Thus, if we want to compute all their contributions using the full formula, we have to specify :

NLOTP2(8) = .T.,
NLOPMX = .F., NQRSPC = 3,
NQL1 = 0, 2, 3
NQL2 = 0, 2, 4,

If we want to save computer-time, noting that the e^- absorb energy everywhere in the plasma, the D only in x ∈ [- 0.1, 0.1] and the T only in x ∈ [0.1, 0.1] U [0.9, 1.10], we set :

NINTQX = 0, 1, 2
QXLEFT (1, 2) = - 0.10,   QXRIGT (1, 2) = 0.10,
QXLEFT (1, 3) = - 0.10, 0.90,   QXRIGT (1, 3) = 0.10, 0.90

If we want to print all the contributions using only the expanded formula, we have only to specify the following parameter:

NLOTP2(9) = .T.,

However, as the local model is limited to harmonic n ∊ [- 2, 2], it cannot take into account the tritium contribution. Assuming that the expanded formula holds for the electrons and the deuterium, the most rationalized way to calculate the absorbed power for each species is : to use the expanded formula for the e- and D and the full formula for the T, as the expanded formula is much faster to compute. In this case we have to specify the following :

NLOTP2(8) = .T.,
NLOPMX = .T., NQRSPC = 3,
NQL1TO = 0, 2, 3
NQL2TO = 0, 2, 4,
NSEXPT = 2,
NSEXPI = 1, 2,
NINTQX = - 1, - 1, 2,
QXLEFT (1, 3) = -0.10, 0.90, QXRIGT (1, 3) = 0.10, 1.10,
NQL1 (3) = 3,
NQL2 (3) = 4.

8.3 Periodic plots

All the following subroutines are called when "OUTPUT (2)" is called in the code and if the corresponding NLOPL2 (i) are true and if NLOPL1(1)=.T.:

If (NLOPL2 (1)) :   if (NLPFIE(i)) call PFIELD(i) , i = 1, 3
Plot of field **E**
If (NLOPL2 (2)) :   if (NLPFIE(i)) call PFIELD(i + 10) , i = 1, 3
Plot of 1st derivative of field **E**

The options NLPFIE (i) are the following :

NLPFIE(1) : Plot real and imaginary part of each component
NLPFIE(2) : As NLPFIE(1), but cut the NX/NXBOUN points near each edge of the x domain and plot the edges

separately. This is useful when LMESHB = .T. is specified, or if there is a large drop near the edge.

NLPFIE(3) : Plot norm and phase of each component.

If (NLOPL2(3)) call PPOSAB (NPLPOS) : Plot power absorption density and integrated power for each species and harmonic, according to array QDENS. Thus either only from full nonlocal formula (NLPOMX = .F.) either from both 2nd order and full formula (NLPOMX = .T.).

NPLPOS = 1 : Plot only power integrated and density with all species on one plot.

= 2 : AS NPLPOS = 1 and plot each species separately with the contribution of each of their harmonics.

If (NLOPL2(4)) call PFLABS : Plot fluxes and absorption as obtained with the 2nd harmonic formula, for all the species.

If (NLOPL2(5)) call PCFLAB : Compare 2nd order power and Poynting

If (NLOPL2(6)) call PQFLAB : Compare power in QDENS and Poynting.

8.4 Final output, prints and plots

All the following subroutines are called when "OUTPUT (3)" is called in the code and if the corresponding NLOTP3 (i) and NLOPL3 (i) are true :

Prints :

If (NLOTP3(1)) call OFULLE(13) : Full information on total E_x, E_y and E_z saved in array CSTO. The different solutions are added up with a weight equal to FACTOR(NSTEP).

If (NLOTP3(2)) call OPOWER : Print the power emitted by the antenna for each run, as well as the value of k_z and ω/ω_{ci}, and the total power. The total power is calculated using FACTOR(NSTEP).

Plots :

If (NLOPL3(1)) : If (NLPFIE(i)) call PFIELD (i + 20), i = 1, 3
Plot total field (CSTO), according to options NLPFIE(i).

If (NLOPL3(2)) : If (NLPFIE(i)) call PFIELD(i + 30), i = 1, 3
Plot 1st derivative of total field (CSPRTO) according to NLPFIE(i).

If (NLOPL3(3)) call PPOWTO : Plot power at antenna versus k_z and/or ω/ω_{ci}, depending on which parameter has changed in between different runs (different NSTEP values).

If (NLOPL1(1)) call PLOTF : close the plot facilities.

8.5 Special output

Different outputs are called individually and from anywhere in the code SEMAL. To find out from where a particular output is called, one should search for a "CALL OUTPUT(i)" in the code, for the corresponding value of i.

Called by :	Active if :	Subroutine called :	Comments
OUTPUT(4)	NLOTP4(4)	OMATRX	Print matrix with only kernel contribution included
OUTPUT(5)	NLOTP4(5)	OMATRX	Print total matrix (kernel + $\nabla\wedge\nabla\wedge$) and right-hand side
OUTPUT(6)	NLOTP4(6)	OCHECK	Check the solution. But this subroutine is not yet completed as it was not found necessary. One should save the matrix before inverting it.
OUTPUT(7)	NLOTP4(7)	OSAVE	Save labels, namelist, power and solution on unit NSAVE (=10, file "fort.10")

9. Quadratures, numerical parameters

There are a few parameters, starting with "N" and having a "G" in second or third position, which determine the number of Gaussian points per interval taken for the quadrature. They should all be set to 2, as it is the best value. The only exception is NGXPP, as mentioned before, which should be set to 1 (one point in middle of interval) if NXPP = NX and to 2 if NXPP = NX/2.

