ASTRA
An Automatic System for Transport Analysis in a Tokamak

G.V. Pereverzev*, P.N. Yushmanov, A.Yu. Dnestrovskii
A.R. Polevoi, K.N. Tarasjan, L.E. Zakharov

I.V. Kurchatov Institute of Atomic Energy
Moscow, USSR

*Presently at Max-Planck-Institut für Plasmaphysik
D-8046 Garching, FRG

IPP 5/42 August 1991

MAX-PLANCK-INSTITUT FÜR PLASMAPHYSIK

8046 GARCHING BEI MÜNCHEN

MAX-PLANCK-INSTITUT FÜR PLASMAPHYSIK

GARCHING BEI MÜNCHEN

ASTRA
An Automatic System for Transport Analysis in a Tokamak

G.V. Pereverzev*, P.N. Yushmanov, A.Yu. Dnestrovskii
A.R. Polevoi, K.N. Tarasjan, L.E. Zakharov

I.V. Kurchatov Institute of Atomic Energy
Moscow, USSR

*Presently at Max-Planck-Institut für Plasmaphysik
D-8046 Garching, FRG

IPP 5/42 August 1991

Part of the work was performed during a stay of one of the authors (G.V. Pereverzev) at Max-Planck-Institut für Plasmaphysik

Die nachstehende Arbeit wurde im Rahmen des Vertrages zwischen dem Max-Planck-Institut für Plasmaphysik und der Europäischen Atomgemeinschaft über die Zusammenarbeit auf dem Gebiete der Plasmaphysik durchgeführt.

Abstract

The set of codes described here - *ASTRA* (Automatic System of TRansport Analysis) - is a flexible and effective tool for the study of transport mechanisms in reactor-oriented facilities of the tokamak type. Flexibility is provided within the ASTRA system by a wide choice of standard relationships, functions and subroutines representing various transport coefficients, methods of auxiliary heating and other physical processes in the tokamak plasma, as well as by the possibility of pre-setting transport equations and variables for data output in a simple and conseptually transparent form. The transport code produced by the ASTRA system provides an adequate representation of the discharges for present experimental conditions.

INTRODUCTION

In 1987, the Kurchatov Institute of Atomic Energy developed TRANSITOR [1], a system for automatic calculation and analysis of transport processes in a tokamak. TRANSITOR had a large set of standard functions, formulas, and subroutines for describing various transport coefficients, auxiliary-heating methods and other physical processes. This provided high flexibility and effectiveness for studies of physical transport processes. A prominent feature of the TRANSITOR system is the availability of service programs which permit the set-up mode of input transport equations and output variables in a simple and convenient form. A specific advantage is that the input text is as close as possible to the usual way of writing of formulas in scientific publications.

In the last few years, the TRANSITOR code was developed further, and it formed the basis for creating ASTRA, a system for automatic transport analysis. While retaining the most important properties of the TRANSITOR code - library of formulas for plasma processes, library of experimental discharges, and advanced service utilities, the new code permits a more adequate description of discharges for present experimental conditions, achieved by a significant increase in the number of physical processes and characteristics that can be included. Thus, the ASTRA system provides the possibility of considering tokamaks with non-circular cross section, calculation of heating and current generation, calculation of the current-rise dynamics accompanied by changes in the plasma shape, adiabatic compression, self-consistent calculation of the boundary conditions, etc. The compatibility of the ASTRA system with the MFDEB (Magnetic Fusion Energy Database, TEXAS, USA) - the international base for data on tokamaks, including discharge parameters from such machines as JT-60, TFTR, JET, T-10, PLT, TEXT, ASDEX, etc., provides a significant extension of its range of applications. Therefore the ASTRA system may be used for processing and analyzing experimental data from any tokamak device.

The present description corresponds to version 3.0 of the

ASTRA code, implemented on the IBM PC/AT in the DOS system. Operation of the code demands a DOS version not lower than 3.2, a Microsoft FORTRAN Compiler and a Microsoft Linker corresponding to versions not under 5.0 and 3.5, respectively, as well as EGA or VGA graphic adapters. It is also possible to work with the WEITEK processor. Then the NDP FORTRAN 2.1 translator and the Phar Lap 2.2d loader are needed. For graphic representation of the output in three-dimensional form, the program SURFER may be employed.

Before passing on to a detailed descriptions of instructions for working with the ASTRA system, we will discuss an example of composing data for modelling transport processes. To set up such an input, the user must answer the following questions:

(*) Which system of equations is going to be solved?

(**) Which form is wanted for the results? Which variables and what form should the output have?

(***) Which are the initial conditions for the calculation?

As an answer to the first question, a list of variables for the description by transport equations (e.g. n_e, T_e) must be given, as well as an indication of other processes (neutral atom transport, impurity radiation, auxiliary heating and current generation methods, etc.). Thus, it is sufficient to write, e.g.

```
Te:Equation
He = HAALC                                    (1)
Pe = Pjoul - Peicl + Pex
```

to set up a program for solving the equation

$$\frac{3}{2}\frac{\partial}{\partial t}(n_e T_e) + \nabla(n_e \cdot H_e \cdot \nabla T_e) = P_e \ .$$

Then the formulas for Joule heating **Pjoul**, heat exchange with ions **Peicl**, and Alcator thermal conductivity **HAALC** will be inserted from the standard-formula library of the ASTRA system and put into the program text wherever necessary. The electron temperature T_e will be obtained as a result of the calculation, while the auxiliary heating power **Pex**, the

electron density n_e, the ion temperature T_i (necessary for calculating **Peicl**), tokamak parameters and a number of other quantities may change from one run to the next and should be determined at the start of the task execution.

The determination of output variables (**) is just as simple. For example, the specification

```
q\-HAALC*Ne*GRAD(Te)
Te\Te;          Texp\Tex                    (2)
Wt_WTOTB
```

provides, as an output, the heat flow $-H_e n_e \nabla T_e$ being called **q**, the profiles of the calculated T_e and the experimental electron temperature T_e^{exp} being called **Te** and **Texp**, respectively as well as the total plasma energy content

$$\mathbf{WTOTB} = \int_V (n_e T_e + n_i T_i) dV$$

being called **Wt**.

The first three of the indicated quantities - the heat flow q, the electron temperature T_e, and the experimental electron temperature T_e^{exp}, appear as output on the screen and on the printer as functions of radius, while the global energy content Wt appears as a function of time.

Generally speaking, giving these data completes the information needed for the program generator. The answer to question (***) is not directly linked with the transport model. Indeed, the equation for T_e may be solved not only with different initial conditions, but also for different machines or for different operating conditions of a certain machine. The corresponding characteristics of the tokamak and of its operation mode, and also the number of variables needed in the calculation, but not determined by the given equations and formulas (1): T_i, n_e, P_e^{exp}, T_e^{exp}, j, and others, are written into a file which must be called at the start of the program execution. Beside the tokamak parameters and the initial data for the calculation, this file may also contain experimental information on the evolution of the plasma parameters (in the example discussed, e.g. $T_e^{exp}(\rho,t)$). The set of such files constitutes the experimental data base of the ASTRA system; it

can be easily extended and at present contains about 100 experimental discharges.

Within the logic just described, the problem in general will not require an answer to question (*). Thus e.g. in the example taken above, part of the input (1) may be omitted. In this case, the output (3) will be determined by data from the experimental data file. Such a working scheme for the ASTRA system gives the possibility of analyzing experimental data without transport modelling. Calling e.g.

```
tauE\WTOTB/(QEXB+UTB*IPL);          [Te]\WEXB/NEXAVB;
<Te>\TEXAVB;                        QTe_TEXC/TEXAVB;
```

will compile a program which does not solve any equations, but will calculate the quantities

$$\langle Te \rangle = \frac{1}{V}\int_V T_e^{exp} dV, \qquad [Te] = \frac{\int_V n_e^{exp} \cdot T_e^{exp} dV}{\int_V n_e^{exp} dV},$$

$$tauE = \frac{\int_V \left[n_e^{exp} T_e^{exp} + n_i^{exp} T_i^{exp} \right] dV}{\int_V P_e^{exp} dV + U_{pl} \cdot I_{pl}}$$

from experimental data.

The examples discussed above, in spite of their simplicity, contain exhaustive information on the input needed for the calculation, which leads to the construction of an operative and useful transport program, with minimal need of specific knowledge about computer software, programming techniques, and numerical methods. The user also needs not to worry about the compatibility of the units used in different formulas, and in the great majority of cases not even about the way of writing these formulas, as the ASTRA library contains practically all formulas required in transport modelling. For working with the ASTRA system, it is only necessary to know how to use a text editor, how to write formulas in FORTRAN language, and what the names of the formulas are the ASTRA library.

After construction of the computer code for a transport model of the form (1), (2), the monitor supervisor of the ASTRA system starts the run of the task with initial conditions chosen by the user (calculation variant). The initial value data file provides the parameters left undetermined in the transport model, e.g. the machine dimensions, field intensity, current of the discharge, and others; besides, this file may contain the evolution of the experimental profiles, for comparison with the calculations.

The high efficiency of the calculations, even when using comparatively small computers of the PC/AT type, is achieved because the user may make interactive corrections in the transport equations, the parameters of the discharge to be analyzed, and the output parameters desired. Changes in the equations must be foreseen by the user when establishing the model. Thus, writing the electron thermal conductivity in the form

```
He = CHE1*HAETI + CHE2*HACTE                    (3)
```

permits changing the coefficients **CHE1** and **CHE2** on the terminal screen and thus modify the calculation run by introducing the electron thermal conductivity as an arbitrary linear combination of transport coefficients corresponding to the η_i mode, **HAETI**, and the collision-free mode, **HACTE**, for trapped electrons.

A graphical representation of the ASTRA system structure is given in Fig. 1.

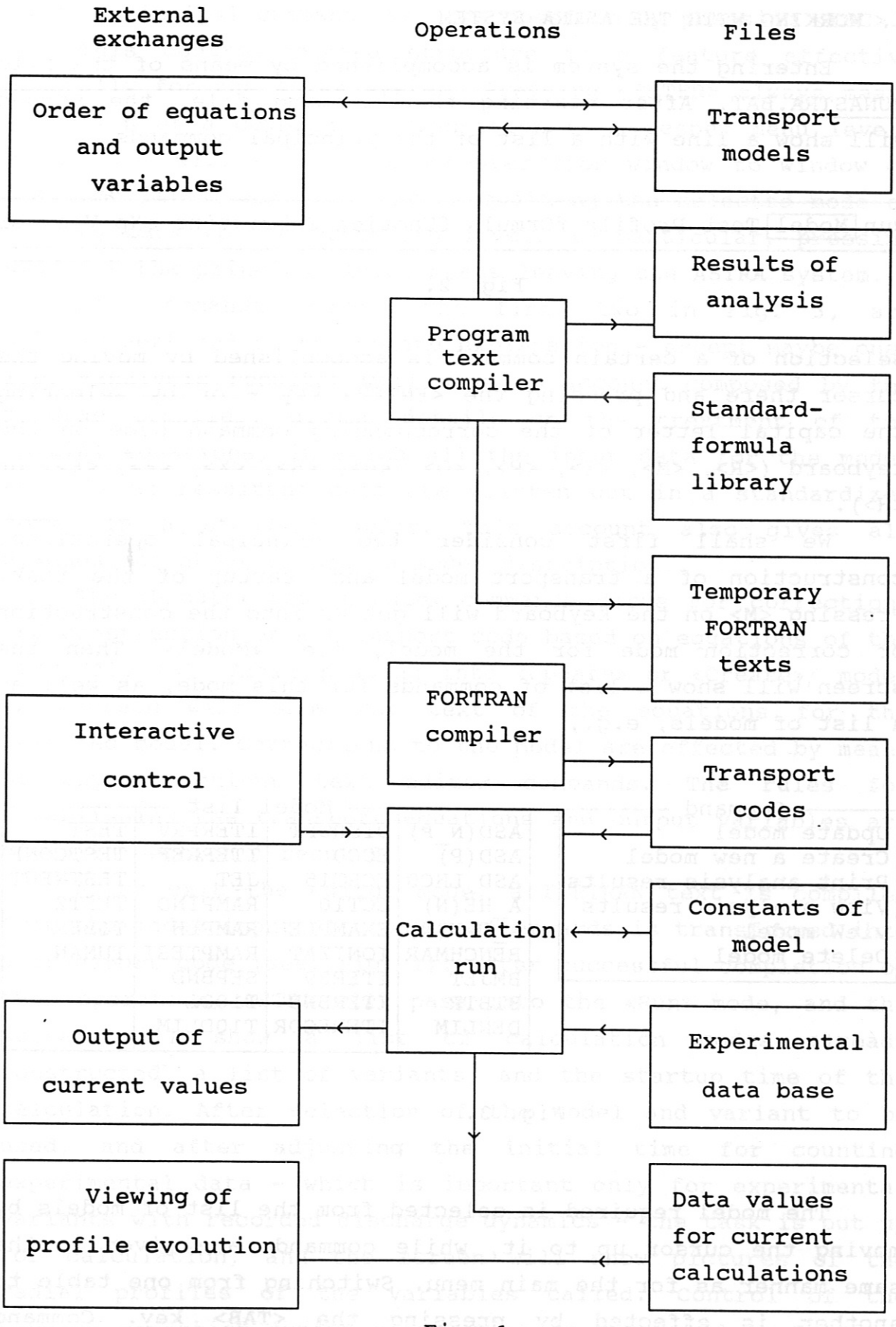

External exchanges
Operations
Files
Order of equations and output variables
Transport models
Results of analysis
Program text compiler
Standard-formula library
Temporary FORTRAN texts
FORTRAN compiler
Interactive control
Transport codes
Calculation run
Constants of model
Output of current values
Experimental data base
Viewing of profile evolution
Data values for current calculations

Fig. 1.

1. WORKING WITH THE ASTRA SYSTEM

Entering the system is accomplished by means of the file RUNASTRA.BAT. After starting this command file, the screen will show a line with a list of the principal commands:

Run [Model] Test Profile fOrmula fUnction Subroutine eXp View sErvice

Fig. 2.

Selection of a certain command is accomplished by moving the cursor there and pressing the <ENTER> key - or by selecting the capital letter of the corresponding command name on the keyboard (<R>, <M>, <T>, <P>, <O>, <U>, <S>, <X>, <F>, <E>, or <H>).

We shall first consider two principal operations: construction of a transport model and startup of the task. Pressing <M> on the keyboard will get us into the construction or correction mode for the model, i.e. «Model». Then the screen will show a list of commands for this mode, as well as a list of models, e.g.,

Command
Update model
Create a new model
Print analysis results
View analysis results
vieW model
Delete model

Model list			
ASD(N_P)	DISRUPT	ITERPRV	TEST
ASD(P)	ECCD	ITERREF	TESTCONF
ASD_LHCD	ECRT15	JET	TESTNEUT
A_HE(N)	ECT10	RAMPIND	TESTZ
A_TAU(N)	EXAMPLE	RAMPLH	TIBER
BENCHMAR	IONIZAT	RAMPTEST	TUMAN
BMJET	ITER89	SEPBND	
BTBTR	ITERBND	T10DL	
DENLIM	ITEREGOR	T10DLIM	

Fig. 3.

