21-77

3 cp. to NTIS

ORNL/TM-5262

A User's Manual for the Oak Ridge Tokamak Transport Code

J. K. Munro
J. T. Hogan
H. C. Howe
D. E. Arnurius

OAK RIDGE NATIONAL LABORATORY

OPERATED BY UNION CARBIDE CORPORATION FOR THE ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
Price: Printed Copy $10.00; Microfiche $3.00

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the Energy Research and Development Administration/United States Nuclear Regulatory Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

ORNL/TM-5262

Contract No. W-7405-eng-26

FUSION ENERGY DIVISION

A USER'S MANUAL FOR THE OAK RIDGE TOKAMAK TRANSPORT CODE

J. K. Munro
J. T. Hogan
H. C. Howe
D. E. Arnurius

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

NOTICE

This document contains information of a preliminary nature. It is subject to revision or correction and therefore does not represent a final report.

Date Published: February 1977

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

CONTENTS

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

FOREWORD

This document is intended as a starting point for a continuing documentation of the Oak Ridge Tokamak Transport Code. The code has been developed by scientists whose chief interest lies in the analysis of tokamak experiments and the formulation of reliable models of extrapolation. Consequently, potential users are cautioned that the version of the code described here is far from optimum in the computational sense. Some present versions of the code, incorporating most of the physical models described in this manual, have halved both execution time and storage requirements. Nonetheless, this manual contains such a detailed description that users who require greater efficiency have the information needed to obtain it.

As this manual is written, the Oak Ridge Tokamak Transport Code includes some features not described here. These include:

a) a Fokker-Planck solution to the fast injected ion distribution function at $N_i \sim 5$-10 radial points (that is, N_i solutions of the two-dimensional time-dependent equations for f_i (x_i, v, ξ, t).

b) a multi-group, discrete ordinate (S_N) cylindrical geometry calculation of neutral gas transport. This NUTRLSN package has been developed by modifying the XSDRN neutron and photon transport solver which has been extensively used in other fields.

c) a quasi-two-dimensional magnetic geometry. In this scheme the (r,t) radial diffusion problem is replaced by one solved on a (ψ,t) grid where ψ is the poloidal flux function. The ISLAND code of D. C. Stevens (as used in the work of Grad, Hu, and Stevens) is then employed to compute the two-dimensional axisymmetric magnetic surface structure.

These additional modules are run with the same basic structure described in this manual and the logic has been designed to accept them. It is expected that documentation of these more sophisticated features will soon be forthcoming.

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

ABSTRACT

A one-dimensional tokamak transport code is described which simulates a plasma discharge using a fluid model which includes power balances for electrons and ions, conservation of mass, and Maxwell's equations. The modular structure of the code allows a user to add models of various physical processes which can modify the discharge behavior. Such physical processes treated in the version of the code described here include effects of plasma transport, neutral gas transport, impurity diffusion, and neutral beam injection. Each process can be modeled by a parameterized analytic formula or at least one detailed numerical calculation. The program logic of each module is presented, followed by detailed descriptions of each subroutine used by the module. The physics underlying the models is only briefly summarized. The transport code was written in IBM FØRTRAN-IV and implemented on IBM 360/370 series computers at the Oak Ridge National Laboratory and on the CDC 7600 computers of the Magnetic Fusion Energy (MFE) Computing Center of the Lawrence Livermore Laboratory. A listing of the current reference version is provided on accompanying microfiche.

A USER'S MANUAL FOR THE OAK RIDGE TOKAMAK TRANSPORT CODE*

1. INTRODUCTION

1.1 GENERAL STRUCTURE OF PROGRAM SPATIAL-TIME GRID

This manual is intended for a user who wishes to change the basic physical model in the code for one or more processes, or to incorporate a model for some process not already treated in order to study its interaction with the basic plasma transport process.

The manual can be most effectively used with two other documents:[1,2]

J. T. Hogan, "Multi-Fluid Tokamak Transport Models," ORNL/TM-5153 (November 1975); *Methods of Computational Physics*, Vol. 16: Computer Applications to Controlled Thermonuclear Research, ed. by J. Killeen, B. Alder, and S. Fernbach (1976).

J. T. Hogan, *Physical Models in the Oak Ridge Tokamak Transport Code*, ORNL/TM, to be published.

The reference version of the Oak Ridge Tokamak Transport Code (ØRTTC) is supplied in Appendix IV on microfiche.

The code is available to members of the controlled fusion community via the Magnetic Fusion Energy (MFE) Computer Center at the Lawrence Livermore Laboratory.

The tokamak transport code was developed as a tool to aid both theorists and experimentalists working with tokamak-type toroidal plasma machines. This code has been structured to allow modular incorporation of new models of various plasma processes as they are developed. The code has been extensively documented internally to aid users who must know the details of the processes being modeled. For those who wish to use the code to help interpret data, the input data is arranged so as to allow for the inclusion of extensive comments in the actual input data without disturbing the normal running of the program.