Other parameters deal only with the integration of the kernel. The first one is MINTKR, which determines whether the kernel is integrated using subroutines CONKRT (MINTKR =1) or CONKRX (MINTKR = 2). The first one is constructed such that the theta loop is the most inner loop, whereas the second has the loop over x as the most inner loop. In this way, when one is studying cases where the Larmor radii ρ are smaller then the mesh size Δx, one should use the first option (MINTKR = 1, default value) in order to maximize the length of the most inner loop (for vectoriel computers). And, if $\rho > \Delta x$, one should set MINTKR = 2.

The parameter MBYYKR (default value = 1) determines whether the special yy term is computed (MBYYKR $\neq$ 0) or not (MBYYKR = 0). The x'' mesh, needed for integrating this term, is the same as for the kernel if NXPPYY (default value = 1) is lower then 2 NXPP. It is doubled if NXPPYY = 2 NXPP. The value of NGXPPY (default value = 1) is chosen such that it is equal to 1 if NXPPYY $\geq$ NX, and to 2 if NXPPYY = NX/2.

The theta mesh, in computing the kernel, is an equidistant mesh with NTH2 intervals. NTH2 = 16 is a good value, it might be left to 8 (default value) to save some computer-time or to 32 to have a better converged solution.

The harmonics taken into account in the kernel are determined by the interval [NL1(jspec), NL2(jspec)] for each species jspec. The default value for all species is NL1 = - 10 and NL2 = + 10. If Bernstein waves are considered at third or more harmonic, it might be needed to enlarge this interval. Note that the loop over the harmonics does not take much time in computing the kernel.

There are a few parameters (CUTX, CUTX1, CUTX2, CUTXP, CUTPOS) which determine at which value y one may neglect exp{- y^2}. This allows to cut the interval to values $|y| <$ CUTX. The different CUTxxx parameters apply to different exponentials in the kernel or in the power absorption (CUTPOS). A good value for all of them is 4 (default value).

The XQ and XQGAUS meshes, on which the power absorption is computed are determined by NQX (number of intervals), NQGX (= 2, number of Gaussian points per interval) and NQXBO. If LMESHB = .T. then NQX/NQXB0 points are packed near the edges as described in Sec. 4.5.2.

The values of NQTH2, NQTP and NQXPP, specifying the number of θ, θ' and x'' intervals in computing the full power formula (Eq.4.19), have to be chosen correctly. In order to have a converged solution, one should have at least : NQTH2 = 32, NQTP = 8 and NQXPP = 32. However, one can reduce considerably the computer-time by reducing these values, without changing much the results. Thus, if one wants to know about how much power is absorbed in each species using the full formula, one might use : NQTH2 = 8, NQTP = 8 and NQXPP = 8. Then, if needed, one can take larger values of NQTH2 and NQXPP, noting that the computer-time is proportional to (NQTH2·NQTP·NQXPP)·NQX.

There is still one parameter to be mentioned here, which is NLINEA. It specifies whether one uses a linear (NLINEA = .T.) or a cubic (NLINEA = .F.) interpolation for the field in the full formula for the power absorption. Note however that NLINEA = .T. (default value), is as good as the other choice and takes less time, which is why we let it to the default value.

10. Input/output units

There are several units used by the code SEMAL. First the standard Fortran unit 5 (parameter NIN in SEMAL) for the input data (labels and namelist), and the unit 6 (NOUT) for the output results. The other units and filenames associated with them by OPEN statements in subroutine MASTER are the following:

Parameter in SEMAL	Unit	File name	Format	Purpose
NPUNCH	7		unformatted	Not used at moment. Free unit for intermediate write and read.
NPLOT	8	fort.8	formatted	Contains plot data
NREASO	9	fort.9	formatted	Unit from which the old namelist and solution are read (if NLDIAG = .T. or NLADSO = .T.).
NSAVE	10	fort.10	formatted	Unit on which the namelist and solution are saved (NLOTP4(7) = .T.).
NTAPKR	11	fort.11	unformatted	Unit on which the kernel matrix is saved (NLSAKR=.T.), and from which it is read when restarting the job (NLRES = .T.).
NTAPEF	13		unformatted	Not used at moment. Free unit for intermediate write and read.

NTAKZ1	15	unformatted	Unit used for computing several matrix with different k_z at once. The different matrices and right-hand sides are read/written from/on these units alternatively.
NTAKZ1- NTAKZS	14		

11. Miscellaneous namelist parameters

NLINPK
NLNAG
The code SEMAL was developed on a Cray-2 where both Linpack and Naglib subroutines are available, which is why we have used them from the beginning. However, if they are not available, then the corresponding parameter has to be set to .FALSE.. In this way, CRPP made (by K. Appert) or Numerical Recipes subroutine will be used instead.
Note that the Numerical Recipes routines solving the full general complex linear matrix system Ax = B are rather slow. Therefore, if NLINPK = .F., one should set NLBAND = .T..

NLSAKR If .TRUE. then the kernel matrix will be saved on unit NTAPKR (= 11, file "fort.11") from time to time, in such a way that it can be kept in a file and restarted later on (with NLRES = .T.). This is useful when only small portions of CPU-time are available, or when one changes only the boundary conditions or the damping, but not the kernel, for several runs.

NLRES If .TRUE., compute the kernel contribution starting from a kernel matrix saved on unit NTAPKR (= 11, file "fort.11").

NLKZSU If .TRUE., the solutions for several k_z are computed at once. Note that the parameter MDKZ has to be changed accordingly. if a computer with large memory is available, this might save much computer-time. On the Cray-2, MDKZ = 3 with MDX1 = 405 is about the maximum, that is computing 3 k_z at once.

NLADSO If .TRUE. then the solutions computed in the job are added up to the solutions calculated in a previous job, and read from unit NREASO (= 9, file "fort.9").

CPUMAX (default: 3550) Maximum CPU-time allowed for the job. If the job calculates solutions from different sets of parameters (NRUN > 1), and if there is not enough time to calculate the next step, then

the job will stop. This allows the job to terminate normally and not due to a system "cpu-time exceeded" break.
Another possibility is if NLSAKR = .T. : if there is not enough CPU-time to continue until the next save, then the job stops. Thus, if CPUMAX is chosen a little bit smaller than the actual CPU-time limit, then there is still time to save (or copy) the file fort.11.