The model required is selected from the list of models by moving the cursor up to it, while commands are given in the same manner as for the main menu. Switching from one table to another is effected by pressing the <TAB> key. Command execution is achieved by pressing the <ENTER> key, while a change of the model-construction mode, i.e. a transition back

to the principal-command list, is possible by pressing <ESC>.

This control-command structure is a feature effective generally for the ASTRA system. Pressing <ENTER> always means execution of a command or transition to a deeper menu level, or a <TAB> will achieve change-over from window to window on the same level, and <ESC> means quitting the selected mode or passing back to the preceding level. In particular, pressing <ESC> at the principal level means leaving the ASTRA system.

All commands, except the first two in Fig. 3, are self-evident and do not demand explanation - except maybe one: i.e. «analysis results» will give an account composed by the program compiler, giving details on the treatment of the initial equations, in which all the input data for the model as well as resultant data are written out in a standardized form, in a standard order. This account also gives all diagnostics about errors in model description.

The «Update» and «Create» commands serve for corrections and construction of a transport code based on equations of the form (1), (3). After passing into «Update» or «Create»/ mode, the screen will show the text of the equations for the selected model. Corrections to the model are effected by means of any convenient text editor commands. The rules for establishing the transport equations and output variables are given in Section 4 resp. 5.

After exit the editor mode, a FORTRAN text is compiled from the model input (1), which afterwards is transformed into a transport code (see. Fig. 1). After successful completion of this operation, ASTRA will pass into the «Run» mode, and the screen will show a list of calculation codes already constructed, a list of variants, and the startup time of the calculation. After selection of the model and variant to be used, and after adjusting the initial time for counting experimental data - which is important only for experimental variants with recorded discharge dynamics - the task is put up for calculation, and the screen will show pictures of the radial profiles of the variables called. Control of the program operation during calculation is described in Sections 7,8.

We will give a short description of the remaining commands form the basic list. «Test» is a command for manual text correction, to be used for obtaining additional diagnostic printouts for error search. The position of the files being edited within the transport-code structure may be gathered from Appendix 2. With the «Profile» command, it is possible to look over results of earlier calculation «View run», compare two calculations with each other «Two runs» in the standard ASTRA graphic modes, and also to follow the time evolution of radial profiles with the help of the program SURFER «3D Plot». For saving the calculation results, it is necessary to follow end of calculation by copying the results of the «Last run» in a new file by means of the «Save last» command.

The commands «fOrmula», «fUnction», and «Subroutine» are destined for inspection, editing and construction of standard formulas, functions, and subroutines, and the command «eXp» is needed for construction and editing of the experimental data base of the ASTRA system. «View» permits a reviewing of all files of the ASTRA complex. «sErvice» serves for copying, library renewal, and service-program editing. For using this command, it is necessary to know the «Password».

2. TRANSPORT EQUATIONS

For describing the transport of energy and particles and the magnetic field in the framework of the ASTRA system , we use the system of one-dimensional equations obtained by averaging over the toroidal magnetic surfaces [2]. In a Gaussian system of units, this set of transport equations has the form

$$\frac{\partial}{\partial t}(V' n_e) + \frac{\partial}{\partial \rho}\left[V'\left(<(\nabla\rho)^2>\Gamma_e - \frac{1}{2}\frac{\dot{B}_0}{B_0}\rho\, n_e\right)\right] = V' S \ ,$$

$$\frac{3}{2}\frac{1}{\rho}(V')^{-5/3}\left(\frac{\partial}{\partial t} - \frac{\rho}{2}\frac{\dot{B}_0}{B_0}\frac{\partial}{\partial \rho}\right)(\rho V'^{5/3} n_e T_e) +$$

$$+ \frac{1}{V'}\frac{\partial}{\partial \rho}\left[V'<(\nabla\rho)^2>(q_e + \frac{5}{2}T_e\Gamma_e)\right] = P_e \ ,$$

(4)

$$\frac{3}{2}\frac{1}{\rho}(V')^{-5/3}\left(\frac{\partial}{\partial t} - \frac{\rho}{2}\frac{\dot{B}_0}{B_0}\frac{\partial}{\partial \rho}\right)(\rho V'^{5/3} n_i T_i) +$$

$$+ \frac{1}{V'}\frac{\partial}{\partial \rho}\left[V'<(\nabla\rho)^2>(q_i + \frac{5}{2}T_i\Gamma_i)\right] = P_i \ ,$$

$$\frac{2\pi B_0}{c}\frac{\rho}{V'}\sigma_{\parallel}\left[\frac{\partial \psi}{\partial t} - \frac{\rho \dot{B}_0}{2B_0}\frac{\partial \psi}{\partial \rho}\right] =$$

$$= \frac{cI^2}{8\pi^2 V'}\frac{\partial}{\partial \rho}\left[\frac{V'}{I}<\left(\frac{\nabla\rho}{r}\right)^2>\frac{\partial \psi}{\partial \rho}\right] - <(\vec{j}_{bs} + \vec{j}_{ext})\vec{B}> \ .$$

In Eqs (4), all quantities depend on two arguments: time t and the "radial" variable ρ. The quantity equivalent to the minor torus radius may be expressed via the toroidal magnetic flux Φ:

$$\pi B_0 \rho^2 \equiv \Phi \equiv \int_{S_\zeta} \vec{B} d\vec{S}_\zeta = \frac{1}{2\pi}\int_V (\vec{B}\nabla\zeta) d^3x = \frac{1}{2\pi}\int \frac{I}{r^2}\, d^3x \qquad (5)$$

where (r,ζ,z) are from a cylindrical coordinate system where the axis coincides with the major torus axis, $\vec{B}$ is the magnetic field, $B_0 = B_0(t)$ is the toroidal magnetic field in the geometric centre of the vacuum vessel $r = R_0$ in the absence of plasma, $\dot{B}_0 = dB_0/dt$, S_ζ - the torus cross section in a plane ζ = Const passing through its major axis.

All these equations are written for variables averaged over a magnetic surface, and the averaging process is defined as

$$\langle f\rangle = \frac{d}{dV}\int_V f d^3x = \frac{d}{dV}\int_0^{\rho} d\rho \int_0^{2\pi} d\zeta \int_0^{2\pi} \sqrt{g} f d\vartheta = 2\pi \frac{d\rho}{dV}\int_0^{2\pi} \sqrt{g} f d\vartheta \qquad (6)$$

where $V = V(\rho,t)$ is the volume of that region in space which is bounded by the given toroidal magnetic surface ρ=const. ϑ is a poloidal, ζ the toroidal angle, and the volume element is $d^3x = \sqrt{g}d\rho d\vartheta d\zeta = \left|\frac{D(x,y,z)}{D(\rho,\vartheta,\zeta)}\right| d\rho d\vartheta d\zeta$. For such an averaging scheme, the formula

$$\langle \mathrm{div}\ \vec{A}\rangle = \frac{1}{V'}\frac{\partial}{\partial\rho} V' \langle \vec{A}\nabla\rho\rangle ,$$

is valid. Putting f = 1 in (6), we obtain the following equality for the volume derivative:

$$V' \equiv \frac{\partial V}{\partial\rho} = \frac{\partial}{\partial\rho}\int_V d^3x = 2\pi \int_0^{2\pi} \sqrt{g} d\vartheta .$$

Beside ρ, the following functions of a magnetic surface are employed: I - the poloidal current function (see Eq. (5)), ψ - the poloidal magnetic field flux. Quantities also dependent on the magnetic surfaces are: the electron density $n(\rho,T)$, the electron and ion temperatures $T_e(\rho,t)$ resp. $T_i(\rho,t)$, the particle source $S(\rho,t)$ and the heat sources $P_{e,i}(\rho,t)$.

The quantity

$$-0.5\rho\dot{B}/B_0 = \langle \vec{u}_\Phi - \vec{u}_\rho, \nabla\rho\rangle = (\partial\rho/\partial t)\Big|_\Phi$$

designates a velocity of the magnetic surface Φ = Const relative to the surface ρ = Const; $\vec{u}_\Phi$ and $\vec{u}_\rho$ are the local velocities of these surface motions. Save where expressly stated otherwise, $\partial/\partial t$ is to be calculated for fixed ρ.

The average fluxes of particles and heat, i.e. $\Gamma_e(\rho,t)$, $q_e(\rho,t)$, $q_i(\rho,t)$, may be expressed via the local particle velocities $\vec{u}_e$ and the local heat flows $\vec{Q}_{e,i}$, in the following manner:

$$\Gamma_e = \langle n_e(\vec{u}_e - \vec{u}_\Phi)\nabla\rho\rangle / \langle(\nabla\rho)^2\rangle$$

$$q_e + \frac{5}{2}T_e\Gamma_e = \langle(\vec{Q}_e - \frac{5}{2}T_e n_e \vec{u}_\Phi)\nabla\rho\rangle / \langle(\nabla\rho)^2\rangle$$

$$q_i + \frac{5}{2}T_i\Gamma_i = \langle(\vec{Q}_i - \frac{5}{2}T_i n_i \vec{u}_\Phi)\nabla\rho\rangle / \langle(\nabla\rho)^2\rangle$$

These fluxes are linked to the generalized forces via the matrix

$$\begin{vmatrix} \frac{\Gamma_e}{n_e} \\ \frac{q_e}{T_e} \\ \frac{q_i}{T_i} \\ \frac{4\pi}{c}\frac{R_0 J_B}{B_0 \rho \mu} \end{vmatrix} = - \begin{vmatrix} D_n & D_e & D_i & D_E \\ \kappa_e^n & \kappa_e & \kappa_e^i & \kappa_e^E \\ \kappa_i^n & \kappa_i^e & \kappa_i & \kappa_i^E \\ C_n & C_e & C_i & 0 \end{vmatrix} \cdot \begin{vmatrix} \frac{1}{n_e}\frac{\partial n}{\partial \rho} \\ \frac{1}{T_e}\frac{\partial T_e}{\partial \rho} \\ \frac{1}{T_i}\frac{\partial T_i}{\partial \rho} \\ \frac{cE_\parallel}{B_p} \end{vmatrix} \qquad (7)$$

Here we have $J_B = \langle \vec{j}_{bs}, \vec{B}\rangle / B_0$, $E_\parallel = \langle \vec{E}, \vec{B}\rangle / B_0$, $B_p = 1/R_0 \cdot \partial\psi/\partial\rho$, $\mu = \partial\psi/\partial\Phi$.

Density n_i and flux Γ_i of ions are determined via the corresponding quantities for electrons:

$$n_i = n_e / Z_{main}, \qquad \Gamma_i = \Gamma_e n_i / n_e,$$

where Z_{main} is the charge of the majority ions in the plasma.

Sources and sinks for particles and heat have the following form for the general case:

$$S = (s_{ion} N - s_{rec} n_i) n_e$$

$$P_e = jE \cdot \frac{V'}{4\pi^2 \rho R_0} - \langle(\nabla\rho)^2\rangle \frac{\Gamma_e}{n_e}\frac{\partial(n_i T_i)}{\partial\rho} + P_e^N + P_e^H + P_e^{CD} - P_{ei} - P_e^{rad}$$

$$P_i = \langle(\nabla\rho)^2\rangle \frac{\Gamma_e}{n_e}\frac{\partial(n_i T_i)}{\partial\rho} + P_i^N + P_i^H + P_i^{CD} + P_{ei}$$

The quantity P_{ei} describes Coulomb heat exchange between electrons and ions, P_{ei}^N are the heat sinks of the electron and ion components, which are linked to the neutral atoms by

$$P_e^N = -(s_{ion} N E_{ion} + s_{rad} N E_1 + s_{rec} n_i T_e) n_e$$

$$P_i^N = \frac{3}{2}(s_{ion} N T_N + s_{cx} N (T_N - T_i) - s_{rec} n_e T_i) n_i$$

In these formulas s_{ion}, s_{rad}, s_{rec} and s_{cx} are coefficients of the rate of ionization, radiation, recombination and charge exchange, $E_{ion} = 12.6$ eV the ionization potential, $E_1 = 10.2$ eV the energy for the ground-state transition of the hydrogen atom, N the density of neutral gas, $P_{e,i}^H$ the power of the sources for additional electron resp. ion heating. The components $P_{e,i}^{CD}$ describe the plasma heating due to induction-free methods of current generation. Finally, the term $jE \cdot \frac{V'}{4\pi^2 \rho R_0} \sim jE$ describes the energy exchange between the plasma and the toroidal electric field. We point out that the term jE is not the same as the Joule heating σE^2, e.g. when the current increases due to non-inductive methods, we have $jE < 0$.

The right-hand part of the equation for the poloidal flux contains the quantity $j_{ext} = \langle \vec{j}_{ext} \cdot \vec{B} \rangle / B_0$, i.e. the density of non-inductive current. It is common practice to introduce the concept of stationary current-generation effectivity, defining this by the formula

$$\eta = \frac{j_{ext}}{P_e^{CD} + P_i^{CD}}$$

We point out that under such a definition, the sign of the quantity η depends on the choice of coordinate system and may be negative for $j_{ext} < 0$. This quantity also depends on the plasma parameters, electric field intensity, and the method of current generation.

For closing the system (4), (7), it is imperative to determine the averages of the metric coefficients, $\langle (\nabla\rho)^2 \rangle$ and $\langle (\nabla\rho/r)^2 \rangle$, entering (4). Besides, it is necessary to give V' and I as functions of the variable ρ: $V'(\rho)$, $I(\rho)$. The required quantities are expressed via the metric-tensor components g_{ik} in the coordinate system (ρ, ϑ, ζ) in the following manner:

$$
\begin{aligned}
(\nabla\rho)^2 &= g^{11} = \frac{g_{22}g_{33}}{g} = \frac{g_{22}r^2}{g}\ , \\
\left(\frac{\nabla\rho}{r}\right)^2 &= \frac{g_{22}}{g}\ , \\
(\nabla\zeta)^2 &= g^{33} = (g_{33})^{-1} = r^{-2}\ , \\
g &= \det g_{ik} = \left(\frac{D(x,y,z)}{D(\rho,\vartheta,\zeta)}\right)^2 = r^2\left(\frac{D(r,z)}{D(\rho,\vartheta)}\right)^2
\end{aligned}
\tag{8}
$$

From (5), (6) we have

$$
4\pi^2\rho B_0 = \frac{\partial V}{\partial\rho}\,\langle Ir^{-2}\rangle = IV'\langle r^{-2}\rangle\ .
$$

Thus, what remains is the problem of finding the metric-tensor components g_{ik} as functions of spatial coordinates, $g_{ik} = g_{ik}(\vec{x})$ or, in other words, of determining the magnetic configuration of the system. Therefore, we must first solve the Grad-Shafranov equilibrium equation [3]. In the version 3.0 of the ASTRA program, a simplified moment description of the equilibrium is used in the form [4]

$$
\begin{aligned}
r &= R_0 + \Delta(a) + a(\cos\vartheta - \delta(a)\sin^2\vartheta)\,, \\
z &= a\lambda(a)\sin\vartheta\ .
\end{aligned}
\tag{9}
$$

Here $\lambda(a)$ is the elongation of the magnetic surface, $\delta(a)$ the "triangularity" (relative shift of the maximal-z point on the magnetic surface, to the interior).