*Research sponsored by the Energy Research and Development Administration under contract with Union Carbide Corporation.

The input data is grouped as much as possible to reflect the modular structure of the program. This parallel organization should facilitate the selection of options for a particular type of simulation calculation. Where possible, an attempt has been made to include at least two alternatives for the important processes modeled: an analytic model that provides a rough approximation but gives a quick calculation, and a more detailed numerical computation which treats the physics of the process with more care.

The program can be run with a set of default input values. Since there are many combinations of options available, a default combination has been chosen which will be designated the standard choice. The "standard choice" data set is also a useful reference point if the user attempts to use a combination of input values for which the code produces numerically unstable profiles. A poor choice of input data values may result in difficulty in getting a simulation started; i.e., problems may develop in the first few time steps and cause the profiles to behave strangely or to grow rapidly to large magnitudes. The basic spatial and time intervals for the standard choice data set are 0.02 × (plasma radius, cm) and tens of microseconds, respectively. Some useful guidelines to consider in putting together a data set that should not lead to numerical difficulty are summarized in Sect. 1.4.8. Models currently under development are not treated in this manual, but a logical structure which will accommodate them has been chosen for the code. Some examples of models not covered here are:

- computation of solutions to a two-dimensional Fokker-Planck equation for the neutral beam injection module
- use of the NUTRLSN neutron transport code for the neutral gas module
- two-dimensional flux surface calculations[3]
- the collisional plasma model[4] to replace the plasma diffusion equations.

1.2 USE OF SWITCHES AND DEFAULT VALUES

The code uses a number of switches for a variety of purposes. These switches can be grouped according to their purpose and/or use into the following types:

Type	Purpose and/or Use
A	Determines choice of particular model of a process
B	Controls input and/or output
C	Determines choice of machine configuration, e.g., neutral beam injection
D	Provides choice of diagnostic options
E	Provides values of exponents for power law expressions.

A detailed description of the various switches is given in Table 1.

Table 1. Summary of switch types and the function or use of each switch

Type A Switches: Choice of analytic or detailed numerical model

NUTE(1) Neutral gas calculation
= 1, algebraic formula
= 2, Boltzmann transport model with slab geometry, no wall reflection
= 3, same as NUTE(1) = 2, but with wall reflection
= 4, NUTRLSN transport code
Options 3 and 4 are not included in the Reference Version (see Appendix IV)

IMP(1) Initial impurity stripping calculation
= 1, corona equilibrium table for carbon and oxygen
= 2, rate equation solutions for carbon and oxygen

IMP(2) Radial distribution of total impurities
= 1, corona equilibrium tables together with radial distribution formulae for carbon and oxygen. High-Z impurities are treated using a crude estimate.
= 2, corona equilibrium tables for carbon, oxygen, and high-Z impurity (Fe) atom charge states. Radial distributions are assigned using formulae.
= 3, corona equilibrium tables together with radial distributions generated by Pfirsch-Schlüter diffusion for carbon and oxygen. High-Z impurities are treated using a crude estimate.
= 4, corona equilibrium tables for carbon, oxygen, and high-Z impurity (Fe) atom charge states. Radial distributions are generated by a Pfirsch-Schlüter diffusion model.

Table 1 (continued)

IMP(3) Impurities: Fixed values (=1) or detailed calculation (=2) according to the value of IMP(2).

IMP(4) Detailed model for iron
= 0, include iron coronal equilibria
= 1, omit detailed description

INJ(3) Current perturbation from injected neutral beam

INJ(4) Electric field effect on fast ions

INJ(5) Loss cone correction to fast ion thermalization

INJ(6) Neutral injection gas
= 0, hydrogen
= 1, deuterium

Type B Switches: I/Ø control

INIT(1) Generate initial plasma profiles analytically or read them in/restart calculation at some arbitrary time.
= 0, generate initial profiles analytically ("normal" start)
= 1, read in initial profiles
= 2, restart calculation at a designated time

Type C Switches: Machine configuration

INJ(1) Neutral beam injection
= 0, no injection
= 1, moments calculation
= 2, debug calculation

INJ(2) Number and type of neutral beam injection used
= 1, co-injection only
= 2, counter-injection only
= 3, two injectors, one co- and one counter-
= 4, four injectors, two co- and two counter-

Type D Switches: Diagnostic options

INIT(4) Diagnostic options switch
= 0, off
= 1, on

Type E Switches: Exponents and power laws used

NUTE(2) Exponent in power law for neutral gas analytic model.

A standard set of default input values has been incorporated into the code; the entire set is given in Appendix I. The choice of default value for each input variable is explained in the detailed description of the input data (Sect. 1.4). The default values for the type A switches will generally pick the analytic model. Type B switches will be set to generate the initial plasma profiles analytically and to provide detailed printout.