12. Conventions and flow chart

The structure of the code SEMAL is based on the Olympus standard [9]. The subroutines are divided into several classes:

0	Control
1	Prologue
2	Calculation
3	Output
4	Individual output
5	Plot diagnostics
P	Actual plot subroutines
U	Utilities

All the subroutines are listed in alphabetical orders at the beginning of the code with their corresponding block name. For example, the subroutine CONMAT is numbered 2.2, which means that it is the second subroutine in class two. Its block name is C2S2 and it appears in the listing after all the C1Sxx subroutines and C2S1, and before C2S3. The common blocks are also numbered and all the variables in the common blocks are listed at the beginning of the code.

We have used the following conventions in naming the different variables in a subroutine:

- All the variables local to the subroutine have a name which starts with the letter "Z" (Zxxx).
- The real variables which are passed as arguments of the subroutine are named Pxxx. The integer arguments start with the letter K.
- The letters I, J, and L are used for integers and, in general, the variables Ixxx and Jxxx are used for indices of a DO loop.
- The letters A-H, O, Q-Y are used for real variables and L-N for integers, which are passed to the subroutine via a common block.
- The variables named NLxxx are in general logical.
- The variables named ZCxxx are in general complex.
- The variables named Cxxx are in a common and are in general complex.

- The subroutines are divided in sections (1, 2, ...) and in subsections (1.1, 1.2,.., 2.1, ...). In general, the labels 100 to 199 are used in Sec.1, 200 to 299 in Sec.2, etc. For example, a label such as 142 (or 1420) is used either in subsection 1.4, or in 1.4.2.

We list below the flow chart of the different subroutines called by SEMAL when it is run with an empty namelist, as in the example given in Sec.3.1. In this case, the perturbed current is calculated using only the local model:

```
SEMAL
    MASTER
        DAYTIM
        COTROL
            MESAGE
            LABREA
            CLEAR
            |   RESETR
            PRESET
            READAT
            AUXVAL
            |   MESH
            |       DENSIT
            |       TEMPER
            |       BTOROI
            |       GAUSS
            |   MESHPW
            |       GAUSS
            |       ISRCHFGT
            |   MESHPP
            |   BASIST
            WRIDAT
            |   DEFLAB
            |   BLINES
            BLINES
            OUTPUT(1)
            |   ORUNPA
            |       BTOROI
            |       DENSIT
            |       TEMPER
            |   OMESH
            |   OEQUIL
            |       BTOROI
            |       DENSIT
            |       TEMPER
```

```
STEPON
|   NEWMOD
|   DEFLAB
|   CONMAT
|   .   CVERO
|   .   CONEXP
|   .       CVERO
|   .       EPSPEC
|   .           FUNPAR
|   .               DENSIT
|   .               TEMPER
|   .               BTOROI
|   .               DISPFN
|   .                   ERROR
|   .                       BESSEL
|   .           CURHOM
|   .           CURINH
|   .   CONROT
|   .       CVERO
|   VACUUM
|   BOUND
|       AWAY
|   SOLVE
|   POWER
|       PRIMES
|   ADDMOD
|   OUTPUT(2)
|       OSHORT
|       OFULLE
TESEND
|   BLINES
|   MESAGE
OUTPUT(3)
ENDRUN
|   BLINES
|   MESAGE
|   DAYTIM
|   RUNTIM
```

If we run the code with NLKREL=.T. and NLKRIO=.T., then the flow chart of the subroutine CONMAT will look like the following:

```
|    CONMAT
|    .    CVERO
|    .    CONKRT
|    .         GAUSS
|    .         DENSIT
|    .         TEMPER
|    .         BTOROI
|    .         DISPFN
|    .              ERROR
|    .                   BESSEL
|    .    CONBYY
|    .         GAUSS
|    .         DENSIT
|    .         TEMPER
|    .         BTOROI
|    .    CONROT
|    .         CVERO
|    .    COPBAKR
```

REFERENCES

[1] Sauter O., Nonlocal Analyses of Electrostatic and Electromagnetic Waves in Hot Magnetized Nonuniform Bounded Plasmas, Ph. D. thesis No. 1000, Ecole Polytechnique Fédérale de Lausanne, Lausanne (1992), and Lausanne Report LRP 457/92 (1992).

[2] Fried B. D., Conte S. D., The Plasma Dispersion Function, Academic, New York, 1961.

[3] Vaclavik J. and Appert K., Plasma Phys. Controlled Fusion 29 (1987) 257.

[4] Appert K., Hellsten T., Vaclavik J. and Villard L., Comput. Phys. Commun. 40 (1986a) 73.

[5] Jaun A. and Vaclavik J., in Theory of Fusion Plasmas (Proc. Joint Varenna-Lausanne Int. Workshop, Varenna, 1990), Editrice Compositori, Bologna (1990) 461.

[6] Press W. H., Flannery B. P., Teukolsky S. A., Vetterling W. T., Numerical Recipes (Fortran Version), Cambridge University Press 1986, Cambridge (1990).

[7] Jaun A., Vaclavik J. and Appert K., Plasma Phys. Controlled Fusion 33 (1991) 521.

[8] Vaclavik J. and Appert K., Nucl. Fusion 31 (1991) 1945.

[9] Keith Roberts, Comput. Phys. Commun. 7 (1974) 237 and 245 ; 9 (1975) 51 ; 10 (1975) 167 ; 29 (1983) 7, 15, 45 and 59.

Appendix A

Variables in namelist NEWRUN

The variables in the namelist are listed below, with a small comment. The last column refers to the subsection of this documentation where the variable is explained. The third column gives the type of the variable (Integer, Real, Complex, Logical), specifies whether it is an array (A) and gives the block number of the common block in which they are defined.