The current densities j (toroidal) resp. $j_\parallel$ (longitudinal), the rotational-transform coefficient μ, the intensity of the longitudinal electric field E - all are expressed by the other variables in the following way:

$$
\begin{aligned}
j &= R_0\langle\vec{j}\cdot\nabla\zeta\rangle = \frac{c}{8\pi^2}\frac{R_0}{V'}\frac{\partial}{\partial\rho}\left\{V'\langle(\nabla\rho/r)^2\rangle\frac{\partial\psi}{\partial\rho}\right\}\ , \\
j_\parallel &= \frac{1}{B_0}\langle\vec{j}\cdot\vec{B}\rangle = \frac{c}{8\pi^2}\frac{I^2}{B_0V'}\frac{\partial}{\partial\rho}\left\{\frac{V'}{I}\langle(\nabla\rho/r)^2\rangle\frac{\partial\psi}{\partial\rho}\right\}\ , \\
\mu &= \frac{1}{q} = \frac{\partial\psi}{\partial\Phi} = \frac{1}{2\pi B_0\rho}\frac{\partial\psi}{\partial\rho}\ , \\
E_\parallel &= \frac{1}{B_0}\langle\vec{E}\cdot\vec{B}\rangle = \frac{2\pi\rho}{cV'}\left.\frac{\partial\psi}{\partial t}\right|_\Phi .
\end{aligned}
\tag{10}
$$

We also write the local relations for the fields, i.e.

$$\vec{B} = I\cdot\nabla\zeta - \frac{1}{2\pi r}\left[\nabla\psi,\nabla\zeta\right] ,$$

$$(\vec{E}\cdot\nabla\zeta) = \frac{1}{2\pi c r^2}\left.\frac{\partial\psi}{\partial t}\right|_{\vec{x}} , \qquad (11)$$

and the expressions for the total toroidal plasma current

$$I_{pl} = \int_{S_\zeta}\vec{j}d\vec{S}_\zeta = \frac{1}{2\pi}\int_V (\vec{j}\cdot\nabla\zeta)\frac{d^3x}{r} = \frac{cV'}{16\pi^3}\left\langle(\nabla\rho/r)^2\right\rangle \left.\frac{\partial\psi}{\partial\rho}\right|_{\rho=\rho_s} \qquad (12)$$

and the toroidal loop voltage

$$U_{pl} \equiv 2\pi R_0 E_\parallel = \frac{4\pi^2 R_0\rho}{cV'}\left.\frac{\partial\psi}{\partial t}\right|_\Phi = \frac{4\pi^2 R_0\rho}{cV'}\left(\frac{\partial\psi}{\partial t} - \frac{\rho}{2}\frac{\dot{B}_0}{B_0}\frac{\partial\psi}{\partial\rho}\right) \qquad (13)$$

The quantity U_{pl} is equal to the voltage on the moving plasma boundary and generally will not coincide with the voltage U_0 which is measured by a fixed loop in experimental conditions:

$$U_0 = \frac{1}{c}\left.\frac{\partial\psi}{\partial t}\right|_{\vec{x}} = \frac{1}{c}\left.\frac{\partial\psi}{\partial t}\right|_\Phi + \frac{1}{c}\mu\left.\frac{\partial\Phi}{\partial t}\right|_{\vec{x}} . \qquad (14)$$

If we do not consider the processes of plasma displacement as a whole or adiabatic compression , we shall have $U_0(\rho=\rho_s) = U_{pl}$. In the opposite case, U_0 will have a contribution from the inductor flux and from other poloidal coils, and a calculation of the quantity U_0 must use the solution of an equilibrium equation for the plasma ring with free boundary in external magnetic fields.

3. PROGRAMMING THE SYSTEM OF TRANSPORT EQUATIONS

The system (4), (7) of equations represents the general form of a transport-equation system. In realistic problems, it is rarely ever necessary to employ the full system of equations. Therefore, the ASTRA system provides the user with the possibility of assembling his own calculation program (model) from a shortened system of equations as well as from an incomplete matrix (7). Parts of the system (4) subject to modifications are the fluxes Γ_e, q_e, q_i, the particle sources

S, heat sources $P_{e,i}$, current sources j_{ext}, as well as the bootstrap current j_{bs} and the longitudinal conductivity $\sigma_{\|}$.

In the ASTRA system, the following system of units and names for the principal variables are used:

ρ	- [m]	current radius
t	- [sec]	current time
$n_{e,i}$	- [10^{19}m^{-3}]	electron density
$T_{e,i}$	- [keV]	electron resp. ion temperature
ψ	- [V·s]	poloidal flux

and for auxiliary variables

$\Gamma_{e,i}$	- [10^{19}m^{-2}s^{-1}]	particle flux
$q_{e,i}$	- [MW·m^{-2}]	heat flux
D, κ	- [m^{2}s^{-1}]	matrix elements (7)
C	- [ ]	matrix elements (7)
S	- [10^{19}m^{-3}s^{-1}]	particle sources and sinks
$P_{e,i}$	- [MW·m^{-3}]	heat sources and sinks
$j_{\|}$	- [MA·m^{-2}]	current density
$\sigma_{\|}$	- [$\mu\Omega$·m^{-1}]	conductivity
B_0	- [T]	toroidal field
U_{pl}	- [V]	loop voltage
I_{pl}	- [MA]	plasma current

We shall also give the names used in the program for simple variables (the FORTRAN program identifiers are underlined in this paragraph):

B_0 = BTOR, R_0 = RTOR,
t = TIME, Δt = TAU,
π = GP, I_{pl} = IPL

and for arrays entering the transport equation system (4):

$T_e(\rho,t)$ = TE, $T_e(\rho,t-\Delta t)$ = TEO,
$T_i(\rho,t)$ = TI, $T_i(\rho,t-\Delta t)$ = TIO,
$n_e(\rho,t)$ = NE, $n_e(\rho,t-\Delta t)$ = NEO,
$n_i(\rho,t)$ = NI, $n_i(\rho,t-\Delta t)$ = NIO,
$\psi'(\rho,t)$ = FP, $\psi'(\rho,t-\Delta t)$ = FPO,
$V'(\rho,t)$ = VNEWR, $V'(\rho,t-\Delta t)$ = VOLDR,
$j_{\|}(\rho,t)$ = CU, $\mu(\rho,t)$ = MU, (15)

$$U_{pl}(\rho,t) = \underline{UT}, \qquad \rho = \underline{ROMTR},$$

$$V' \langle(\nabla\rho)^2\rangle = \underline{G2GR2}, \qquad \frac{I}{(B_0 R_0)} = \underline{IPOL},$$

$$\frac{1}{\langle r^{-2}\rangle}\,(B_0 R_0/I)^2 \rho \langle(\nabla\rho/r)^2\rangle = \underline{G22G}, \qquad R_0^2\langle r^{-2}\rangle = \underline{R2MTR}.$$

For convenient practical working, we will write the system of transport equations in units and names of the ASTRA system

$$\frac{1}{V'}\frac{\partial}{\partial t}(V'\,\underline{NE}) + \frac{1}{V'}\frac{\partial}{\partial\rho}\left[\left(\underline{G2GR2}\;\Gamma_e - V'\,\frac{1}{2}\frac{\dot{B}_0}{B_0}\,\rho\;\underline{NE}\right)\right] = \underline{SN} + \underline{SNN}\cdot\underline{NE}\ ,$$

$$\frac{3}{2}\frac{1}{\rho}(V')^{-5/3}\left(\frac{\partial}{\partial t} - \frac{\rho}{2}\frac{\dot{B}_0}{B_0}\frac{\partial}{\partial\rho}\right)\left(\rho V'^{5/3}\,\underline{NE}\;\underline{TE}\right) +$$

$$+ \frac{1}{V'}\frac{\partial}{\partial\rho}\left[\underline{G2GR2}\;\left(625\,q_e + \frac{5}{2}\,\underline{TE}\;\Gamma_e\right)\right] = 625\left(\underline{PE} + \underline{PET}\cdot\underline{TE}\right)\ ,$$

$$\frac{3}{2}\frac{1}{\rho}(V')^{-5/3}\left(\frac{\partial}{\partial t} - \frac{\rho}{2}\frac{\dot{B}_0}{B_0}\frac{\partial}{\partial\rho}\right)\left(\rho V'^{5/3}\,\underline{NI}\;\underline{TI}\right) + \qquad (16)$$

$$+ \frac{1}{V'}\frac{\partial}{\partial\rho}\left[\underline{G2GR2}\;\left(625\,q_i + \frac{5}{2}\,\underline{TI}\;\Gamma_i\right)\right] = 625\left(\underline{PI} + \underline{PIT}\cdot\underline{TI}\right)\ ,$$

$$0.4\pi\;\underline{IPOL}^2\;\underline{CC}\left[\frac{\partial\psi}{\partial t} - \frac{\rho\dot{B}_0}{2B_0}\frac{\partial\psi}{\partial\rho}\right] = \frac{\partial}{\partial\rho}\left[\underline{G22G}\;\frac{\partial\psi}{\partial\rho}\right] -$$

$$- 0.8\pi\;\underline{GP}\;\underline{RTOR}\;\underline{IPOL}^{-3}\underline{R2MTR}^{-1}\left(\underline{CD} + J_B\right)\ .$$

In the ASTRA system, it is possible to give the right-hand parts of the transport-equation system in the form of two components:

$$S = \underline{SN} + \underline{SNN}*\underline{NE}\ ,$$

$$P_e = \underline{PE} + \underline{PET}*\underline{TE}\ , \qquad (17)$$

$$P_i = \underline{PI} + \underline{PIT}*\underline{TI}\ .$$

This means that it is possible to use an explicit or implicit approximation for the right-hand parts in a difference scheme, for controlling the stability of the difference scheme.

Usually the stability of the scheme is guaranteed by diffusion terms in the equation, and therefore as a rule only one (arbitrary) component form each pair may be used. The quantity $\underline{CD} = \langle \vec{j}_{ext}, \vec{B} \rangle / B_0$ designates the current from an arbitrary driver (e.g. a non-inductive source). The particle Γ and heat q_e, q_i fluxes, as well as the bootstrap current $J_B = \langle \vec{j}_{bs}, \vec{B} \rangle / B_0$ are expressed via the problem's variables in the following form:

$$\begin{vmatrix} \frac{q_e}{\underline{NE}\ \underline{TE}} \\ \frac{q_i}{\underline{NI}\ \underline{TI}} \\ \frac{\Gamma}{\underline{NE}} \\ J_B\left(\frac{0.4\pi R_0}{B_0 \rho \mu}\right) \end{vmatrix} = \begin{vmatrix} \underline{HE} & \underline{XE} & \underline{DE} & \underline{CE} \\ \underline{HI} & \underline{XI} & \underline{DI} & \underline{CI} \\ \underline{HN} & \underline{XN} & \underline{DN} & \underline{CN} \\ \underline{HC} & \underline{XC} & \underline{DC} & 0 \end{vmatrix} \circ \begin{vmatrix} -\frac{1}{\underline{TE}} \frac{\partial\, \underline{TE}}{\partial r} \\ -\frac{1}{\underline{TI}} \frac{\partial\, \underline{TI}}{\partial r} \\ -\frac{1}{\underline{NE}} \frac{\partial\, \underline{NE}}{\partial r} \\ 1 \end{vmatrix} \qquad (18)$$

The longitudinal conductivity $\sigma_{\parallel}$ - $\underline{CC}$ (Current Conductivity) is directly contained in the last of the system equations. Setting quantities

$$\underline{SN},\ \underline{SNN},\ \underline{PE},\ \underline{PET},\ \underline{PI},\ \underline{PIT},\ \underline{CD},\ \underline{CC},\ \underline{HE},\ \underline{XE},\ \underline{DE},$$
$$\underline{CE},\ \underline{HI},\ \underline{XI},\ \underline{DI},\ \underline{CI},\ \underline{HN},\ \underline{XN},\ \underline{DN},\ \underline{CN},\ \underline{HC},\ \underline{XC},\ \underline{DC}$$

completely determines the equation system to be solved. It is, however, by no means obligatory to solve all equations and to fix all the values in the transport-coefficient matrix. From the 23 quantities enumerate above, any selection may be fixed by indicating which equations, in what order, and with which coefficients they are to be solved. If the plasma conductivity $\underline{CC}$ is not given by the user, it will be assumed in Spitzer form. Besides, boundary conditions are required as well as initial conditions. In many cases, however, these quantities are given at the start of the task by the choice of calculation variant itself and do not demand any special determination.

4. SETTING UP THE TRANSPORT EQUATION

The transport coefficients, output variables, and other parameters in the ASTRA system are stated via control lines of the form (1)-(3). Text after a "!" sign is commentary and not interpreted by the program generator. For condensing the write-up of short control lines they may be joined into one by using the ";" sign. If the "!" sign stands in the first position, the entire line - not only the part up to ";" is taken as commentary. Also, any line which is not a control line, i.e. does not have one of the signs "=", ":", "\" or "_", is considered commentary. Blanks in command lines are ignored.

A control line for an equation has the form

VAR:EQT, (19)

where VAR fixes the variable for which the transport equation is to be solved, and EQT is the type of the equation. The parameter VAR may assume only the following values:

NE - electron density;
TI - ion temperature;
TE - electron temperature;
CU - current density.

For the equation type EQT, the following options are valid:

FULL - calculation following equation (4);
EQUATION - calculation from (4) without particle fluxes (for T_e and T_i). This is used if the corresponding parts of the flux $\frac{5}{2}\Gamma_{e,i}T$ are already included in the transport coefficients;
ASSIGNED - calculation after a given formula.

The EQT parameter permits trimming up to one letter: F, E, or A.

For conciseness in writing, it is assumed that missing equations or coefficients are not described at all in the control lines. In particular, if no equation for NE is given, then the plasma density will be assumed fixed, and determined within the initial conditions at the start of the run. The

absence of any particular transport coefficients in the model description means that the corresponding equations do not contain such terms. If e.g. XE and CE do not turn up, the thermal flux in the equation for the electron temperature will have only terms proportional to ∇T_e and ∇n_e. Thus it is only necessary to state those equations and coefficients that are really indispensable for the model to be discussed.

The introduction of transport coefficients means indicating the names of the corresponding coefficient - VAR - and the formula - FML - after which it is to be calculated. The control line for transport coefficients has the form

```
VAR=FML.                                                    (20)
```

The variable VAR may be any one of the transport-matrix coefficients of a source (quantities underlined in equations (16) and (18). The rules for writing the formulas FML are rather simple (see Section 6). The position of these lines in the description of the equations is not important. All transport coefficients are calculated before the re-calculation of n_e, T_e, T_i and ψ begins. As examples for coefficient control lines we refer to the expressions (1) and (3).