1.3 BRIEF DESCRIPTION OF MODULAR STRUCTURE

Each of the seven basic modules currently in the code is discussed in a separate section. Each section includes a description of the input data required and a summary of the function of each subroutine used by the module. These modules and their functions are briefly as follows:

MAIN — reads the input data and provides primary program control (Sect. 2.5)

PLASMA — obtains the solution for the set of partial differential equations describing the time evolution of the plasma (Sect. 3)

NEUTRA — calculates the effects of the inward-diffusing cold neutral gas (Sect. 4)

IMPRTY — calculates the effects due to the presence of the impurities carbon and oxygen and a high-Z element (Fe) (Sect. 5)

BEAM — models the neutral beam injection process (Sect. 6)

DIAGNØ — generates diagnostic tests to measure the accuracy and performance of the simulation (Sect. 7)

SURFAC — models the interaction between the plasma surface and the external world (diverters, torus walls, etc.)(Sect. 8).

Labeled common blocks have been used extensively to store variables and arrays whose values are used or generated in more than one module or subroutine. The common blocks were constructed to parallel the modular structure of the transport code as much as possible. Common block names were chosen to suggest a common physical use or association of the variables and arrays grouped in each block. A list of the variables (arrays) in each common block is given in Appendix II, together with variable and array definitions.

1.4 DESCRIPTION OF INPUT DATA AND DEFAULT VALUES

An input data set, completely internally documented, is presented in Appendix I to enable the user to run the plasma simulation code without having to consult a series of memos. Changes to the input data can be made by editing the appropriate NAMELIST list in this input data set. The standard choice, with complete documentation, is built into BLØCK DATA, so that it will always be available in the code for reference. All input data appears in the labeled common blocks.

In the sample input data set, variables appropriate for each NAMELIST list grouping are described immediately preceding the NAMELIST data set. Units of the input variables are given with their definitions. Default values for the input data are given in parentheses following the definitions of the variables; these values are the ones appropriate for the ORMAK machine.

1. Machine Configuration. The parameters associated with the tokamak geometry (ORMAK values are the default) are:

AM	— minor radius of torus, cm	(23.0 cm)
RØ	— major radius of torus, cm	(79.8 cm)
BT	— toroidal magnetic field, G	(18 kG)
R9[*]	— initial plasma radius, cm	(23.0 cm)
N	— number of spatial grid (radial) points	(51)

2. Timing Intervals. Two groups of timing intervals appear in the code. The first group listed below deals with the basic plasma model. The second group is associated with the neutral beam injection module and appears with the input for that module in Sect. 7.

TMAX	— maximum duration of current pulse, msec	(65.0 msec)
TC	— rise time of current, msec	(10.0 msec)
TPØ[†]	— fall off time of external neutral density, msec	(4.0 msec)
TSTAR	— starting time of simulation problem, msec	(0.0 msec)
NT	— total number of time intervals for current pulse and plasma simulation	(1000)

[*] See subroutine GETSET.

[†] See subroutine SURFAC.

3. Plasma Data. This group of input variables is used primarily by the plasma module for generating the initial plasma variable profiles.

TEØ — boundary temperature for electrons, eV (10.0 eV)

TIØ — boundary temperature of ions, eV (10.0 eV)

TEB — initial central value of electron temperature, eV (20.0 eV)

TIB — initial central value of ion temperature, eV (20.0 eV)

VØLT — toroidal voltage, V (3.0 V)

DENØ — initial central value of plasma electron density, cm^{-3} (6×10^{12} cm^{-3})

DENB — boundary electron density, cm^{-3} (5×10^{12} cm^{-3})

ZPC — initial value of total current, amps (15.0 kA)

ZFI — additional amount of total current after time TC, amps* (85.0 kA)

AMU — mass of plasma ions, amu, determined according to the equation:

$$\mathrm{AMU} = (2{*}N_{D+} + N_{H+})/(N_{D+} + N_{H+}),$$

where N_{D+} and N_{H+} are the respective densities of deuterons and protons (1.0 amu)

XI(1) — exponent for initial profile of current density, (2.0)

$$j = j_o[1 - (R/R9)^{XI(1)}]$$

XI(2) — exponent for initial profile of electron temperature, (2.0)

$$T_e = (TEØ-TEB)[1-(R/R9)^{XI(2)}] + TEB$$

XI(3) — exponent for initial profile of ion temperature, (2.0)

$$T_i = (TEØ-TIB)[1-(R/R9)^{XI(3)}] + TIB$$

XI(4) — exponent for initial profile of electron density, (2.0)

$$N = (DENØ-DENB)[1-(R/R9)^{XI(4)}] = DENB$$

INIT(1) — switch for starting mode for initial plasma variable profiles (0)
= 0, for profiles generated analytically
= 1, for profiles read from data cards

*See function PC.

INIT(3) — switch for examining MHD stability characteristics of the plasma (1)
= 0, determines the location of the singular surface for m = 3,5
= 1, calculates the radial eigenfunctions and tearing mode growth rate according to Furth, Rutherford, and Selberg,[5] for m = 3,5. Can be used to evaluate transport coefficients in subroutine CDC.