ABT(6)	*Values defining cubic toroidal magnetic field	RA 2.2	4.4
ACHARG(MDSPEC)	*Charge number per species	RA 2.3	4.1
ADN(6)	*Values defining cubic density profiles	RA 2.2	4.4
AMASS(MDSPEC)	*Mass number per species	RA 2.3	4.1
ANOMAL	*Factor multiplying collision damping coeff.	R 4.1	7
ANTAMP(MDDATA)	*Amplitude of antenna currents [A/m**2]	CA 2.5	5.2.1
ANTDIR(MDDATA)	*Antenna directions (S-numbers)	CA 2.5	5.2.1
ANTKY(MDDATA)	*Y components of wave vectors	RA 2.5	5.2.1
ANTKZ(MDDATA)	*Parallel components of wave vectors	RA 2.5	5.2.1
ATP(6,MDSPEC)	*Values defining cubic temperature profiles	RA 2.2	4.4
AXPLAC(10)	*Positions for x packing	RA 3.1	4.5.1
AXWIDT(10)	*Widths for x packing	RA 3.1	4.5.1
BNOT	*Toroidal magnetic field at x=0	R 2.2	4.1
CENDEN(MDSPEC)	*Central densities [1/m**3]	RA 2.2	4.1
CENTMP(MDSPEC)	*Central temperature [eV]	RA 2.2	4.1
CPUMAX	*Maximum CPU-time for the job before stopping	R 1.1	11
CUTPOS	*Width of nonlocality in computing power absorption	R 3.1	9
CUTX	*Width of nonlocality in computing kernel	R 3.1	9
CUTX1	*As CUTX, but imposed on different parameters	R 3.1	9
CUTX2	*As CUTX1	R 3.1	9
CUTXP	*As CUTX1	R 3.1	9
DAMPX0(10)	*Position of special damping points	RA 2.2	7
DAMWDT(10)	*E-fold constants for damping profile	RA 2.2	7
DEBX	*Number of debye lengths for edge packing	R 3.1	4.5.2
DEBXPP	*Number of debye lengths for xpp edge packing	R 3.1	6.2
DELNOT	*Constant amplitude of artif. damping profile	R 2.2	7
DELROT(10)	*Local amplitudes for artificial damping profile	RA 2.2	7
DELTAS	*Temperature e-fold (if NLCESA or NLCHIU)	R 2.2	4.2
DELTS2	*Temperature e-fold constant in DII-D (if NLCHIU)	R 2.2	4.3
DENLIM	*Density at limiter (if NLCESA or NLCHIU)	R 2.2	4.2
DENSHI	*Density at Faraday shield in DIII-D (if NLCHIU)	R 2.2	4.3
DENWN2	*Density at RLIMIT+WN2 in DIII-D (if NLCHIU)	R 2.2	4.3
DLOMCI(MDDATA)	*Applied shift from fundamental frequency	RA 2.5	5.2
EDGPRO	*X coordinate of edge of profiles	R 2.2	4.1
EQUDEN(MDSPEC)	*Constants defining edge values of densities	RA 2.2	4.1
EQUTMP(MDSPEC)	*Constants defining edge values of temperatures	RA 2.2	4.1
EXPDEN(MDSPEC)	*Exponants determining density profiles	RA 2.2	4.1
EXPTMP(MDSPEC)	*Exponants determining temperature profiles	RA 2.2	4.1

EYBOUN(2,MDDATA)	*Applied E-y at XPLEFT and XPRIGT (if model=5)	CA 2.5	5.2.3
EZBOUN(2,MDDATA)	*Applied E-z at XPLEFT and XPRIGT (if model=5)	CA 2.5	5.2.3
FACTOR(MDDATA)	*Form factor for adding different solutions	CA 2.5	8.4
FNOT(MDDATA)	*Applied linear frequency [1/sec] (if DLOMCI=0)	RA 2.5	5.2
LMESHB	*T: Pack mesh very near edges	L 3.1	4.5.2
LPPDIF	*T: xpp domain differ from x domain	L 3.1	6.1
MBYYKR	*Determines if special yy term is computed	I 4.1	9
METHOD	*Determines choice of basis functions	I 4.1	5.1
MINTKR	*Determines method for integrating the kernel	I 4.1	9
MODEL	*Determines the physical model used	I 4.1	5.1
MXSPAC	*Index step in XX between successive output values	I 4.1	8
NBASIS	*Number of basis function	I 3.1	5.1
NBDAMX	*Number of points for the artificial damping profile	I 2.2	7
NBNDRI	*Determines applied boundary conditions	I 4.1	5.2.1
NBOPAK	*Determine scale of edge packing (if LMESHB)	I 3.1	4.5.2
NGTH	*Number of theta Gaussian points	I 3.1	9
NGX	*Number of x Gaussian points for quadrature formula	I 3.1	9
NGXPP	*Number of xpp Gaussian points	I 3.1	4.5.1
NGXPPY	*Number of xpp Gaussian points for special yy term	I 3.1	9
NIN	**Unit for input datas (=5)	I 1.1	10
NINTQX(MDSPEC)	*Nb. of intervals in which power should be computed	IA 3.1	8.2
NL1(MDSPEC)	*First harmonic number to be computed in kernel	IA 3.1	9
NL2(MDSPEC)	*Last harmonic evaluated in kernel per species	IA 3.1	9
NLADSO	*T: Add up solutions from previous job	L 1.1	11
NLALDM	*T: Damping in ALPHXX with local model	L 4.1	7
NLBAND	*T: Compute matrix as a band matrix	L 4.1	5.1
NLCESA	*T: Use Cesario (FTU) type of profiles	L 2.2	4.2
NLCHIU	*T: Use Chiu (DIII-D) type of profiles	L 2.2	4.3
NLCOLL	*T: Collisional damping with local model	L 4.1	7
NLCUB	*T: Use cubics to define profiles	L 2.2	4.4
NLDIAG	*T: run only diagnostics using old solution	L 1.1	5.1
NLEDMP	*T: Diagonal artificial damping included	L 4.1	7
NLHFS	*T/F: High/Low Field Side antenna	L 4.1	4.1
NLINEA	*T/F: Linear/Cubic interp. in comput. power density	L 3.1	9
NLINPK	*T: use Linpack routines to solve linear system	L 1.1	5.1
NLKREL	*T: Compute electrons in kernel (nonlocal formula)	L 4.1	5.1
NLKRIO	*T: Compute ions in kernel (F:use local model)	L 4.1	5.1
NLKZSU	*T: Compute kernels for several kz at once	L 4.1	11
NLNAG	*T: use Naglib. F: use equiv. Num. Recipes routines	L 1.1	5.1
NLOPL1(MDOUT)	*Initial plot control	LA 4.1	8.1
NLOPL2(MDOUT)	*Periodic plot control	LA 4.1	8.3
NLOPL3(MDOUT)	*Final plot control	LA 4.1	8.4
NLOTP1(MDOUT)	*Initial output control	LA 4.1	8.1
NLOTP2(MDOUT)	*Periodic output control	LA 4.1	8.2
NLOTP3(MDOUT)	*Final output control	LA 4.1	8.4
NLOTP4(MDOUT)	*Special individual output control	LA 4.1	8.5
NLPFIE(10)	*Plot of fields additional control	LA 4.1	8.3
NLPOMX	*T:Compute power using expanded and nonlocal formulae	L 4.1	8.2
NLRDEQ	*T: Reduced mesh for equilibrium output	L 4.1	8