To test the correct input of transport coefficients and other model parameters, the program generator delivers an account of the model analysis (Fig.1). This account in the standard form contains the results from processing of control lines. Thus e.g. after analysis of the control lines (1) and (3), the generator of the program will serve the following protocol:

```
------> Equation for TE
HE(r)=HAALC(r)
PE(r)=PJOUL(r)-PEICL(r)+PEX(r)                              (21)
HE(r)=CHE1*HAETI(r)+CHE2*HACTE(r)
```

Here all variables varying with the radius take the affix (r); a warning is issued in case of non-identified variables. This warning is also displayed on the screen during the program compilation phase.

A separate discussion is necessary for equations of the type ASSIGNED in which the variable is calculated after some formula, but not from the equations (16) or (18). One of the typical cases demanding such instructions is the distribution of current density with a plasma conductivity calculated in accordance with the steady state condition ($E_{\parallel}$ = Const). In this case, it is naturally not suitable to give the transport coefficients - the variable itself must be defined. This is achieved by a control line of the form (20), in which VAR has the same values as in the control line of the equations (19). For example, a stationary distribution of current density with Spitzer conductivity is given by the following two control lines:

```
CU:AS;    CU=CCSP                                    (22)
```

If, however, a control line of this type is used with equations of the form Equation of Full, then it will only determine the initial conditions. For example:

```
TE=.1*FPR !Parabolic Te distribution;   TE:EQ         (23)
```

will define the initial electron-temperature distribution on a parabola with the temperature 0.1 keV on the axis. Now if the initial conditions for the calculation model are not given, they will be determined from initial experimental profiles from the calculation variant chosen.

The boundary conditions may also be given by control lines of the form (20), in which the variable VAR may assume the following values:

TEB	-	electron temperature at the boundary;
TIB	-	ion temperature at the boundary;
NEB	-	plasma density at the boundary;
IPL	-	total plasma current;
UEXT	-	loop voltage at the plasma boundary.

When the variable IPL and UEXT do not appear in the model, the total plasma current will be determined form the initial-data file and be assumed constant in time. In the analogous case that the boundary values for plasma temperature or density are not given by the model, they will be assumed constant and

equal to the experimental initial values for the calculated variant. If IPL and UEXT are given simultaneously, the determination for UEXT is ignored. Thus e.g. the control lines

```
TEB=TEXB;      NEB=.3;   IPL=.2*SIN(30*TIME)        (24)
```

determine the boundary electron temperature from experimental data, the boundary density as $0.3*10^{19}m^{-3}$, the time dependence of the current from a sine law, and the boundary ion temperature as constant and equal to the experimental initial value.

Besides the cases described above, a control line of the form (20) may give any other variables for the transport problem, or auxiliary parameters. In particular, it may be convenient to introduce variables for multiple employ in output. Then a correct communication with the output demands that the variable VAR is a standard variable of the program (see Appendix 1). Control lines of the form (20) may be put in at any position in the model description. In the transport code all these quantities are always calculated before the start of the re-calculation of basic variables.

Beside solutions of the basic transport equations (4) and (7), the problems arising often demand calculations for auxiliary quantities, for which the ASTRA system does not provide a standard calculation algorithm. The calculation of the distribution of neutrals may serve as such an example, necessary for determining particle sources in the plasma. Large calculation blocks of this kind are composed in the form of separate subroutines, which are called up in the model by means of the control line

```
SUB:DTeq:BEeq:ENeq:Keq                              (25)
```

where SUB is the name of the subroutine (with parameters), DTeq gives the time interval (in seconds), for which the subroutine in question is called, BEeq and ENeq determine the time interval beyond which the calculation will not turn to the subroutine, and Keq is the letter of the control key which, when pressed together with the <Ctrl> key during the calculation, will lead to the given subroutine being called (the letters <C>, <R>, <W>, <B>, and <P> are forbidden,

because they control the operation mode of the entire program). The indispensable component of this control line is only the name of the subroutine, i.e. SUB, with the first colon. If DTeq is not given, then the subroutine is called at every time step. In the absence of BEeq and ENeq, the program is called without time limitations. If Keq is not given, it is impossible to call the subroutine from the keyboard. As an example, we cite the control charge

```
NBINJ(Zimp)::10000::F !Zimp - main impurity charge
NEUT:                                                    (26)
SETNE(NEass,NEinc):0.1
```

This set of data implies the call of a calculation subroutine for neutral-beam heating, NBINJ, only via the key <Ctrl><F> (calling time for standard output is 10000 sec), a calculation subroutine for the wall-neutral density, NEUT, at every time step, of the program that processes the pre-set density SETNE over 0.1 sec. It must also be remarked that all the parameters of the subroutine command line (25) may be changed also during calculation runs. It is possible to call up to 20 subroutines in the same model.

The solution of equations for n_e, T_e, T_i and j and the calling of subroutines is executed in that order in which they appear in the model description. This order may become important if one of the subroutines uses results from another's calculations (e.g. SETNE uses the neutral particle density calculated in the subroutine NEUT) or if the subroutine must distinguish between various values of n_e, T_e, T_i and ψ in the actual and the following time steps.

5. ASSIGNMENT OF THE OUTPUT VARIABLES

The transport codes created by the ASTRA system can simultaneously deliver up to 64 radial profiles and 32 functions of time. All the output quantities may be visualized in the form of graphs, but also be transferred in digital form to the screen or to the line printer. No definite rigid output is demanded beforehand - or given by the system - , and therefore all quantities of interest to the user must be described within the model. This approach was employed

because, depending on the aims of the model's authors, the output information desired may change over such a wide range of possibilities that it appeared impossible to predict all the interesting variants and to establish an output data list that would be convenient for everybody.

The control line for the output of the variables as functions of radius has the form

```
NAME\FML\SCALE                                    (27)
```

The output call consists of the formula FML from which the output variable is calculated, the NAME of the output variable, which will only be used as an identifying legend on the output, and also the output SCALE necessary for the graph. The variable SCALE may be omitted, and then the scale is chosen automatically. If SCALE > 0, the scale is fixed and equal to SCALE. For SCALE < 0, the scale is selected automatically, but in this case it will be common to all the variables of the group with the same negative scale. If now SCALE coincides with the ordinal number of the output line for some variable, then the scale will be chosen automatically only for that variable - if it does not coincide, then for all variables of the group with the same negative scale.

The position of the corresponding graph or printed column depends on the relative location at which that variable may be found in the model description. Thus, e.g., the lines

```
Trel\TE/TE(0.);      NUe*\NUEST\2
Ptot\PETOT\-4;       Pei\PEICL\-4                 (28)
```

signify output on the first channel (first graph, first column etc.) of the normalized temperature TE/TE (0.) under the name Trel with automatic scale setting. On the second channel, output will appear for the ν_e^* profile with name NUe* and fixed scale factor 2, and the third and fourth channels will give the output for the profile of the total power transferred to the electrons and the power of the electron-ion Coulomb heat exchange, i.e. the profile, with equal automatic scaling selected in accordance with the value of PEICL..

The control lines for the output of the variables as functions of time have the form

NAME_FML_SCALE (29)

The rules here are the same as for the output of radial profiles. What must be kept in mind is that the channels for functions of time have as output only simple numerical variables, e.g. the energy confinement time τ_E (TAUEB), the temperature on the axis $T_e(0)$ (TEC) etc. Therefore the formulas that give a time-dependent output must not contain variables that are really radial profiles. More details on this will be discussed in the following section.

6. FORMULAS

When a specific model is established, formulas will be used to determine coefficients in equations, boundary and initial conditions, and output variables ((20) - (29)). There are no restrictions on the manner in which these formulas are written, except that each formula must be confined to one line. It must, however, be kept in mind that the formulas will accept only the standard variables of the ASTRA system. This means, among others, machine parameters, variables of the problems obtained from calculations with equations (16) and (18), boundary conditions, dummy variables, and library formulas. A list of the variables is given in Appendix 1, the list of library formulas (or functions) is constantly augmented, while the set of the remaining variables is fixed. The current list of formulas may be obtained by means of the commands «fOrmula» and «fUnction» in the main menu. Furthermore, the command «View» offers the text of each formula and commentaries for review, where the formula's significance is explained. A list of all the simple variables and arrays, with dimensions and convenient description may also be obtained by means of the command «View» («Variables» and «Arrays»). Examples for the writing of formulas are given in the expressions (1)-(3), (24), and (28).

We ought to direct attention to certain differences in the writing of formulas in the ASTRA system: radial variation of parameters is taken for granted and not explicitly indicated. This variation stems form the desire to shorten and simplify the statements. We will explain this by an example.

Thus, the statement

```
HE = 2 * HAMM * TE/TI                                    (30)
```

means that the electron thermal conductivity HE will be calculated in each radial position according to the right-hand expression, where all quantities dependent on radius are taken in the same position along the radius, i.e. a formula like

```
HE(r) = 2 * HAMM(r) * TE(r)/TI(r)                        (31)
```

is generally assumed.

If, however, the necessity arises to use the value of a variable in a formula, this variable depending on the radius in some radial point, then this point must be indicated expressly. It is e.g. possible to write

```
HE = 2 * HAMM * TE(0.)/TI(0.)
HE = 2 * HAMM * TE(AB/2)TI(AB/2)                         (32)
HE = 2 * HAMM * TE(0.2)/TI(0.2)
```

then the radial dependence is only assumed for the thermal conductivity HAMM, while the ratio of electron and ion temperature is in the first case taken on the axis, in the second at the midpoint of the radius, and in the third at r = 0.2 m.

For the most frequently used values of the variables, on the axis and the boundary of the plasma, abbreviations are provided: instead of TE(0) or MU(0) it is permitted to write TEC or MUC, and for TN(AB) or MU(AB) the short forms are TNB and MUB. This means that when giving the radial position it is always possible to write C (Centre) for (0.) and B (Boundary) for (AB). Strictly speaking, the suffix B at the end of the identifier of a radial profile designates not the realistic chamber boundary (AB), but the boundary of the calculation domain ABC, which, unless specifically stated otherwise, coincides with AB. In the general case, however, the calculation limit may also be set below AB; this obtains, in particular, for a shift to the outer magnetic surface relative to the centre of the discharge chamber (SHIFT≠0).

Another remark must be made regarding the employ of formulas for output of quantities depending only on time (29).

In this case, it follows from the meaning of the output quantity that the formula must give a numerical value, and therefore it must not receive quantities that may be interpreted as radius-dependent. In this output mode, it is impossible to use TE, HAMM, WTOT etc. in the formulas, while TEC, MUB, WTOT(AB/R) etc. are possible. Variables where a dependence on radius is presumed are marked with a suffix (r) in Appendix 1. The same marker if affixed to the corresponding variables in the account of the model analysis.

7. INTERACTIVE CONTROL OF THE TRANSPORT CODE

When working with the transport code of the ASTRA system, it is possible to act on the calculation itself as well as on the output of results.

Control of transport-code runs is possible via changes in the basic discharge parameters during the calculation, as well as in the transport-equation constants and subroutines and the program-control parameters.

For introducing new discharge parameters, the <V> (Variables) key must be pressed. Thereupon the screen will show a table of current values (Fig. 4). Any parameter may be changed, with the exception of geometric characteristics of the tokamak chamber, i.e. RTOR, AB, ELONM, TRICH. Besides it must be kept in mind that if a certain physical parameter may be given in a variety of ways, then its value will be attributed according to the following priorities:

higher - calculation in special subroutines,
↓ input from terminal,
assignment in the model,
assignment in the experimental variant;
assignment from a special file created in the previous run
lower - default value

Thus e.g. the subroutine SETNE (26) re-calculates the boundary neutral density NNCL and NNWM, and therefore in models where this subroutine is used, the values NNCL and NNWM in the table of principal parameters have merely informational value, and changes of these variables from the terminal will be

suppressed at the next subroutine call.

The transport-equation constants and subroutines (Fig. 5) may be changed with the <C> (Constants) key. The meaning of these quantities needs no previous conventions and is determined by the user in the model and in the subroutines. The use of the constants in the model by means of which the calculations are effected, as well as a full description of the model, may be obtained by pressing the <L> (Listing) key.

Enter new variable

AB	2.15	ABC	2.15	ABEAM	0	AIM1	0
AIM2	0	AIM3	0	AMJ	2	AWALL	2.25
BTOR	4.85	CONTR	0	DBM1	0	DBM2	0
DBM3	0	EBEAM	0	ELONG	2	ELONM	2
ENCL	.002	ENWM	.02	HBEAM	0	IPL	22
LEXT	9	NEA	1	NEAVA	0	NECHA	0
NNCL	.001	NNWM	.0001	QBEAM	0	QEB	580.96
QIB	516.39	QNB	0	QTOTA	0	RBMAX	0
RBMIN	0	RTOR	6	SHIFT	0	TEA	0
TIA	0	TRIAN	0	TRICH	.6	UEXT	.1
WEA	0	WHALF	0	WIA	0	WIDTH	0
WPFL	0	WQRAD	0	WTEM	0	WTOTA	0
ZMJ	1						

Fig. 4.

Enter new constant

CF1	1	CF2	1	CF3	1	CF4	1
CF5	1	CF6	1	CF7	1	CF8	1
CF9	1	CF10	1	CF11	1	CF12	1
CF13	1	CF14	1	CF15	1	CF16	1
CV1	0	CV2	0	CV3	0	CV4	0
CV5	0	CV6	0	CV7	0	CV8	0
CV9	0	CV10	0	CV11	0	CV12	0
CV13	0	CV14	0	CV15	0	CV16	0
CBND1	1	CBND2	1	CBND3	1	CBND4	1
CBM1	1	CBM2	1	CBM3	1	CBM4	1
CBMH1	0	CBMH2	0	CBMH3	0	CBMH4	0
CBMR1	0	CBMR2	0	CBMR3	0	CBMR4	0
CBMS1	1	CBMS2	1	CBMS3	1	CBMS4	1
CBMI1	1	CBMI2	1	CBMI3	1	CBMI4	1
CNEUT1	1	CNEUT2	1	CNEUT3	1	CNEUT4	1
CCD1	1	CCD2	1	CCD3	1	CCD4	1
CHE1	1	CHE2	1	CHE3	1	CHE4	1
CIMP1	1	CIMP2	1	CIMP3	1	CIMP4	1
CRAD1	1	CRAD2	1	CRAD3	1	CRAD4	1
CMHD1	1	CMHD2	1	CMHD3	1	CMHD4	1
CPEL1	1	CPEL2	1	CPEL3	1	CPEL4	1
CFUS1	1	CFUS2	1	CFUS3	1	CFUS4	1

Fig. 5.

Enter new times

dRout	1.6	dTout	1.6	dPout	160	Time	.107
TAUmin	.016	TAUmax	16	TAUinc	1.1	DELvar	.1
ITER	1	MeshEq	0	Tinit	0	DeltaT	240

Fig. 6.