D11(1) — electron thermal transport coefficient, for scaling diffusion coefficient = D11(1) *pseudoclassical (3.5)

D11(2) — ion thermal transport coefficient, for scaling diffusion coefficient = D11(2) *neoclassical (0.3)

D11(3) — particle diffusion coefficient = D11(3) *pseudoclassical (1.0)

D11(4) — resistivity = D11(4) *neoclassical (1.0)

D11(5) — threshold (nue*) of trapped electron scaling (0.0)

D11(6) — threshold (nui*) of trapped ion scaling (0.0)

4. <u>Neutral Gas Data</u>. This group of input variables is used almost entirely by the neutral gas module for picking a path through a consistent choice of options associated with a particular neutral gas model.

FCDEN — Initial Franck-Condon boundary density, cm^{-3} (18×10^9 cm^{-3})

FCF — final boundary density of neutrals, cm^{-3} (9×10^9 cm^{-3})

NØX — number of nodes x_j in [0,1] for use with the discrete integral equation associated with the integral equation that has the hot neutral density as its solution. Approximations to the hot neutral density are obtained at the x_j. Approximations at nonnodal points are obtained by linear interpolation. In regions where the slope of the hot neutral density changes rapidly, more nodal points should be added. (21)

NØZ — number of uniformly distributed points in the interval [-1,1]. For each node x_j, the integration interval [-1,1] is decomposed into the subintervals: $[-1,m^-(QC)]$, $[m^-(QC),n^-(QF)]$, $[n^-(QF), x_j]$, $x_j,n^+(AF)]$, $[N^+(QF),m^+(QC)]$, and $[m^+(QC),1]$. The points determined by NØZ that lie in $[-1,m^-(QC)]$ and $[m^+(QC),1]$ are used as integration nodes. (101)

NSIMP — number of uniformly distributed integration nodes in a Simpson's approximation to the integral on $[m^-(QC),n^-(QF)]$ and on $[n^+(QF), m^+(QC)]$.

NSMO — number of uniformly distributed nodes in $[n^-(QF),x_j]$ or in $[x_j,n^+(QF)]$. Used as nodes for a Simpson's approximation in one case and as a partition of the integration interval in another. (5)

NUNIF — switch to select mode for describing the nodal points x_j. See NØX above. (0)

= 0, x_j defined in subroutine NEUTIO to be (J-1)/(NØX-1) j = J = 1, ..., NØX.

> 0, desired x_j should be placed at the beginning of array W1 in common NUTRLP. Require W1(1) = 0, and W1(NØX) = 1.0.

QF,QC — used to partition integration interval. See NØZ above and Sect. 4.4.3 in the neutral module. (.001,.11)

EPSN1 — the functional iteration used to approximate the hot neutral density at the nodes x_j is terminated whenever the magnitude of the absolute error between the solution of the discrete integral equation and the current approximation is less than EPSN1 at each x_j. (.001)

INTRPS — the values of the functions N_i, T_i, and EATN at nonnodal points are approximated by interpolation.

= 1, for linear interpolation

= 2, for quadratic interpolation

= 3, for cubic interpolation (2)

IDEBUG — a switch controlling printout from subroutine NUKERN (1)

$\leqslant 0$, no additional output
= 1, A,B,C, γ = Lipschitz constant, estimated number of functional iterations, and max $\{q(z,x)\}$
= 2, all of the above, plus the arrays NI, TI, and EATN
= 3, all of the above, plus approximations to hot neutral density at nodes x_j

EØ — energy of neutrals leaving the wall, eV (10.0)

EPSNO — a new basic neutral density profile is requested if and only if the magnitude of the relative error between

A*NI(J)/SQRT(TI(J))

for the last calculated profile and for the corresponding current values, exceeds EPSNO for some node x_j. NI and TI are the normalized ion density and temperature, respectively. (0.5)

IPNO — an output switch for subroutine NTF (1)

= 0, no extra output
= 1, prints range of relative error array described under EPSNO and prints the integration time
= 2, prints the above plus the arrays containing the new and old values for

A*NI(J)/SQRT[TI(J)]

and for the basic neutral density profiles

NWDW1 — number of elements in the array W1 in common NUTRLP. A minimum of 5N + 4NØX + 3NØZ elements in W1 is required whenever NUTE(1) = 2 (1300)

NWDWSC — number of elements in the array A in common LBMIBT. This array will be used as the scratch array WSC in the argument list to subroutine NUKERN. (8008) A minimum of

$$N\emptyset X(N\emptyset Z + 10NSMO + 5) + NSIMP\ (8N\emptyset X + 1)$$
$$+ 4(N\emptyset Z + NSMO) - 2$$

elements in A is required whenever NUTE(1) = 2

NUTE(1) — switch for model of neutral gas calculation (2)

= 1, for analytic model

= 2, for slab model without wall reflections

= 3, for slab model with wall reflections (not included in this description)

= 4, NUTRLSN transport code (not included in this description)

NOTE: The analytic model option will always be available. The remaining three options cannot be simultaneously linked with the rest of the program. This restriction exists to reduce core memory requirements. One of options 2-4 must be selected and the subroutines for that option linked with the rest of the program. This linkage is accomplished on IBM 360/370 series computers by inserting an INCLUDE card with a reference to the name of the cataloged data set which contains the subroutines for the desired option. Option 2 will be assumed to be the only one available in any subsequent discussions in this user's manual about the neutral gas module.