NLRES	*T: Restart from saved intermed./final kernel matrix	L 1.1	11
NLSAKR	*T: Save intermediate/final kernel matrix	L 1.1	11
NLTOKA	*T: Tokamak geometry: XPLEFT(RIGT)=RMAJOR-(+)RMINOR	L 4.1	4.1
NMESHX	*If 1, pack x mesh according to NPOIDX, SOLPDX,AXPLAC	I 3.1	4.5.1
NOUPOS	*Output option for power absorption	I 4.1	8.2
NOUT	**Tape unit for output datas (=6)	I 1.1	10
NPLOT	**Unit for plot datas used in PLOTDEC (=8)	I 1.1	10
NPLPOS	*Plot option for power absorp.	I 4.1	8.3
NPOIDX	*Number of points around which x mesh is packed	I 3.1	4.5.1
NQGTH	*Nb of Gaussian points for theta in power absorption	I 3.1	9
NQGTP	*Number of theta-prime Gaussian points in power	I 3.1	9
NQGX	*Number of Gaussian points for x in power absorption	I 3.1	9
NQGXPP	*Number of Gaussian points for xpp in power absorption	I 3.1	9
NQL1(MDSPEC)	*First harmonic evaluated in power absorption	IA 3.1	8.2
NQL1TO(MDSPEC)	*Power 1st harm. with expanded or nonlocal formula	IA 3.1	8.2
NQL2(MDSPEC)	*Last harmonic number in power	IA 3.1	8.2
NQL2TO(MDSPEC)	*Last harmonic. If .NOT.NLPOMX: NQLiTO = NQLi, i=1,2	IA 3.1	8.2
NQRSPC	*Nb of species computed in nonlocal absorption density	I 3.1	8.2
NQTH2	*Number of theta mesh intervals for power	I 3.1	9
NQTP	*Number of theta-prime interval for power absorption	I 3.1	9
NQX	*Number of x intervals for power density	I 3.1	9
NQXBO	*NQX/NQXBO=nb of points for edge packing in power	I 3.1	9
NQXPP	*Number of xpp interval for power absorption	I 3.1	9
NRSPEC	*Total number of species (electron+ions)	I 3.1	4.1
NRUN	**Number of sets of parameters to run in the job	I 1.1	5.2
NSEXPI(MDSPEC)	*Species using expanded formula for power if NLPOMX	IA 3.1	8.2
NSEXPT	*Number of species given in NSEXPI: NSEXPI(1..NSEXPT)	I 3.1	8.2
NTH2	*Number of theta intervals in kernel	I 3.1	9
NX	*Number of x intervals	I 3.1	4.5.1
NXBOUN	*NX/NXBOUN=nb of points packed near edges (if LMESHB)	I 3.1	4.5.2
NXPP	*Number of xpp intervals	I 3.1	4.5.1
NXPPBO	*NX/NXPPBO points for xpp edge packing (if LPPDIF)	I 3.1	6.2
NXPPYY	*Number of xpp intervals for special yy term	I 3.1	9
QXLEFT(MDLTOT/2, MDSPEC)	*Power comput. in [QXLEFT(i,.),QXRIGT(i,.)], i=1,NINTQX	RA 3.1	8.2
QXRIGT(MDLTOT/2, MDSPEC)	*If NINTQX: =0 whole domain, <0 species not computed	RA 3.1	8.2
RLIMIT	*Limiter coordinate (if NLCESA or NLCHIU)	R 2.2	4.2

RMAJOR	*Major radius [m]	R 2.4	4.1
RMINOR	*Minor radius [m]	R 2.4	4.1
SLAMDA	*Density e-fold in scrape-off layer (NLCESA or NLCHIU)	R 2.2	4.2
SOLPDX	*Fraction of points left for uniform background x mesh	R 3.1	4.5.1
TEMLIM	*Temperature at limiter (if NLCESA or NLCHIU)	R 2.2	4.2
WDGAP	*Width of e-fold in DIII-D (if NLCHIU)	R 2.2	4.3
WDSHI	*Width of Faraday shield in DIII-D (if NLCHIU)	R 2.2	4.3
WGIMPD	*Waveguide impedance at XPRIGT for field reflection coeff.	R 2.5	8.2
WN2	*Width of linear drop of DII-D density (if NLCHIU)	R 2.2	4.3
WT2	*Width of first e-fold DIII-D temperature (if NLCHIU)	R 2.2	4.3
XANTEN	*Position of antenna [m]	R 2.5	5.2.1
XPLEFT	*Left x boundary [m]	R 3.1	4.1
XPPLFT	*Left xpp boundary [m], if LPPDIF	R 3.1	6.1
XPPRGT	*Right xpp boundary [m], if LPPDIF	R 3.1	6.1
XPRIGT	*Right x boundary [m]	R 3.1	4.1
XSHLOW	*Inner wall coordinate [m]	R 2.4	5.2.1
XSHUP	*Outer wall coordinate [m]	R 2.4	5.2.1