The program-control parameters are called (Fig. 6) by the <D> (Delta) key. By means of this table, changes may be made in

- output frequency on the screen dRout
- write-out frequency for the time evolution of simple variables dTout,
- write-out frequency dPout of the radial profiles, for inspection of their evolution in mode 4 and via the command «Profile» from the main menu,
- current time count Time,
- minimal and maximal stepwidth TAUmin and TAUmax,
- control of automatic time-step selection DELvar and TAUinc (see Appendix 2),

- iteration-control parameters ITER (for the main program) and NITER (for the subroutine for calculating penetration of neutrals),
- time scale parameters in modes 5 and 6: Tinit - start-up time, DeltaT -scale,
- number of radial points for equilibrium calculations, MeshEq,
- all parameters for calling subroutines (25).

Besides, in mode 7 the <D> key also changes the time interval for output of calculated trajectories [Tmin,Tmax] and the interval Tmet between markers on these graphs.

For the first run of the model, all the variables from lists that are called by the <V>, <C> or <D> key are given their standard values (unless, of course, they are not determined by some other way). During the calculation, these parameters may be corrected and transferred to the list of constants corresponding to the given model (<I> (Insert) key). Then a new calculation with this model will at once be carried out with the pre-selected parameter values. Besides changes in the calculation parameters, there are three commands for controlling the operation mode of the task. They are <Ctrl><P> (Pause), which transfers the program to a waiting stage, <Ctrl><R> transfers the program to the normal calculation mode, and <Ctrl><C> or </>, which halts the calculation. In the Pause mode, all the control keys are operative, which e.g. makes inspection or output of all pictures possible in one moment of time. Besides, in this mode, a step-by-step calculation may be realized with the <Space> key, which is especially useful in adjusting the models.

For writing the results into a file for subsequent processing, the <F> (File) key is employed. Then the subdirectory DAT will have a file with a name composed of the model name and the experimental variant considered, with an exact copy of the text that will be put out in this mode on the printer via the command <P>.

8. SELECTION OF OUTPUT

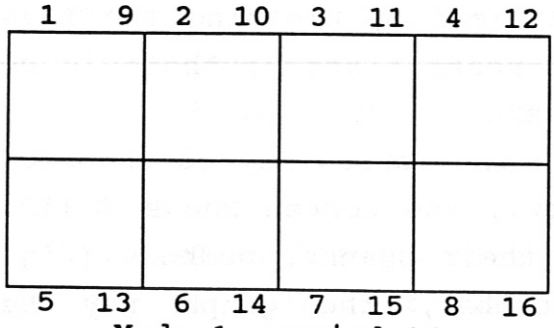
The information output to the screen is effected in several representations (modes). In the basic mode (mode 1) the screen will receive radial profiles as given for the model. Therefore the screen is divided into eight windows, each of which may show two graphs (see Fig. 7). Variables given in lines of the form (29) are printed in numerical form, in the lower lines of the screen. For inspection of certain profiles on a larger scale, output in mode 2 and 3 is used. Here the whole screen is divided into two windows, each of which can show up on 4 curves; in this case, zero means below the picture for mode 2, in the middle for mode 3. An inspection of the radial-profile evolution is possible in mode 4. Then the screen is divided into two windows. Each window is supplied with one family of profiles corresponding to the radial profile in various moments of time already calculated. The name of this profile is selected from the complete list of radial profiles. After termination of the calculation the «Profile» command from the main menu may be employed to inspect the evolution of radial profiles in three-dimensional representation.

For inspection of graphs of simple time-dependent variables, as given by lines of the form (29), modes 5 and 6 are used. In this variant, mode 5 means dividing the screen into 2 horizontal windows, each of which can depict up to 4 graphs, and mode 6 - into four windows with 2 graphs each. The trace of the discharge in the plane of time variables, i.e. the dependence of the variable on the other, may be viewed in mode 7.

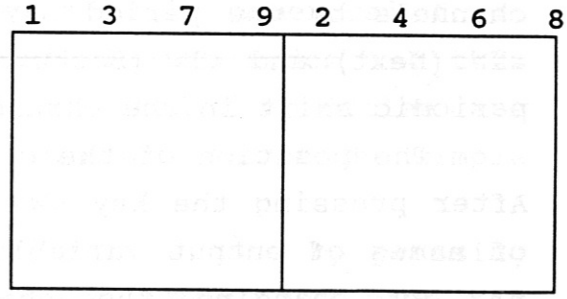
In mode 8, the screen shows the form of magnetic surfaces. In mode 0 (zero), graphs will not be drawn, and the screen show the numerical values of the time variables. The output modes are called by pressing the keys 0 - 8.

As mentioned earlier, the position of a graph of some function on the screen depends on its ordinal number in the model description. The positioning of the graphs, their identifiers and scales in the various modes are given in Fig. 5. For higher-number output channels, the positioning is

Radial profiles



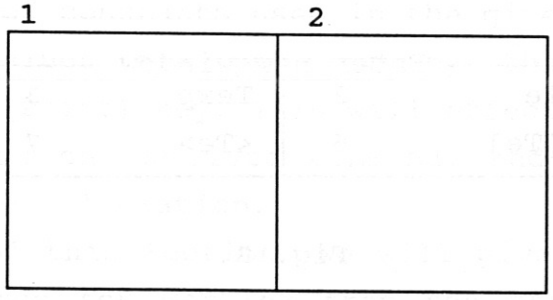
Mode 1, period 16



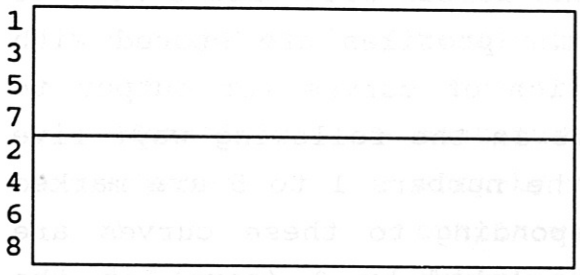
Modes 2 and 3, period 8

Radial profile evolution

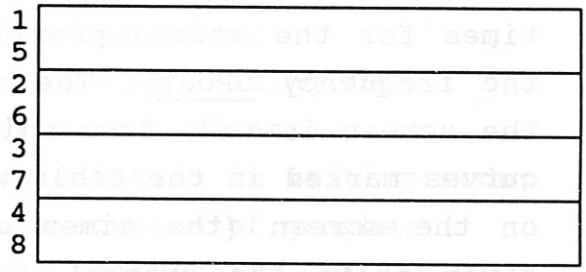
Mode 4



Time evolution



Mode 5, period 8



Mode 6, period 8

One time dependent variable versus another

Mode 7

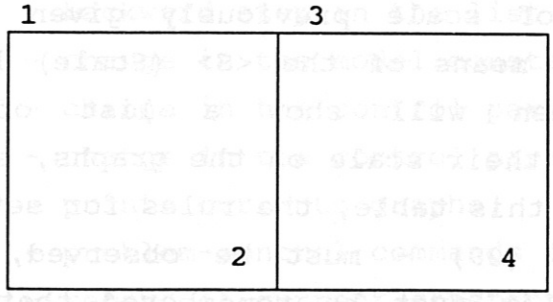


Fig.7

periodically repeated. A shift in the numbers of the output channels by one period may be realized by pressing the keys <N> (Next) and <B> (Backward). In modes 4 and 7, there is no periodic shift in the channel numbers.

The position of the graphs on the screen may be changed. After pressing the key <W> (Window), the screen shows a list of names of output variables and their channel numbers (Fig. 8). By changing the channel number, the graph may be positioned in the required window. Putting the channel number ≤ 0 means that the corresponding profile will not be graphed at all.

Enter new window

q	1	Te	2	Texp	3	Wt	4
tauE	5	[Te]	6	<Te>	7	QTe	8

Fig. 8.

The selection of the moments for inspecting the evolution of radial profiles in mode 4 is effected by pressing the key <M> (Marker) in this mode. Then the screen will show a list of times for the stored profiles (the profiles are stored with the frequency dPout). The selection of curves for output to the screen is made from this list in the following way: five curves marked in the table with the numbers 1 to 5 are marked on the screen (the times corresponding to these curves are given above the graphs); curves marked by 0 (zero) in the table are not marked on the screen and therefore not distinguished from each other; the rest of the curves are not illuminated.

A change of scale previously given in the model is accomplished by means of the <S> (Scale) kay. This implies that the screen will show a list of output-variable identifiers and their scale on the graphs, similar to Fig. 8. When correcting this table, the rules for setting the scales - by SCALE (27), (29) - must be observed, as described in Section 4, and it must be remembered that a missing scale entry in the model (automatic scaling) corresponds to the value "0" in the table of scales.

Beside the graphical presentation, all variables designated for output may be represented in numerical form. For this, the <T> (Type) key is used. In the modes 1-4 this means that the current values of all radial profiles are given, and in the modes 6-7 - the time evolution of the simple variables.

Output of the graphs to the printer is done by pressing the <G> (Graphics) or <Q> (Quality) key. This will make the printer put out an exact copy of the current state of the screen, and also additional information about the calculation, including the name of the model, experimental variant, date, time, and values of constants used in the given calculation.

Numerical values are put out by the printer after pressing the <P> (Print) key. This will effect printing of the same quantities as on the Type command, and also additional information on the calculation.

At the end of this Section, we will give a complete list of control commands for working from the terminal. This may also be brought onto the screen by pressing the <H> (Help) key:

<Ctrl>C or /	- termination of calculation;
<Ctrl>P	- transfer of program to waiting mode;
<Ctrl>R	- continuation of calculations;
<Space>	- step-by-step calculation in waiting mode;
0	- output without execution of graphs;
1-4	- output of radial profiles;
5-6	- output of functions of time;
7	- trajectory in a phase plane;
8	- picture of magnetic surfaces;
B (Back)	- backward step in the list of pictures;
C (Constant)	- change in the model constants;
D (Delta)	- change in the control parameters;
F (File)	- change in the control parameters;
G (Graphics)	- prints currents graphs;
H (Help)	- problem-control commands from terminal;
I (Insert)	- record of current constant values;
L (List)	- prints the original system of equations;
M (Marker)	- change in curve markings;

N (Next) - forward step in list of pictures;
P (Print) - puts numerical values on the printer;
Q (Quality) - prints currents graphs;
R (Refresh) - puts graphs on screen again;
S (Scale) - change in graph scales;
T (Type) - prints on screen in digital mode;
V (Variable) - changes the basic variables;
W (Window) - changes the window number;
X (eXit) - execution of DOS command[s].

8. INSERTION OF USER PROGRAMS INTO ASTRA SYSTEM

The present set of formulas, functions, and subroutines of the system ASTRA may be insufficient for solving problems set up by the user. In this case, the user may create his own formulas, functions, and subroutines, which must satisfy certain conditions of compatibility if they are to joined to the codes constructed for the ASTRA system. To achieve this, the adherence to certain rules is indispensable. We will first look at the rules for creating subroutines.

Rule 1. All transport-code variables must be transferred to subroutines by including general COMMON blocks of ASTRA, deposited in the files STATUS.INC, CONST.INC, e.g.

```
SUBROUTINE NBINJ(ZIMP)
INCLUDE 'STATUS.INC'
INCLUDE 'CONST.INC'
```

further subroutine test.

Rule 2. For traffic of variables not having special names with the main program, the parameter list of the ASTRA system provided unnamed constants (List 1.2 of Appendix 1) and the arrays CAR1-CAR16. To ensure compatibility of unnamed constants and arrays used in the current subroutine, it is, however, desirable to transfer them in explicit form via formal parameters. And besides, to avoid mutual interaction of separate blocks all of them must be described in a commentary to the subroutine (with indication of physical meaning and dimensionality). An example for a text fragment of a model with calling subroutine would be something like

```
! Calculation of the quasilenear tearing mode island width
! Input:  MP - poloidal mode number, NT - toroidal mode number
! Output: XS - island position [m],  WIDTH  - island width [m]
      ISLAND(MP,NT,XS,WIDTH):
      Pos_XS;                  Wid_WIDTH;
```

Rule 3. In subroutines it is forbidden to re-assign the variables marked by asterisks in List 1.2 of Appendix 1.

Rule 4. To avoid overflow of memory, we recommend description of very large subroutine arrays in COMMON blocks.

The rules for creating functions are analogous to the rules for creating subroutines. There are, however, additional requirements, connected to the use of functions in lines that are analyzed by the program generator.

Rule 5. In the ASTRA system, a function may have only one parameter, having the meaning of the current value of the variable ρ (radius).

Rule 6. In the function text, its identifier is supplemented by the letter "R" after the name in the function list of the ASTRA complex. Thus, e.g., the function names WE and BETAP in the text and identifiers appear as WER and BETAJR:

```
REAL FUNCTION WER (R)
INCLUDE  'CONST.INC'
.....
WER=...
END
```

The rules for composing formulas are really simple:

Rule 7. Formulas appear as blocks of FORTRAN text for calculating a certain quantity. They do not require any description. It is not convenient to introduce other variables into the formulas, with the exception of the one that corresponds in name to the formula name. Now if the introduction of a new variable appears unavoidable, it is recommended to begin the variable name with a "Y". The name must in no case coincide with names of standard parameters, functions, or formulas within the ASTRA system.

ACKNOWLEDGMENTS

We are grateful to Mr. J. Eisert for translation of the paper from Russian to English and one of us (G.V.P.) wishes to express his appreciation to NET-team and to Tokamak Theory Division of Max-Planck-Institut fur Plasmaphysik for help and support received during this work.

APPENDIX 1. STANDARD SPECIFICATIONS FOR THE ASTRA SYSTEM

1.1 Basic parameters of machine and discharge: The units (dimensions) are given in square brackets. This parameter list may be viewed on the screen via the «View» - «Variables» command.

AB	[m] - limiter position along the horizontal axis
ABC	[m] - calculation limit for the horizontal axis
ABEAM	[ ] - relative atomic mass of beam atoms (H-1, D-2)
AMJ	[ ] - (mass of main ions)/Mp
AMN	[ ] - (mass of other than main ions)/Mp
AWALL	[m] - wall position on the horizontal axis
AIM1-3	[ ] - (atomic mass of impurities 1-3)/Mp
BTOR	[T] - toroidal vacuum field in chamber centre
CONTR	[ ] - Co=-1, Contr=+1 direction of NBI
ELONG	[ ] - extension of the limiting magnetic surface
ELONM	[ ] - extension of the chamber
ENCL	[keV] - energy of cold neutrals at the boundary
ENWM	[keV] - energy of warm neutrals at the boundary
IPL	[MA] - total plasma current
LEXT	[mkHn]- external inductance
NEA	[$10^{19}/m^3$] - electron density at the boundary
NEAVA	[$10^{19}/m^3$]- mean (volume average) electron density
NECHA	[$10^{19}/m^3$] - mean (averaged over the central vertical chord) electron density
NNCL	[$10^{19}/m^3$]- limit density of cold neutrals
NNWM	[$10^{19}/m^3$]- limit density of warm neutrals
QBEAM	[MW] - total beam power
QEB	[MW] - electron heat flux at the boundary
QIB	[MW] - ion heat flux at the boundary
QNB	[$10^{19}/s$] - particle flux at the boundary
QTOTA	[MW] - total power input into the plasma
RTOR	[m] - major radius (to the chamber centre)
SHIFT	[m] - shift of the limiting magnetic surface
TEA	[keV] - electron temperature limit
TIA	[keV] - ion temperature limit
TRIAN	[m] - triangularity of the limiting magnetic surface

TRICH [m] - triangularity of the chamber
UEXT [V] - experimental peripheral voltage
WEA [MJ] - energy content of electrons
WHALF [a.u.] - H-alpha radiation to the wall
WIA [MJ] - energy content of the ions
WIDTH [m] - width of SOL
WPFL [10^{19}/s] - particle flux to the wall
WQRAD [MW] - radiation in SOL
WTEM [keV] - temperature of divertor or limiter plates
WTOTA [MJ] - total energy content of plasma
ZMJ [ ] - charge of the main ion component

1.2 Free constants not having a pre-defined physical meaning: they may be used for control of the transport code and for substitution in describing additional effects, in the corresponding sub-routines (see also paragraph 8 below).