NUTE(2) — exponent in power law for neutral gas analytic model (4)

5. Impurity Ion Data. This input data is used to some degree throughout the code, but is most closely associated with the impurity module (Sect. 5).

PCIMP — percentage of electron density due to high-Z impurity. This parameter has two different meanings, according to the impurity diffusion model option determined by the value of IMP(2). For IMP(2) = 1 or 3, PCIMP is due to some average effective high-Z impurity; for IMP(2) = 2 or 4, it is due to iron impurities (0.01)

ZEFF — effective charge of ions (4.0)

ZIMP — maximum charge of the high-Z impurity ion (25)

XCAR — percentage of electron density due to carbon impurities (2)

XØX — percentage of electron density due to oxygen impurities (2)

IMP(1) — switch for initial impurity stripping (1)
= 1, corona equilibrium for carbon and oxygen
= 2, dynamic rate calculation for carbon and oxygen

IMP(2) — switch for diffusion of impurities (1)
= 1, corona equilibrium tables with radial distribution formulae for carbon and oxygen. High-Z impurities are treated using a crude estimate
= 2, corona equilibrium tables for carbon, oxygen, and high-Z impurity (Fe) atom charge states. Radial distributions are assigned using formulae
= 3, corona equilibrium tables with radial distributions generated by Pfirsch-Schlüter diffusion for carbon and oxygen. High-Z impurities are treated using a crude estimate
= 4, corona equilibrium tables for carbon, oxygen, and high-Z impurity (Fe) atom charge states. Radial distributions are generated by a Pfirsch-Schlüter diffusion model

IMP(3) — switch for impurity model (1)
= 1, use fixed values
= 2, use calculation determined by choice of switch value for IMP(2)

IMP(4) — switch for detailed treatment of iron (1)
= 0, include iron coronal equilibria
= 1, omit detailed description

6. Neutral Beam Injection Data. This group of input variables is used almost entirely by the neutral beam injection module for choosing a path through a consistent set of options associated with a particular neutral beam injection model.

DTCH — time increment for calculation of profile H(r), msec (10.0 msec)

DTCGL — time increment for calculation of GE, GI, and KE, msec (2.0 msec)

RC — major radius of beam line point of tangency, cm (75.0 cm)

RB — neutral beam radius, cm (15.0 cm)

BHLF — neutral beam half-width, assuming Gaussian profile, cm (7.5 cm)

TNJCT — starting time for injection, msec (20.0 msec)

DTNJCT — neutral beam injection current rise time, msec (6.0 msec)

N1 — number of computational time steps in plasma current rise time interval (500)

N3 — number of computational time steps in injected neutral beam current rise time interval (200)

INJ(1) — neutral beam injection calculation switch (1)
= 0, no injection
= 1, moments calculation
= 2, debug (quick-and-dirty) calculation

INJ(2) — switch for number and type of neutral beam injectors used (3)
= 1, co-injection only
= 2, counter-injection only
= 3, two injectors only, one co- and one counter-
= 4, four injectors, two co- and two counter-

INJ(6) — injected gas (0)
= 0, hydrogen
= 1, deuterium

For the following switches, 0 = off and 1 = on:

INJ(3) — switch for current perturbation from fast ions (1)

INJ(4) — switch for electric field effects on fast ions (0)

INJ(5) — switch for loss cone correction to fast ion thermalization (0)

7. I/Ø Data and Control Switches. This group of variables and switches governs how often printout is obtained during the simulation run and what level of diagnostic information is derived from the diagnostics module.

NR — number of printout sets obtained during execution of the code (5)

8. Using Other Data Sets. Data sets with the same numerical structure (i.e., same N, NT, N1, etc.) will have a better chance of success if the ohmic heating input rate chosen initially is close to the one for the reference value. Thus, ηj^2 should be close to the value given here; variations should incorporate the fact that $\eta \sim T_e^{-3/2}$ and $j \sim$ total current/area.

1.5 COMMENTS ON USE OF NAMELIST FEATURE FOR INPUT DATA

The format for the data in the NAMELIST data sets is as follows:

1. Column 1 must always remain blank.
2. A set of data associated with a particular NAMELIST list name must begin with an ampersand in column 2 followed immediately by the NAMELIST name. The set of data must end with a line containing an ampersand in column 2 followed immediately by END.
3. Values for single variables are entered, beginning in column 2, according to the format:

   VAR1 = value1, VAR2 = value2, etc.
4. Values for arrays may be entered in two ways:
   a. individually:

      ARRAY(1) = value1, ARRAY(3) = value3, etc.

   b. in a list:

      ARRAY = value1, value2, value3, etc.