Appendix B

Dispersion relation valid to all orders in Larmor radii

code DISPAL

In parallel to the development of the code SEMAL, we have constructed a code which solves the general dispersion relation, valid to all orders in Larmor radii. It uses exactly the same input parameters as the code SEMAL and finds solution(s) k_x at different position x. It solves the real part of the dispersion relation, obtained with the plasma parameters evaluated in x, and searches only for a real root k_x. As it solves the exact dispersion relation, i.e. without a polynomial approximation of it, there is not a definite number of roots. Some examples of dispersion relations obtained with DISPAL are shown in Appendix C of Ref.[1]. The dispersion relation valid up to second order in Larmor radii is obtained in SEMAL by setting the option NLOTP1(6)=.T. and, if the plot is desired, NLOPL1(1 and 2)=.T.

We describe here in detail the procedure used to obtain Fig.B.1, which is the same as Fig.C.5 of Ref.[1]. It gives the dispersion relation of the O and X modes near $\omega = |\omega_{ce}|$ and $\omega = \omega_{uH}$ (upper-hybrid frequency). The parameters correspond to the TCV tokamak in Lausanne (Tokamak à configurations variables) with a deuterium plasma. The input data are the following:

```
Dispersion relation of O and X modes
 f = 39 GHz, TCV parameters
$NEWRUN
BNOT=1.43,
CENDEN(2)= 1.0E+19,
CENTMP= 1500., 1500.,
NRSPEC= 2,
AMASS(2)= 2., ACHARG(2)= 1.,
WKY= 0.0, WKZ= 140.,
DLOMCI= 0.0,
FNOT= 3.9E+10,
RMAJOR= 0.87,
EDGPRO= 0.24,
NX= 10,
NK= 60,
XPLEFT= -0.24,      XPRIGT= 0.24,
AKMIN2= 1.0,        AKMAX2= 1.0E+09,
$END
```

These parameters are saved in file "data.B1". Note that the variables ANTKY and ANTKZ of SEMAL have been changed to WKY and WKZ, respectively. The variables AKMIN2 and AKMAX2 determine the interval in which a root k_x^2 is searched. DISPAL creates three files: "output" (unit 6), "fort.7" (unit 7) and "fort.8" (unit 8). The first one contains the output, consisting of a list of the equilibrium parameters and of the roots found. The file "fort.8" contains plot data for plotting each root individually and for a plot of all the roots together (using fort.8 as input to the code CREAPICT, as mentioned in Sec.8.1).

The file "fort.7" contains the labels, namelist and the coordinates:

$$(x \, , \, \mathrm{sign}[k_x^2] \sqrt{|k_x^2|})$$

of all the roots found for each mode. This file can be used as input file for the program READPOINTS, which reads all the coordinates (on unit 7) and plots all the modes on one plot of k_x versus x.

As an example, we show on Fig.B.2 the plot obtained after having run successively DISPAL, with "data.B1" as input, READPOINTS, with "fort.7" as input, and CREAPICT, with the resulting file "fort.8" as input. Then what has to be done, in order to obtain the complete dispersion relation shown in Fig.B.1, is to shorten the x interval such as to resolve the details of the dispersion relation. The k_x^2 interval has also to be changed. For example, to obtain the roots near x=-0.24 m, where they are very close to one another, we have modified the following parameters:

```
XPLEFT= -0.24,          XPRIGT= -0.192,
AKMIN2= 3.8E+05,        AKMAX2= 1.0E+06,
```

This is done in file "data.B2". Before running DISPAL with "data.B2", one has to copy the file "fort.7" to another file, say "fort.7.figB1". Otherwise, we loose the roots found with "data.B1" as input file. Then, we run DISPAL with "data.B2" and obtain the roots for $x \in [-0.24,-0.192]$. These roots are listed in file "fort.7" and can be copied in the editor and added to the file "fort.7.figB1". In this way, this file contains the roots found with both "data.B1" and "data.B2" as input file.

Repeating this procedure with the files "data.B3", "data.B4",..., and "data.B12" gives all the points plotted on Fig.B.1. However, one must be careful to attribute the different roots to the correct branch. We have numbered the branches 1, 2, 3 and 4 in this case. Once all the points are introduced in the file "fort.7.figB1", we copy this file onto file "fort.7" and use it as input to the code

READPOINTS. This produces the plot shown in Fig.B.1 and also creates a file "fort.9" (unit 9), which is equivalent to fort.7 but where all the points belonging to the same branch are put together and sorted in increasing value of x. The file "fort.9" has been copied onto the file "fort.9.figB1".