CF1:16 - entire model (default values at the beginning =1)
CV1:16 - entire model (default values at the beginning =0)
CBND1:4 - boundary
CBM1:4 - injection of neutral beam
CBMH1:4
CBMR1:4
CBMS1:4
CBMI1:4
CNEUT1:4 - neutrals
CCD1:4 - current drive
CHE1:4 - He thermal conductivity coefficients
CIMP1:4 - impurities
CRAD1:4 - radiation
CMHD1:4 - MHD modes
CPEL1:4 - pellet injection
CFUS1:4 - fusion

1.3 Radial profiles: variables that are always defined within the problem are marked with an asterisk (*). a list of these variables may also be moved to the screen, via the command «View» - «Arrays».
*AMAIN(r) [ ] - (mass of main ion component)/Mp;

*AMETR(r) [m] - horizontal radius of magnetic surface;

CAR1-16(r) [ ] - free variables for substitution in subroutines;

*CU(r) [MA/m^2] - current density;

CUBM(r) [MA/m^2] - current density generated by a beam of suprathermal ions, with inclusion of screening electron current and toroidal corrections F_{te} for trapped electrons, $(1-Z_{eff}^{-1}F_{te}) \int e_b v_\parallel f_b d^3v$;

CUBTS(r) [MA/m^2] - bootstrap current;

CUFI(r) [MA/m^2] = $\int e_b v_\parallel f_b d^3v$ - current density of suprathermal ions;

CUX(r) [MA/m^2] - experimental current density;

*ELON(r) [ ] - elongation of magnetic surface;

*FP(r) [V·s] - poloidal flux;

*FPO(r) [V·s] - poloidal flux in preceding time step;

*G22G(r) [m] = $\langle(\nabla\rho/r)^2\rangle \cdot (B_0 R_0^2 V'/(4\pi^2 I)$;

*G2GR2(r) [m^2] = $V' \langle(\nabla\rho)^2\rangle$;

*GN(r) [$10^{19}/m^2/s$] - particle flux;

GNX(r) [$10^{19}/m^2/s$] - experimental particle flux;

*IPOL(r) [MA] = $I/(B_0 R_0)$ - normalized diamagnetic current;

*MU(r) [ ] - rotational transform (1/q);

NALF(r) [$10^{19}/m^3$]- alpha particle density;

*NE(r) [$10^{19}/m^3$]- electron density;

*NEO(r) [$10^{19}/m^3$] - electron density in preceding time step;

NEX(r) [$10^{19}/m^3$] - experimental electron density;

*NI(r) [$10^{19}/m^3$] - summary ion density;

NIBM(r) [$10^{19}/m^3$] - $\int f_b d^3v$ - density of suprathermal ions;

NIMN(r) [$10^{19}/m^3$] - density of additional ions;

NIZ1-3(r) [$10^{19}/m^3$] - density of impurity sorts 1-3;

NN(r) [ ] - relative neutral density;

NNBM1(r) [$10^{19}/m^3$] - density of beam neutrals with energy EBEAM;

NNBM2(r) [$10^{19}/m^3$] - density of beam neutrals with energy EBEAM/2;

NNBM3(r) [$10^{19}/m^3$] - density of beam neutrals with energy EBEAM/3;

NTRIT(r) [$10^{19}/m^3$] - density of tritium;

*NUEST(r) [ ] - relative electron collision frequency;

*NUIST(r) [ ] - relative ion collision frequency;

PBEAM(r) [MW/m^3] - power density of energy contribution to the main plasma;

PBLON(r) [10^{19}keV/m^3] = $\int M_b v_{\parallel}^2 f_b d^3v$ - beam pressure along the magnetic filed;

PBPER(r) [10^{19}keV/m^3] = $\int M_b v_{\perp}^2 f_b d^3v/2$ - beam pressure transverse to the magnetic field;

PEBM(r) [MW/m^3] - power density of the beam energy contribution to the thermal electrons;

*PETOT(r) [MW/m^3] - total input power for the electrons;

PEX(r) [MW/m^3] - experimental total power for the electrons;

PIBM(r) [MW/m^3] - power density for the beam energy contribution to thermal ions;

*PITOT(r) [MW/m^3] - total input power for the ions;

PIX(r) [MW/m^3] - experimental radiation power;

*R2MTR(r) [ ] = $R_0^2 \cdot \langle 1/r^2 \rangle$;

*ROMTR(r) [m] - the experimental value of the radius of the current magnetic surface;

SCUBM(r) [MA/m^2/s] - current source due to the beam;

*SHIF(r) [m] - the shift of the magnetic surface;

SNEMB(r) [10^{19}/m/s] - source of thermal neutrals formed in charge exchange of the neutral beam;

SNNBM(r) [10^{19}/m/s] - source of thermal neutrals formed in charge of exchange of the neutral beam with thermal ions;

*SNTOT(r) [10^{19}/m/s] - total particle source;

SNX(r) [10^{19}/m/s] - experimental particle source;

*SQEPS(r) [ ] = $\sqrt{\varepsilon}$, ε is inverse aspect ratio;

SQG11(r) [ ] - reserved;

*TE(r) [keV] - electron temperature;

*TEO(r) [keV] - electron temperature in preceding time step;

TEX(r) [keV] - experimental electron temperature;

*TI(r) [keV] - ion temperature;

*TIO(r) [keV] - ion temperature in the preceding time step;

TIX(r) [keV] - experimental ion temperature;

TN(r) [keV] - neutral temperature;

*TRIA(r) [m] - triangularity of magnetic surface;

*UT(r) [v] - peripheral voltage;

*VNEWR(r) [m^2] - $dV/d\rho$ in the current time step;

*VOLDR(r) [m^2] - $dV/d\rho$ in the preceding time step;

*VP(r) [m/s] = $c*E_{\parallel}/B_p$ - pinch velocity;

VPFP(r) [m/s] =$(\partial\rho/\partial t)_{\Phi}$ - movement of the Φ = Const surface under adiabatic compression;

VTOR(r) [10^6m/s] - toroidal rotation rate;

*ZEF(r) [] - Z_{eff};

ZEFX(r) [] - experimental Z_{eff};

*ZMAIN(r) [] - charge of the main ion component;

1.4 Library formulas: This library is constantly being updated. Each ASTRA system user as a rule contributes his own formulas. Their list and complete information on each formula may be obtained via the «View» - «Formula» command. Below, we give the most general formulas.

BETPL(r) [ ] = $4\pi\cdot Ne\cdot Te/B_{pol}^2$ - β poloidal;

CCNEU(r) [MA/(V·m)] conductivity due to electron-neutral collisions [2];

CCSP(r) [MA/(V·m)] = $70\cdot T_e^{3/2}(1.1+Z_{eff})/(2.66+Z_{eff})Z_{eff}$ - classical Spitzer plasma conductivity, [2];

CCSPX(r) [MA/(V·m)] = $70\cdot T_{ex}^{3/2}(1.1+Z_{eff})/(2.66+Z_{eff})Z_{eff}$ - classical Spitzer conductivity, calculated from the experimental temperature, [2];

CHOTF(r) [MA/(V·m)] = $2\pi\cdot T_e\cdot(5.+Z_{eff})/(3.+Z_{eff})/N_e$ - hot conductivity after Fisch;

CNHH(r) [MA/(V·m)] = CCSP*$(1-\sqrt{\varepsilon}\cdot K_{33})$ - neoclassical conductivity after Hinton-Haseltine [2];

CNHR(r) [MA/(V·m)] - neoclassical conductivity after Hirshman [5];

DCHH(r) [ ] - neoclassical transport coefficient after Hinton and Haseltine DC=DCHH, [2];

DNEXP(r) [m^2/s] = -QNTOT/(V'·∇Nex) - experimental diffusion coefficient;

EFLHN(r) [V/m] - η_N, current generation efficiency from lower-hybrid waves with narrow spectrum. Appears in the formula $P_{hf}(MW/m^3)=j_{hf}(MA/m^2)*(\eta_N-U/2\pi/(R+\Delta))$;

EFLHW(r) [V/m] - η_W, current generation efficiency for lower-hybrid waves with wide-band spectrum. Appears

in the formula $P_{hf}(MW/m^3)=j_{hf}(MA/m^2)*(\eta_W-U/2\pi/(R+\Delta))$;

ENHHO(r) []= K_0, - neoclassical transport coefficient after Hinton and Haseltine DC=DCHH, [2];

DCHH(r) [] - neoclassical transport coefficient after Hinton and Haseltine DC=DCHH, [2];

ENHH1(r) [] - $\sqrt{\varepsilon}\cdot K_{13}$ neoclassical transport coefficient after Hinton and Haseltine [2];

ENHH2(r) [] - $\sqrt{\varepsilon}\cdot K_{23}$ neoclassical transport coefficient after Hinton and Haseltine [2];

ET(r) [V/m] - electric field;

ETAE(r) [ ] = $d\ln(T_e)/d\ln(n)$;

ETAI(r) [ ] = $d\ln(T_i)/d\ln(n)$;

FLIN(r) [ ] = r/a - linear function;

FPR(r) [ ] = $1-(r/a)^2$ - parabolic profile;

FR(r) [m] = r - radius;

HAALC(r) [m^2/s] = $5\cdot10^{17}/NE$ - anomalous ALCATOR thermal conductivity, [7];

HABOM(r) [m^2/s] = $\dfrac{c\cdot T_e}{16\cdot e\cdot B}$ - Bohm thermal conductivity;

HACTE(r) [m^2/s] - anomalous thermal conductivity in the CTE mode [8];

HAED(r) [m^2/s]= $2.5\cdot\sqrt{\varepsilon}\cdot c\cdot Te\cdot\rho_s/(e\cdot B_t\cdot L_n)$ - anomalous thermal conductivity in the electron drift mode, [9];

HAETI(r) [m^2/s]= $\frac{5}{2}\omega_*/k^2\cdot[\eta_i\cdot 2T_I/T_e\cdot L_n/R]^{1/2}\cdot\exp(-(\eta_{crit}/\eta_i)^4)$ anomalous thermal conductivity in the η_i mode, [8];

HAMM(r) [m^2/s] = $5\cdot10^{19}\varepsilon^{1.75}\sqrt{T_e/A}/(n_e qR)$ - anomalous thermal conductivity after Merezhkin and Mukhovatov, [10];

HANAL(r) [m^2/s] = $\varepsilon/(q\cdot R\cdot n_e\cdot\sqrt{A})$ - anomalous neo-ALCATOR thermal conductivity;

HAPA(r) [m^2/s] = $c\cdot T_e\cdot\rho_s\cdot|1/L_n-0.5/L_T|/e\cdot Bt$ - anomalous thermal conductivity after Parail, [11];

HAPYU(r) [m^2/s] = $c^2/\omega_p^2\cdot V_{Te}\cdot\varepsilon/(q\cdot R)$ - anomalous thermal conductivity after Parail and Yushmanov, [12];

HAQ1(r) [m^2/s] = $\begin{cases}8\cdot10^{19}/n_e/R, & q\le 1\\ 0, & q>1\end{cases}$ - anomalous thermal conductivity inside the surface, q = I, [8];

HAQ1C(r) [m^2/s] = $\begin{cases}1, & q\le 1\\ 0, & q>1\end{cases}$ - anomalous thermal

conductivity inside the surface, q = I;

HARNQ(r) $[m^2/s] = \rho_s^2 \cdot \nu_e \cdot q^2$ - anomalous thermal conductivity with ρ_s, ν, q [13];

HARPL(r) $[m^2/s] = 0.01 \cdot q^2 \cdot n^{1/3}/T_e^{5/6}$ - rippling mode induced heat conductivity [14];

HASCL(r) $[m^2/s] = c^2/\omega_p^2 \cdot \nu_{ee} \cdot (1+1.4 \cdot Z_{eff})$ - anomalous thermal conductivity of the Ohkawa type, [15];

HCHH(r) [ ] - neoclassical transport coefficient after Hinton and Haseltine, [2];

HCHR(r) [ ] - neoclassical transport coefficient after Hirshman, [6];

HEEXP(r) $[m^2/s] = (QE_{tot} - dW_e/dt)/(V' \cdot Ne \cdot \nabla Te)$ - experimental thermal conductivity;

HEGN(r) $[m^2/s] = (QE_{tot} - QE_{conv})/(V' \cdot Ne \cdot \nabla Te)$ - experimental thermal conductivity without convection;

HEXP(r) $[m^2/s] = QE_{tot}/(V' \cdot n_e \cdot \nabla Te)$ - electron heat conductivity after experimental data;

HNGSB(r) $[m^2/s] = \sqrt{\varepsilon} \cdot \rho_i^2 \cdot \nu_i \cdot 0.57 \quad (\nu_i = \nu_{ii} \cdot (1+1.4 \cdot (Z_{eff}-1)))$ - neoclassical ion thermal conductivity in the banana regime after Galeev-Sagdeev, [16];

HNGSE(r) $[m^2/s]$ - neoclassical electron thermal conductivity in the banana regime after Galeev-Sagdeev, [16];

HNGSI(r) $[m^2/s]$ = 1./(1./HNGSP+1./HNGSB)+HNPSI - neoclassical ion thermal conductivity after Galeev-Sagdeev, [16];

HNGSP(r) $[m^2/s] = 1.3 \cdot q \cdot \rho_i^2 \cdot V_{Ti}/R$ - neoclassical ion thermal conductivity in the plateau regime after Galeev-Sagdeev, [16];

HNPSI(r) $[m^2/s] = \rho_i^2 \cdot \nu_i \cdot (1+1.6 \cdot q^2), (\nu_i = \nu_{ii} \cdot (1+1.4 \cdot (Z_{eff}-1)))$ - neoclassical ion thermal conductivity in the Pfirsch-Schluter regime, [17];