After the input data has been read in, it can be written back out using the NAMELIST feature. If a new variable (array) is added to the input data, the appropriate NAMELIST statement defining the variables (arrays) associated with the NAMELIST name must have the variable (array) name added to it. These statements all occur in the group of nonexcutable statements at the beginning of subroutine GETSET.

1.6 PROGRAMMING CONSIDERATIONS

The language used is FØRTRAN IV. The code was developed using the IBM 360 series computers and the ØPT = 2 version of the FØRTRAN-H compiler. While the ORNL IBM ØS/360 FØRTRAN-H compiler can accept statements such as:

XK = (I - 1)*1./NM1/NM1 ,

and can handle the mixed mode arithmetic properly, other compilers do not have this capability. An attempt has been made to weed out this type of machine-dependent statements and to replace these statements with sequences of statements that should not cause difficulty on most compilers. We have tried to use ANSI standard FØRTRAN whenever potentially troublesome statements such as the one in the example above are found.

REFERENCES

1. J. T. Hogan, *Multi-Fluid Tokamak Transport Models*, ORNL/TM-5153 (November 1975); *Methods of Computational Physics*, Vol. 16: Computer Applications to Controlled Thermonuclear Research, ed. by J. Killeen, B. Alder, and S. Fernbach (1976).

2. J. T. Hogan, *Physical Models in the Oak Ridge Tokamak Transport Code*, ORNL/TM, to be published.

3. H. Grad, P. N. Hu, and D. C. Stevens, *Proc. Natl. Acad. Sci.* USA, 72: 3789 (1975).

4. H. K. Meier, W. I. van Rij, C. O. Beasley, Jr., and J. E. McCune, *The Collisional Plasma Model: A Velocity-Space Orthogonal-Function Representation for the Distribution Function of a Collisional Plasma*, ORNL/TM-5314 (March 1976); H. K. Meier, W. I. van Rij, and C. O. Beasley, Jr., *Techniques for Computing the Numerical Values of the Matrix Elements of the Collision Operator in the Collisional Plasma Model*, ORNL/TM-5315 (March 1976); W. I. van Rij, H. K. Meier, C. O. Beasley, Jr., and J. E. McCune, *Kinetic Equations for the Collisional Plasma Model*, ORNL/TM-5316 (March 1976); C. O. Beasley, Jr., J. E. McCune, H. K. Meier, and W. I. van Rij, *Numerical Study of Drift-Kinetic Evolution of Collisional Plasmas in Tori*, ORNL/TM-5317 (March 1976).

5. H. P. Furth, P. H. Rutherford, and H. Selberg, *Phys. Fluids* 16: 1054 (1973).

2. DESCRIPTION OF MAIN PROGRAM MODULE AND SUPPORTING SUBROUTINES

2.1. OVERVIEW OF CODE STRUCTURE

The module MAIN exercises primary program control, initializes all variables and arrays, computes plasma power balance at each time step of the simulation, generates a history of the plasma evolution, and contains most of the program output statements. A diagram of this module is shown in Fig. 1.

The subroutine call statements in the basic time loop of the main program have been limited as much as possible to the basic modules used for the simulation. Switches used to select a particular option have been left in each module rather than put in the main program. In some ways this obscures the logic in the program, since each module must be consulted in order to learn what logic and conventions are used. However, this program structure facilitates adding or changing modules. When module additions or changes are introduced, provision should be made to read in data and supervise variable and array initializations in GETSET, and to handle all output in subroutine PRTØUT (by use of an alternate entry point). Data transfer between GETSET, module, and PRTØUT can be made using a labeled common, unless a great deal of data is involved. Appendix II contains tables listing the variables and arrays with their definitions to aid the user who wants to add a new module or modify an existing one.

Almost all output is handled by subroutine PRTØUT, through the use of alternate entry points. This arrangement was chosen because of the economy of being able to use a few basic format statements to print the same sets of arrays at different stages of the simulation merely by inserting a call to the appropriate entry point name. This subroutine is never called as such.

2.2 INITIALIZATION PROCEDURE

All initializations of variables, arrays, and machine parameters are handled in subroutine GETSET, using the NAMELIST feature of FØRTRAN IV.