Note that two changes have still been made to the file "fort.9.figB1". First, we have added a zero root at x=0.024, (0.024, 0.0), to the second mode. This tells the code READPOINTS that the second mode is separated into two parts: x<0.024 and x>0.024. In this way, the two lowest lines (dashed) at the left and right of Fig.B.1 are not joined by a spurious line. Secondly, we have added the following lines at the end of the file "fort.9.figB1":

```
X RESONANCE
     1
     0.02296
```

In this way, READPOINTS will draw a vertical line at x=0.02296, which corresponds to the first harmonic of the electron cyclotron frequency. In order to draw a vertical line alo at $\omega = \omega_{uH}$ (which lies at x=0.1708), we have specified:

```
X RESONANCE
     2
     0.02296    0.1708
```

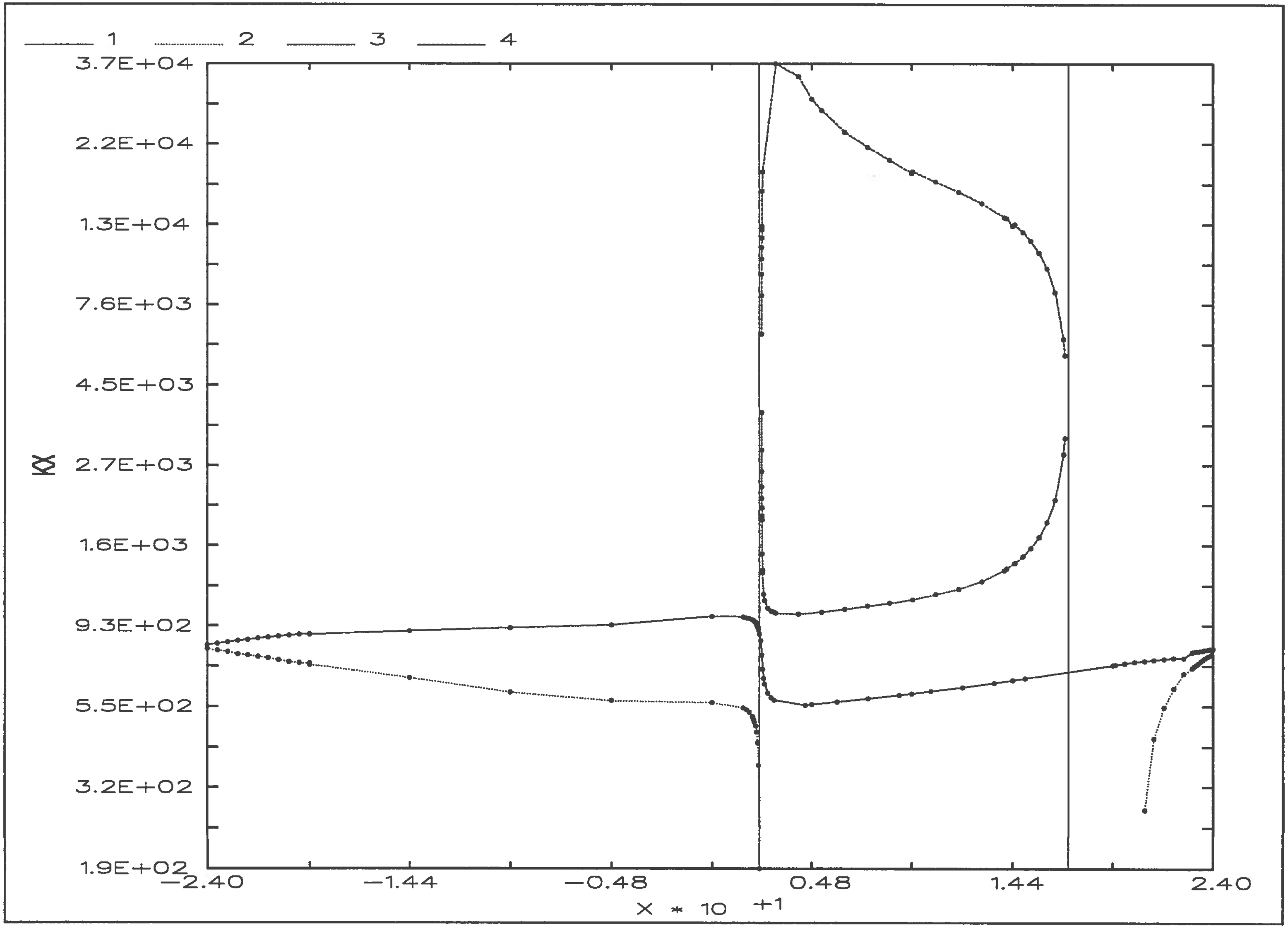
Note that the files "fort.7.figB1", "fort.9.figB1" and "fort.9" can be copied at any time onto the file "fort.7" and used as input to the code READPOINTS. To obtain the plots, we have copied the file "fort.8" onto the file "fort.13" (unit 13), which is the input file for the code CREAPICT. Finally, note that the y-axis of the dispersion relation plots has an "asinh" scaling.

The procedure is really difficult to explain in words, but it enables one to search for roots in different x and k_x^2 intervals. This is very useful to be able to resolve the fine structure around resonances, conversion or turning points without consuming too much CPU-time. One should keep NX≤10 and NK≤60 and change the x and k_x^2 domains in which the dispersion relation is solved, rather than increasing NX and/or NK.