I*(r) $[MA] = \dfrac{IPOL(r)}{2\pi R_0} \int_0^r CU^*(r)/IPOL(r)^2 \cdot V' \, dr$ - total plasma current inside magnetic surface r = ρ = Const. Formulas for IBM (beam current), IBTS (bootstrap), ICD (driven current), IOHM (Ohmic), ITOT (summary current) are available;

NUE(r) $[s^{-1}] = \nu_{ee} \cdot (1+1.4 \cdot Z_{eff})$ - total electron

collision frequency;

NUEE(r) [s^{-1}] = $4\sqrt{\pi}/3 \cdot n_e \cdot e^4 \cdot \lambda/\sqrt{M_e}/Te^{1.5}$ - frequency of electron-electron collisions;

NUI(r) [s^{-1}] - total frequency of ion collisions;

NUIS(r) [ ] = $\nu_i \cdot q \cdot R_t/(\varepsilon^{3/2} \cdot v_{Te})$ = ν_i^* ;

NUPP(r) [s^{-1}] = $4 \cdot \sqrt{\pi}/3 \cdot Ni \cdot e^4 \cdot \lambda/\sqrt{M_i}/Te^{1.5}$ - proton-proton collisional frequency (λ=15);

PBDHE(r) [MW] - neutral beam induced fusion power;

PBRAD(r) [MW/m^3] = $5.06 \cdot 10^{-5} \cdot Z_{eff} \cdot n_e^2 \cdot T_e^{1/2}$ - electron bremsstrahlung, [18];

PDT(r) [MW/m^3] - α-heating, [19];

PEDT(r) [MW/m^3] - α-heating of electrons, [20];

PEGN(r) [MW/m^3] = $5/2 \cdot div(Te \cdot \Gamma_n)$ - power density of the energy input to the electron due to the particle flux;

PEI(r) [$MW/m^3/keV$] = $3/\tau_{ei} \cdot Ne \cdot m/M$ - electron-ion heat exchange due to Coulomb collisions when written in implicit form ($\tau_{ei}=7.3 \cdot 10^{-8} \cdot Te^{1.5}/(Z^2 \cdot Ni)$);

PE =...+ PEI*TI; PET =...- PEI;

PI =...+ PEI*TE; PIT =...- PEI;

PEICL(r) [MW/m^3] = $3/\tau_{ei} \cdot Ne \cdot (Te-Ti) \cdot m/M$ - electron-ion heat exchange due to COULOMB collisions, PE =...-PEICL, PI =...+PEICL, ($\tau_{ei}=7.3 \cdot 10^{-8} \cdot Te^{1.5}/(Z^2 \cdot Ni)$);

PEIGN(r) [MW/m^3] = $-\Gamma_n/Ne \cdot \nabla(Ne \cdot Ti)$ - electron-ion heat exchange due to flux of particles,

PE = PEIGN, PI = -PEIGN;

PENEU(r) [MW/m^3] = $\langle\sigma \cdot v\rangle \cdot Ne \cdot Nn \cdot (0.013keV)$ - electron thermal losses due to ionization of cold neutrals,

PE =...- PENEU;

PENLI(r) [MW/m^3] = $\langle\sigma \cdot v\rangle \cdot Ne \cdot Nn \cdot (0.0102keV)$ - neutral radiation power density, PE =...- PENLI;

PICX(r) [MW/m^3] = $3/2 \cdot \langle\sigma \cdot v\rangle \cdot Ni \cdot Nn \cdot (Ti-Tn)$ - ion heat source due to charge exchange with cold neutrals,

PI = -PICX +...;

PIDT(r) [MW/m^3] - α-heating of ions, [20];

PIGN(r) [MW/m^3] = $5/2 \cdot div(Ti \cdot \Gamma_n)$ - power input to the ions due to particle flux, PI = PIGN +...;

PINEU(r) [MW/m^3] = $3/2 \cdot Nn \cdot [\langle\sigma_e \cdot v\rangle \cdot Ne \cdot Tn + \langle\sigma_i \cdot v\rangle \cdot Ni \cdot Tn - \langle\sigma_{cx} \cdot v\rangle \cdot Ni \cdot (Ti-Tn)]$ - ion heat source from

ionization and charge exchange, PI = ...+PINEU;

PIONZ(r) [MW/m^3] = 3/2·Nn·[<σ_e·v>·Ne·Tn + <σ_i·v>·Ni·Tn - ion heating due to ionization PI = PIONZ+...;

PITCX(r) [MW/m^3/keV] = 3/2<σ·v>·Ni·Nn - ion heat source from cold-neutron charge exchange, when written in implicit form, PIT=PITCX, PI=PITCX·TN;

PJOUL(r) [MW/m^3] = j·E - Joule heating power density;

POH(r) [MW·m^3] = $\sigma_\parallel \cdot E^2$ - Joule heating power density;

PRCAR(r) [MW·m^3] - Carbon radiation power density, [21];

PRFER(r) [MW·m^3] - Iron radiation power density, [21];

PROXI(r) [MW·m^3] - Oxygen radiation power density, [21];

PRWOL(r) [MW·m^3] - Wolfram radiation power density, [21];

PSYNC(r) [MW/m^3] = $C_s \cdot \langle N_e \rangle \cdot \langle T_e \rangle \cdot B_0^2 \cdot \Phi_s$ - electron synchrotron radiation ($C_s \approx 6.2 \cdot 10^{-2}$), Φ_s /- transparency factor;

PTOT(r) [MW/m^3] = $P_e + P_i$ - total heating power density;

Q*(r) [MW] = $\int_0^r P^*(r) \cdot V' dr$;

SNNEU(r) [10^{19}/m^3/s] = Ne<σ_ev>Nn + Ni<σ_iv>Nn -<σ_{rec}v>Ne^2- particle density of a source due to ionization of neutrals by electron and ion collisions, SNN = SNNEU +...;

SNNIE(r) [10^{19}/m^3/s] = <σ_e·v>·Nn·Ne - density of particle source to ionization by electron collisions, SNN = SNNIE +...;

SNNII(r) [10^{19}/m^3/s] = <σ_i·v>·Nn·Ni - density of particle source due to ionization by collisions with ions, SNN = SNNII +...;

SNNR(r) [10^{19}/m^3/s] = <σ_{rec}·v>·Ne^2 - particle sink due to recombination, SNN = -SNNR +...;

SURF(r) [m^2] - magnetic surface area;

SVCX(r) [10^{19}m^3/s] - <σ_{cx}·v> charge exchange, [22];

SVD1(r) [10^{19}m^3/s] - <σ·v> reaction L + D = n + ^{3}He, [19];

SVD2(r) [10^{19}m^3/s] - <σ·v> reaction D + D = t + p, [19];

SVDHE(r) [10^{19}m^3/s] - <σ·v> reaction D + ^{3}He = α + p, [23];

SVDT(r) [10^{19}m^3/s] - <σ·v> reaction D + T = α + n, [19];

SVIE(r) [10^{19}m^3/s] - <σ_i·v> ionization by electron collisions, [22];

SVII(r) [10^{19}m^3/s] - <σ_i·v> ionization by ion collisions,

[22];

SVREC(r) [$10^{19}m^3/s$] - $\langle\sigma_{rec}\cdot v\rangle$ recombination;

TAU89(r) [s] - ITER-scaling of 1989;

TAUNA(r) [s] = $0.007\cdot q\cdot a\cdot R^2\cdot\langle N_e\rangle$ - Neo-ALCATOR scaling;

TITER(r) [s] - - ITER-scaling of 1989;

VTI(r) [10^6cm/s] = $T_i^{1/2}/(Am_p)^{1/2}$ - ion thermal velocity;

WE(r) [MJ] = $3/2\int_0^r Te\cdot Ne\cdot V'dr$ - energy content in electrons;

WI(r) [MJ] = $3/2\int_0^r Ti\cdot Ne\cdot V'dr$ - energy content in ions;

WTOT(r) [MJ] = $3/2\int_0^r (Te+Ti)\cdot Ne\cdot V'dr$ - total energy content;

XCHH(r) [ ] - neoclassical transport coefficient after Hinton and Haseltine [2], XC = XCHH +...;

XCHR(r) [ ] - neoclassical transport coefficient after Hirshman [6], XC = XCHR +...;

XEXP(r) [m^2/s] = $QI_{tot}/V'\cdot Ni\cdot\nabla Ti$ - ion thermal conductivity after experimental data;

XIEXP(r) [m^2/s] = $(QI_{tot}-dW_i/dt)/V'\cdot Ni\cdot\nabla Ti$ - ion thermal conductivity after experimental data;

XIGN(r) [m^2/s] = $(QI_{tot}-QI_{conv})/(V'\cdot Ni\cdot\nabla Ti)$ - experimental thermal ion conductivity without convection;

ZICAR(r) [ ] - mean charge of carbon impurities [21];

ZIFER(r) [ ] - mean charge of iron impurities [21];

ZIOXI(r) [ ] - mean charge of oxygen impurities [21];

ZIWOL(r) [ ] - mean charge of tungsten impurities [21];

1.5 List of functions: they are employed just like formulas, but organized as separate subroutines. Their list and full information on each function may be obtained by means of the command «View» - «Function».

BETAJ(r) [] $= \frac{2c^2}{J_{pl}^2}(\int pdS - p)$ - poloidal beta;

BETBM(r) [] - beta of the poloidal beam;

IBM(r), IBTS(r), ICD(r), IOHM(r), ITOT(r), [MA] - partial currents inside magnetic surface r;

LINT(r) [ ] $= \frac{2}{r^2 B_{pol}^2}\int_0^r rB_{pol}^2 V'dr$ - internal inductance;

NEAV(r) $[10^{19}/m^3] = \frac{1}{V}\cdot\int_0^r Ne(r)\cdot V\,d\rho$ – volume-averaged density;

NECH(r) $[10^{19}/m^3] = \frac{1}{a}\cdot\int_0^r Ne(r)dr$ - line-averaged density;

NEXAV(r) $[10^{19}/m^3] = \frac{1}{V}\cdot\int_0^r Ne(r)\cdot Vdr$ - average density on the central chord;

PTOT(r) $[MW/m^3]$ = PETOT+PITOT - summary density of energy sources in the plasma;

Q*(r) $[MW] = \int_0^r P^*(r)\cdot V'dr$ - volume integrals. The next functions are available:

QBTOT(r), QDT(r), QEDT(r), QEGN(r), QEICL(r), QEIGN(r), QETOT(r), QEX(r), QIDT(r), QIGN(r), QINEU(r), QITOT(r), QIX(r), QJOUL(r), QOH(r), QRAD(r), QRADX(r), QTOT(r);

QEDWT(r) [MW] = dWe/dt;

QENEU(r) $[10^{19}s^{-1}] = \int_0^r SNNEU\cdot V'dr$ - total particle source due to ionization of neutrals by electron and ion collisions;

QIDWT(r) [MW] = dWi/dt;

QNDNT(r) $[s^{-1}] = d/dt\int_0^r Ne\cdot Vd\rho$ - rate of change of total particle number;

QNTOT(r) $[10^{19}s^{-1}] = \int_0^r SNTOT \cdot V' dr$ - external particle source;

QNX(r) $[10^{19}s^{-1}] = \int_0^r SNX \cdot V' dr$ - external particle source;

TAUE(r) $[s] = \int_0^r W_{tot} \cdot V \, dr / (QTOT - dW_{tot}/dt)$ - total energy confinement time;

TAUEE(r) $[s] = \int_0^r W_e \cdot V dr / (QETOT - dW_e/dt)$ - electron energy confinement time;

TAUEI(r) $[s] = \int_0^r W_i \cdot V dr / (QITOT - dW_i/dt)$ - ion energy confinement time;

TAUG(r) $[s] = W_{tot} \Big/ \left[\int_0^r (PEX+PIX+PJOUL) \cdot V' dr - dW_{tot}/dt \right]$ - global energy confinement time;

TAUIG(r) $[s] = W_i \Big/ \left[\int_0^r (PIX+PEICL) \cdot V' dr - dW_i/dt \right]$ - global ion energy confinement time;

TAUP(r) $[s] = \int_0^r Ne(r) \cdot V' dr \Big/ \left[QNTOT - d/dt \left(\int_0^r Ne(r) \cdot V' dr \right) \right]$ - particle confinement time;

TEAV(r) $[keV] = \frac{1}{V(r)} \cdot \int_0^r Te(r) \cdot V' dr$ - volume-averaged electron temperature;

TENDN(r) $[keV] = \int_0^r Te \cdot Ne \cdot dV \Big/ \int_0^r Ne \cdot dV$ - density-averaged electron temperature;

TEXAV(r) $[keV] = \frac{1}{V(r)} \cdot \int_0^r T_e^{exp}(r) \cdot V' dr$ - experimental

volume-averaged electron temperature;

$TIAV(r)$ [keV] $= \frac{1}{V(r)} \cdot \int_0^r Ti(r) \cdot V' dr$ - volume-averaged ion temperature;

$TINDN(r)$ [keV] $= \int_0^r Ti \cdot Ne \cdot dV \Big/ \int_0^r Ne \cdot dV$ - - density-averaged ion temperature;

$TIXAV(r)$ [keV] $= \frac{1}{V(r)} \cdot \int_0^r T_i^{exp}(r) \cdot V' dr$ - external volume-averaged ion temperature;

$WE(r)$ [MJ] - electron content;

$WI(r)$ [MJ] - ion energy content;

$WTOT(r)$ [MJ] - plasma energy content;

$WTOTX(r)$ [MJ] - external plasma energy content;

$ZNDN(r)$ [ ] $= \int_0^r Z_{eff} \cdot Ne \cdot V' dr \Big/ \int_0^r Ne \cdot V' dr$ - averaged Z_{eff}.

1.6 Special functions for convenience in calculations:

GRAD(Arg) - gradient (argument Arg - variable from list 1.3)

FJUMP(T1) and FRAMP(T1,T2) - unit step function in the point T1 and linear increase from 0 in point T2

$$FJUMP(T1) = \begin{cases} 0, & TIME<T1 \\ 1, & TIME>T1 \end{cases},$$

$$FRAMP(T1,T2) = \begin{cases} 0, & TIME<T1 \\ (TIME-T1)/(T2-T1), & T1<TIME<T2; \\ 1, & TIME>T2 \end{cases}$$

$VINT(Arg)(r) = \int_0^r Arg \cdot V' dr$ - integral over the volume;

$IINT(Arg)(r) = \frac{IPOL(r)}{2\pi R_0} \int_0^r Arg/IPOL(r)^2 \cdot V' dr \cdot IPOL(r)/(2\pi R_0)$ - integral of current density over the toroidal cross section;

$TimDer(Arg) = d/dt(Arg)$ - time derivative;

$TimInt(Arg) = \int_0^{TIME} Arg \cdot dt$ - time integral.