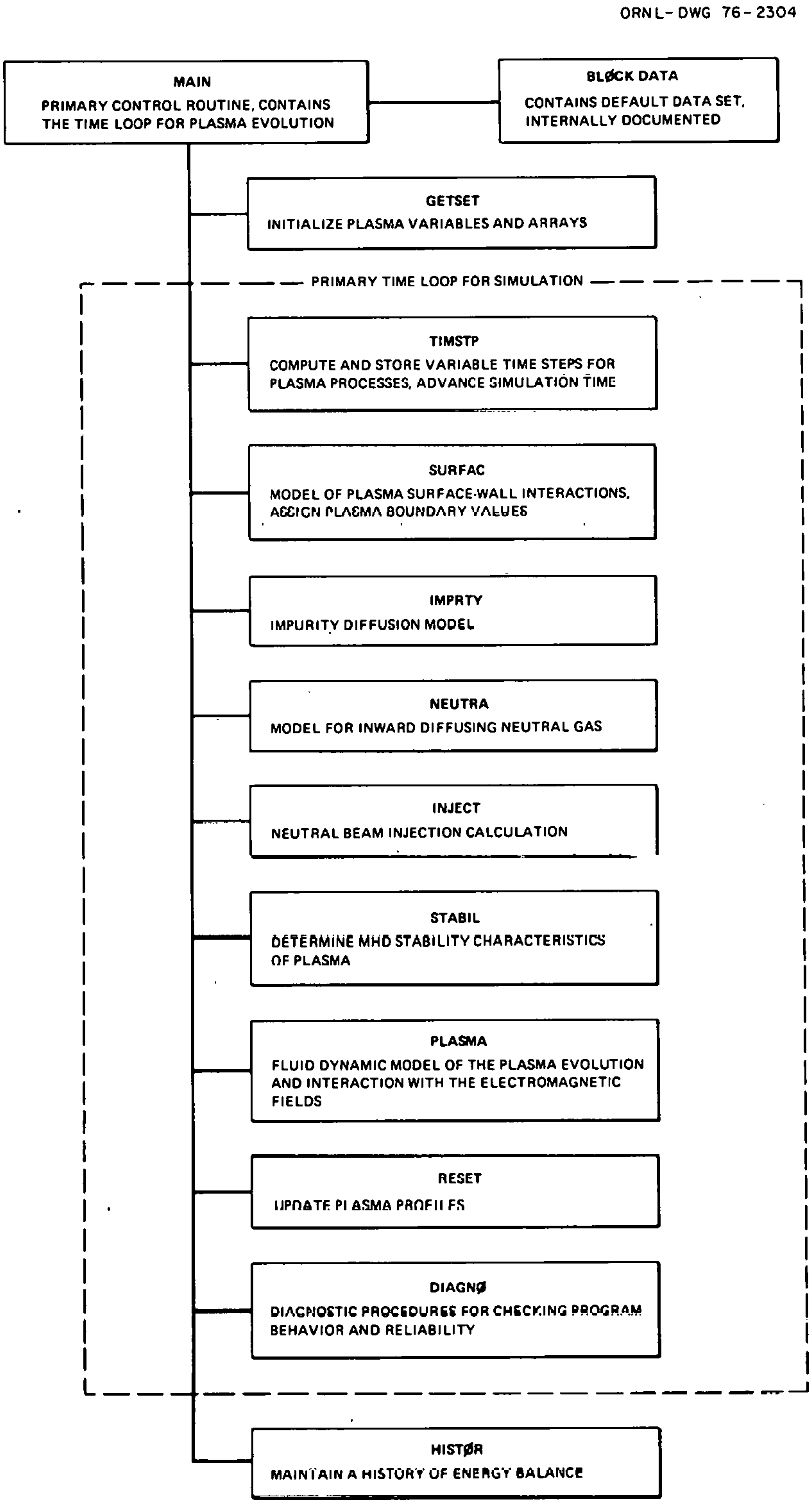

ORNL-DWG 76-2304
MAIN
PRIMARY CONTROL ROUTINE, CONTAINS THE TIME LOOP FOR PLASMA EVOLUTION
BLØCK DATA
CONTAINS DEFAULT DATA SET, INTERNALLY DOCUMENTED
GETSET
INITIALIZE PLASMA VARIABLES AND ARRAYS
PRIMARY TIME LOOP FOR SIMULATION
TIMSTP
COMPUTE AND STORE VARIABLE TIME STEPS FOR PLASMA PROCESSES, ADVANCE SIMULATION TIME
SURFAC
MODEL OF PLASMA SURFACE-WALL INTERACTIONS, ASSIGN PLASMA BOUNDARY VALUES
IMPRTY
IMPURITY DIFFUSION MODEL
NEUTRA
MODEL FOR INWARD DIFFUSING NEUTRAL GAS
INJECT
NEUTRAL BEAM INJECTION CALCULATION
STABIL
DETERMINE MHD STABILITY CHARACTERISTICS OF PLASMA
PLASMA
FLUID DYNAMIC MODEL OF THE PLASMA EVOLUTION AND INTERACTION WITH THE ELECTROMAGNETIC FIELDS
RESET
UPDATE PLASMA PROFILES
DIAGNØ
DIAGNOSTIC PROCEDURES FOR CHECKING PROGRAM BEHAVIOR AND RELIABILITY
HISTØR
MAINTAIN A HISTORY OF ENERGY BALANCE

Fig. 1. Diagram of MAIN module and subroutines called.