dispersion relation for O and X modes

f = 39 GHz, TCV parameters

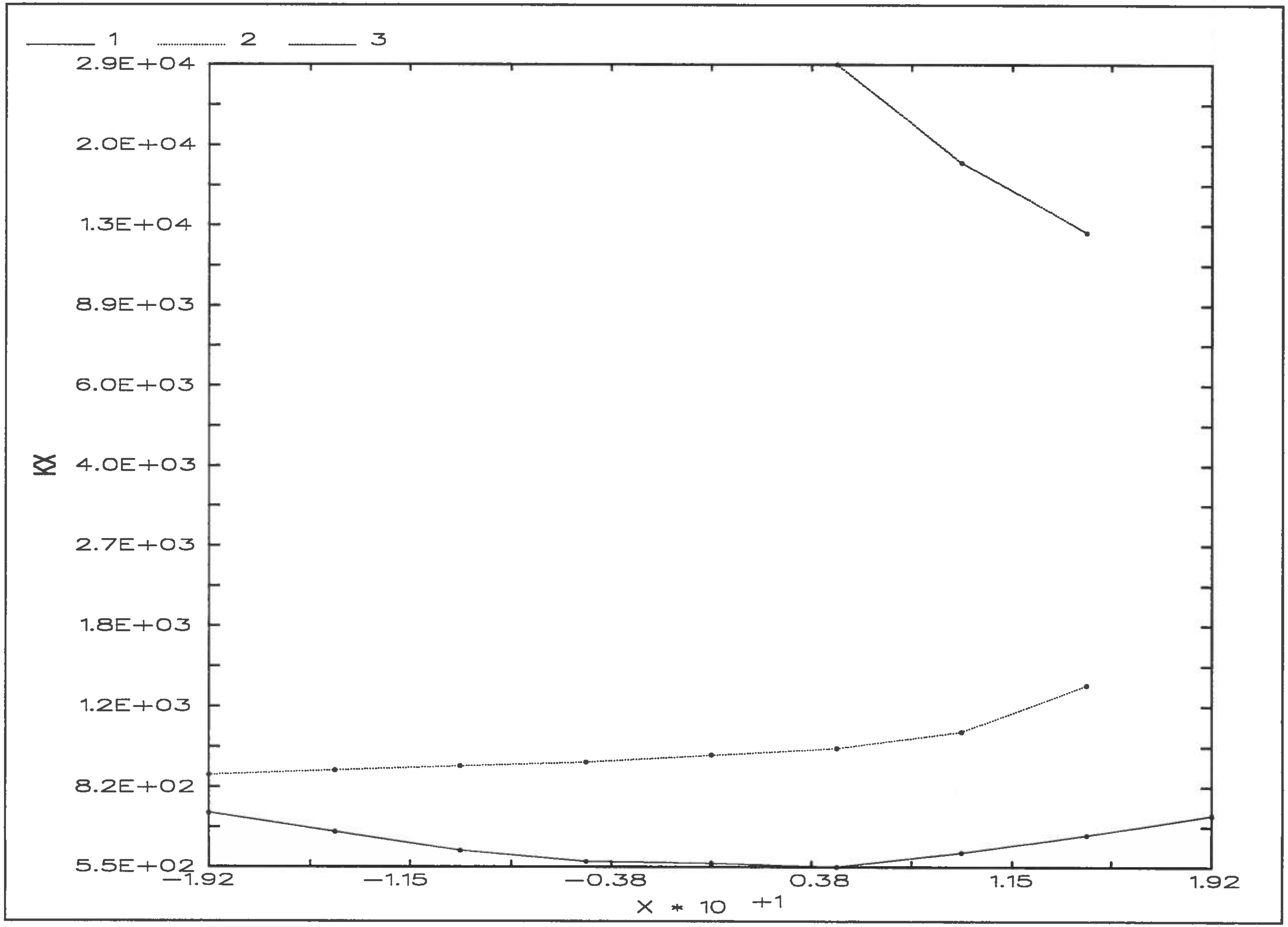
FIGURE B.1



dispersion relation for O and X modes

f = 39 GHz, TCV parameters

FIGURE B.2



Appendix C

CREAPICT : Plot of data

In this Appendix, we explain how the data saved on files "fort.8" (unit 8), by SEMAL, DISPAL or READPOINTS, are plotted on a particular device (screen, postscript file, etc.). The input file for CREAPICT is "fort.13" (unit 13), thus one has to copy the file "fort.8" onto the file "fort.13".

The structure of the file "fort.13" is rather simple. First it contains the name of a subroutine and then the arguments needed for the particular subroutine. Thus, CREAPICT reads first the name of a routine, then the corresponding arguments and finally calls the subroutine with the now known arguments. This enables us also to easily modify the file "fort.13" in an editor if needed.

The first line of "fort.13" should always contain the string 'PLOTS' at the beginning of the line (the second lines contains three numbers, one integer and two reals, but they are not used anymore in CREAPICT). The call to PLOTS generates calls to initialisation UNIRAS subroutines. The last line of "fort.13" should contain the string "PLOTF" at the beginning, which specifies that the plots are finished and that closing UNIRAS subroutines have to be called.

The UNIRAS subroutines called for initialisation, closure or separation of the plots into different segments (pages) are the following:

GOPEN :	OPEN UNIRAS
GCLOSE :	CLOSE UNIRAS
GROUTE('SEL driver ; EXIT') :	CHOOSE PLOT DRIVER
GSEGCR(IPAGE) :	OPEN SEGMENT (AND CREATE FILE UNIPICT)
GSEGCL(IPAGE) :	CLOSE SEGMENT
GCLEAR :	JUMP TO NEXT PAGE ON OUTPUT DEVICE

These calls have to be changed accordingly to the plot package used on the local computer system. Note that with UNIRAS, segments can be created and then plotted, with the help of the code READPICT, in whatever order, with different segments on the same page, and on whatever plot device.

The code CREAPICT contains also many home-made subroutines plotting 1-D graphes with different set-ups and scalings. All these routines use only six

UNIRAS subroutines, which are:

GVECT(X,Y,N) : Plots a polyline specified by (X(i),Y(i)),i=1,N.

GDOT (X,Y,N) : Same as GVECT, but plot dots at each point instead of joining the points with a line.

GDASH(J) : Specifies which type of line is used by routine GVECT: J=0, continuous line, =1, dotted line, etc.

GWICOL(-SIZE,1) : Specifies the thickness, SIZE, of the line or the dots drawn by GVECT or GDOT. In general, SIZE= 1.0 or 2.0.

GCHAR(TEXT,X,Y,HCHAR) : Draws a character string TEXT at coordinate (X,Y) and with a character height of HCHAR.

GCHARA(IANGLE) : Specifies at which angle IANGLE (in degrees) the character string is plotted by GCHAR.

Thus, in order to create an interface in between the UNIRAS package and the plot package used on a new computer system, one has only to create these six subroutines, such that they call the corresponding routines of the new plot package.