1.7 Parameters of numerical calculations

HRO [m] - step along radius;
ROB [m] - position of limiter (along ρ);
ROWALL [m] - position of wall (along ρ);
ROC [m] - limit of calculation (along ρ);
VOLUME [m^3] - total volume;
GP = 3.1415926 (GP2=2*GP);
NA1 - number of mesh point inside a plasma (NA=NA1-1);
NB1 - full number of mesh points;
NS - number of the point where boundary flux is calculated;
DROUT [s] - time interval for output of radial profiles;
DTOUT [s] - time interval for output of functions of time;
DPOUT [s] - time interval for storing radial profiles;
TAUMIN [s] - minimal step in time;
TAUMAX [s] - maximal step in time;
TAUINC [] - maximal increment of time step increase in time (=TAUold/TAUnew);
DELVAR [] - maximal relative change of the variables within one time step;
TIME [s] - current time in the calculation;
TAU [s] - time step;
ITER(RITER) - number of iterations in the numerical scheme;
NITER(RNIT) - number of generations of neutrals;
MESHEQ - number of mesh points in the equilibrium code;
DEQLBR [s] - time interval for calls of the EQLIBR subroutine;
DTEQ(J) [s] - time interval for calling the J subroutine;
BEEQ(J) [s] - time of first call of the J subroutine;
ENEQ(J) [s] - termination time for calling of the /J/ subroutine;
KEQ(J) - ASCII code of the key after which the J subroutine is called.

1.8 List of subroutines

CONIPL(IST,IEN)- provides rise of total plasma current IPL

with velocity 0.1 MA/s < dI/dt < 1 MA/s/ up to the value IEN. The rise rate of the current is selected from the condition q(0) < q(a). IST - blind parameter

GNXSRC - calculated the stationary particle flux GNX due to the ionization of neutral atoms

MIXE(GTEC,TDS,HARS)- model for central disruption

ISLAND(M,NM,XS,W) - calculation of the quasilinear tearing mode island width [24].

Input: Output:

M - poloidal mode number; XS [m] - island position;

NM - toroidal mode number; W [m] - island width;

LHCD(QLH,CULH,PLH,NR1,ALFA,NLH,NV1,PASNUM,MUTE) - calculates the current drive and plasma heating by lower hybrid (LH) waves. All parameters are real! CULH and PLH are arrays given for the transport problem mesh.

Input parameters:

1. QLH [MW] - launched power
2. NLH - position of maximum in the model LH wave spectrum:

$$P(N_{\parallel}) = P_0 \exp\left[- \left[\frac{N_{\parallel}-NLH}{\Delta_1}\right]^2 \right] \cos^2\left[\pi \frac{N_{\parallel}-NLH}{\Delta_2}\right],$$

$\Delta_1 = 0.3$, $\Delta_2 = 0.8$, P_0 is a normalized constant.

3. ALFA - diffusive broadening parameter for the wave spectrum during propagation in plasma (no broadening for ALFA=0)
4. CCD2 - fitting parameter for adjusting calculated efficiency to experimental values. It also takes into account 2D effects. 5. CCD1 - stop-iteration condition in multi-passage absorption. Absorbed power is just the fraction CCD1 of the radiated power: $P_{abs} \geq CCD1 \cdot P_{launch}$ (CCD1 < 1);
6. CCD3 - gives the fraction of power lost due to reflection of the wave from the wall in multi-passage absorption (CCD3 ≤ 1);
7. NR1 - number of radial mesh points (usually NR1>20);
8. NV1 - number of velocity space mesh points (usually NV1>40);
9. MUTE - control parameter for output of interim results in graphic form: MUTE=0 means no output, MUTE>10

effects output of LH current profiles and power absorption for each passage of the wave along the radius;

Output parameters:

1. CULH [MA/m^2] - LH current density;
2. PLH [MW/m^3] - absorbed-power density;
3. CAR1 [MJ] - energy content in fast electrons;
4. PASNUM - number of passages required for absorption of a certain power fraction given by parameter CCD1.

NBINJ - calculation of distribution function, current density CUFI and particle density NIBM, transversal PBPER and longitudinal PBLON pressure of suprathermal ions, plasma current density CUBM, generated by beam, electron source SNEBM and flux density of thermal neutrals SNNBM, momentum SCUBM and energy PIBM transferred to ions resp. (PEBM) to electrons of the plasma for injection of a beam of fast neutrals [25].

NEUT - calculates distribution NN and temperature TN of neutrals after the input parameters ABC, AMAIN, ZMAIN, NNCL, NNWM, ENCL, NITER, ENWM from list 1.1 and NE, TI, TE, ZEF from list 1.2

SETBND(W1,W2,CON,M,LDV,KSI,GNN) - calculates boundary conditions in a null-dimensional model [26].

Input parameters:

1. Fluxes of particles QNB and heat QIB+QEB across the separatrix,
2. W1 and W2 [keV] - ionization cost in SOL and near the plates;
3. CON, M, LDV, KSI, CBND1- parameters for describing the divertor geometry.
4. CBND2-4 - fitting parameters.

Output parameters:

NEA, TEA, TIA, WHALF, WIDTH, WPFL, WQRAD, WTEM (see list 1.1), GNN - neutral flux into a plasma from the recycling zone.

SETNE(NEGIVN,NESLOP) - this subroutine provides output of the volume-averaged density NEAVA (list 1) at a given value of NEGIVN [$10^{19}/m^3$] with the velocity NESLOP [$10^{19}/m^3/s$].

SNEAV(GIVNNE,DT,DELT,ONmin) - this subroutine provides output of the volume-averaged density NEAVA (list 1) at a given value of GIVNNE [$10^{19}/m^3$] for a time DT [s] with incremental growth DELT [ ] where the neutral-particle flux is bounded from below: QN>QNmin.

APPENDIX 2. CONTROL OF THE TIME STEP

The precision of calculation and the stability of the solution depends on the time step. As a rule, in a realistic calculation an instability is due to non-differential terms in the equations (sources and sinks) or with equilibrium calculations, especially when rapid jumps occur in the plasma parameters. By shortening the step, this instability may be suppressed. In transport codes created by the ASTRA system, an automatic selection of the time step TAU is employed, which is determined according to the algorithm

$$TAU = \frac{\hat{TAU}}{\max\left\{ \frac{1}{TAUinc}, \left|\frac{\hat{u}}{u} - 1\right| \frac{1}{DELvar} \right\}}$$

where u means the values of the variables NE, TE, and TI in each lattice point along the radius. The "^" sign is used to mark the corresponding values from the preceding time interval. Thus, the variable TAUinc gives the maximal increment of the step during one step, and DELvar is the admissible relative change in the variables. Beyond re-calculating variables after the formula cited above, TAU satisfies the following direct bounds:

$$TAUmin < TAU < \min\left\{ TAUmax, \frac{dTout}{2}, \frac{dPout}{2} \right\},$$

i.e. TAUmin and TAUmax give, in explicit form, the minimal and maximal step values.

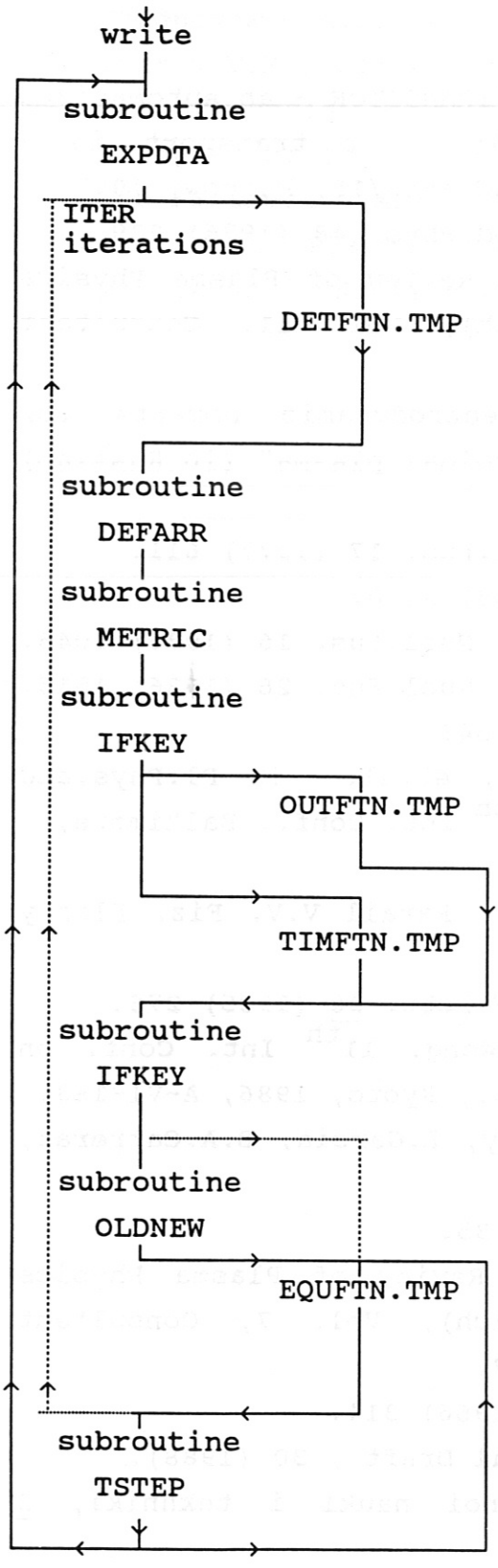
Beside the step control, the stability of the scheme and the order of approximation may be improved by introducing iterations. Usually, however, iterations are not needed, setting the standard value at ITER=1.

Distribution of the files of the ASTRA system over the directories

DIRECTORY	FILES	CONTENTS
Root	*.lib	subroutine libraries
	*.bat	startup files for the ASTRA system
DAT	*.*	calculation results
EQU	*	initial texts of the models
	*.txt	results of model-equation analysis
	*.cns	control constants for the model
	*.var	control constants for the model
FML	*	library of formulas for the ASTRA system
FNC	*.for	library of functions for the ASTRA system
SBR	*.for	library of external subroutines
FOR	*.for	library of internal subroutines
	*.inc	text blocks for including in other parts of the ASTRA system
TMP	*ftn.tmp	temporary FORTRAN texts
	*.dir	lists of all files in the subdirectories
	map.map	"map" file of the last created model
TSK	*.exe	executable transport codes
EXP	*	library of experimental discharges and initial conditions
EXE	*.bat	command files for the ASTRA system
	*.exe	service programs for the ASTRA system
SRV	*.for	texts of service subroutines

Structure of transport codes created by the ASTRA system

Permanent part	Temporary texts	Operations executed
*.INC		Description of variables and setting of standard initial values
write		Message: 'Initial iteration start'
subroutine INICON		Input of model constants Inclusion of parameters from experimental variant
	DATFTN.TMP	Assignments of the model constants
subroutine IFKEY		Assignment of names and scale factors for output variables Verification of the key control by calculation and output
subroutine METRIC		Equilibrium calculation Calculation of the metric coeff.
5 iterations		
	DETFTN.TMP	Assignment of the constants and of the radial profiles taken from the model
	INIFTN.TMP	Assignment of initial values
subroutine DEFARR		Calculation of the variables always determined in the model
subroutine INIVAR		Calculation of magnetic field, current and voltage
subroutine METRIC		Equilibrium calculation Calculation of the metric coef.
subroutine IFKEY		Verification of the key control by calculations and output



Message: 'Time evolution start'

Calculation of time unfolding of experimental parameters

Calculation of the boundary conditions and the variables taken from the model
Calculation of transport coefficients

Calculation of the variables always determined in the model

Equilibrium calculation
Calculation of the metric coeff.

Verification of the key control by calculations and output

Calculation of radial profiles for output

Calculation of variables for output of time functions

Drawing and printing of graphs
Printing of results
Control of code operation mode

Re-calculation of values on the preceding time interval

Solution of equations and calls for external subroutines in accordance with the sequence of description in the model

Time step evaluation. In case of radical changes: return to EQUFTN.
For normal steps, t=t+dt

——— Loop for all iteration steps except the last.

References.

1. Pereverzev G.V., Yushmanov P.N. "TRANSITOR - an automatized system for simulation and analysis of transport in a tokamak" (in Russian) Preprint IAE-4545/15, Moscow, 1987.
2. Hinton F.L.,Haseltine R.D. Rev.Mod.Phys. 48 (1976) 239.
3. Zakharov L.E. Shafranov V.D. in Review of Plasma Physics (Edited by Acad. M.A.Leontovich), Vol. 11, Consultant Bureau. New York - London, p.153.
4. Zakharov L.E. "Method of electrodynamic moments for equilibrium calculations in a toroidal plasma" (in Russian) Preprint IAE-4114/6, Moscow 1985.
5. Hirshman S.P., Hawryluk R.J. Nucl.Fus. 17 (1977) 611.
6. Hirshman S.P. Phys.Fluids 31 (1988) 3150.
7. Jassby D.L.,Cohn D.R.,Parker R.R. Nucl.Fus. 16 (1976) 1045.
8. Romanelli F.,Tang W.M.,White R.B. Nucl.Fus. 26 (1986) 1515.
9. Liewer P.C. Nucl. Fus. 25 (1985) 543.
10. Merezhkin V.G.,Mukhovatov V.S., et.al. in Pl.Phys.and Contr. Nucl. Fus. Res. (Proc. 9th Int. Conf., Baltimore, 1982) 1, IAEA, Vienna (1983) 83.
11. Dnestrovskij A.Yu., Ivanov V.V., Parail V.V. Fiz. Plazmy 14 (1988) 902.
12. Parail V.V. Yushmanov P.N. ZhETF Lett. 30 (1985) 276.
13. R.E.Waltz, R.R.Domingues, S.K.Wong. 11th Int. Conf. on Plasma Phys. and Contr. Fus. Res., Kyoto, 1986, A-VI-1-3.
14. T.S.Hahm, P.H.Diamond, P.V.Terry, L.Garcia, B.A.Carreras, Phys. Fluids 30 (1987) 1452.
15. T.Ohkawa, Phys.Lett. 67A (1978) 35.
16. Galeev A.A., Sagdeev R.Z. in Review of Plasma Physics (Edited by Acad. M.A.Leontovich), Vol. 7, Consultant Bureau. New York - London, p.257.
17. Shafranov V.D. Plasma Phys. 8 (1966) 314.
18. Uckan N.A. ITER-TN-PH-8-6 Final Draft , 30 (1988).
19. Putvinskii S.V. Voprosy atomnoi nauki i tekhniki, 2 (1988), 3.
20. D.R.Mikkelsen, C.E.Singer, Nucl. Tech./Fusion 4 (1983) 237.
21. Post D.E. Jensen R.V. e.a. PPPL-1352 1977.
22. Freeman R.L.,Jones E.M., CLM-R137 (1974).
23. Golovin I.N. Preprint IAE-4885/8 (1989).

24. Chudnovskii A.N. Preprint IAE-3887/6, (1984).
25. Leonov V.M., Polevoi A.R. Preprint IAE-5013/8,(1990).
26. Dnestrovskij A.Yu.,Krasheninnikov S.I.,Yushmanov P.N. Analysis of Zero-Dimensional Dependencies for the Tokamak SOL-Parameters and Their Effect on Transport Processes in the Main Plasma Preprint IAE-5142/6 (1990); to be published in Nuclear Fusion 1991.