NAMELIST permits the user to include helpful documentation in the input data set without interfering in any way with the process of reading in the data set.

2.3 BALANCES AND HISTORIES

The plasma power and energy balances, comparisons between contributions of various terms in these balances, and the histories of selected plasma variables are calculated in subroutine ØRMPWR. Time histories are generated in the following arrays:

Array Name	Physical Process Contributing to Energy History
B1	ohmic heating
B2	thermal conduction and convection of electrons
B3	thermal conduction and convection of ions
B4	line radiation
B5	energy loss due to charge exchange
B6	stored energy of electrons
B7	stored energy of ions
B8	electron energy input from neutral beam injection
B9	ion energy input from neutral beam injection

Power spatial integrals and fluxes are generated in the following arrays for the time intervals for which printout is specified:

Array Name	Associated Physical Process
C1	ohmic heating, integral over space
C2	electron thermal conduction, flux
C3	electron convection, flux
C4	electron radiation loss; integral of line, recombination, and synchrotron radiation
C5	electron-ion transfer for electrons, integral
C6	ion thermal conduction, flux
C7	ion convection, flux
C8	energy loss due to charge exchange, integral
C9	electron energy due to neutral beam injection, integral

C10	ion energy due to neutral beam injection, integral
C11	ionization loss due to electrons, integral
C12	electron-ion transfer for ions (proton), integral

2.4 GRAPHICS OPTIONS

No graphics options are available in the plasma simulation code per se. The graphics programs which have been developed at ORNL assume that a simulation run has been made and that the output has been written into a particular file or data set. Two forms of graphic display have been used: the graphics package (ØRGRAPH)[1] developed for use at ORNL with the IBM 360 computers, and several interactive programs developed for use with the Tektronix 4000 series[2] storage tube terminal and the Tektronix Advanced Graphics[3] software package. The code's output format reflects the use of the Tektronix 4000 series terminal and its associated hard copy producer.

2.5 DESCRIPTIONS OF SUBROUTINES

Each subroutine and function will be described, wherever possible, according to the following format:

Purpose or use

Equations used

Called from:
Subroutines called:
Commons required:
NAMELIST names:
Variables and arrays (output) changed:
Significant internal variables:
Limitations on variables:
Limitations in subroutine applications:

ROUTINE MAIN

The routine MAIN only calls subroutines; it does no direct computation. It contains the main time loop for the simulation, and calls for the final printout of plasma histories at the end of the simulation run.

Subroutines called: GETSET, TIMSTP, NEUTRA, STABIL, SURFAC, PLASMA, RESET, DISGNØ, EMPRTY, INJECT, HISTØR

Commons required: INDEX

SUBRØUTINE GETSET

Subroutine GETSET reads in all the input data, initializes the variables and arrays used by each module in the simulation, and prints a report of the program input. The initializations of quantities for the various modules are made by calls to the executive subroutines for each module in the same order in which the data is read. This structure should make the initialization procedure more obvious. A set of default input data is passed from subprogram BLØCK DATA into this subroutine through labeled common blocks, so that the program can be run as soon as it is implemented on a particular computer. The documentation of the input data internal to BLØCK DATA was taken directly from an input data set in which the NAMELIST list of data appeared where the default values are set in DATA statements.

Two options exist for initializing the plasma variable profiles. The choice of option is made with the switch INIT(1). If INIT(1) = 0, the following analytic expressions are used (see ENTRY IPPC in executive subroutine PLASIM of the plasma module):

$$\text{(i)} \quad j_Z(r) = \begin{cases} \text{ZJØ} \left[1 - (\frac{r}{\text{R9}})^2\right]^{\text{XI(1)}} & , \ \text{R} < \text{R9} < \text{AM} \\ 0.001 & , \ \text{AM} \geqslant \text{R} \geqslant \text{R9} \end{cases}$$

$$\text{(ii)} \quad T_e(r) = \begin{cases} (\text{TEØ} - \text{TEB}) \left[1 - (\frac{r}{\text{R9}})^2\right]^{\text{XI(2)}} + \text{TEB} & , \ \text{R} \leqslant \text{R9} < \text{AM} \\ \text{TEB} & , \ \text{AM} \geqslant \text{R} \geqslant \text{R9} \end{cases}$$

$$\text{(iii)} \quad T_i(r) = \begin{cases} (\text{TIØ} - \text{TEB}) \left[1 - (\frac{r}{\text{R9}})^2\right]^{\text{XI(3)}} + \text{TIB} & , \ \text{R} \leqslant \text{R9} \leqslant \text{AM} \\ \text{TIB} & , \ \text{AM} \geqslant \text{R} \geqslant \text{R9} \end{cases}$$

(iv) $$N_e(r) = \begin{cases} (\text{DEN}\emptyset\text{-DENB}) \left[1 - (\frac{r}{R9})^2\right]^{XI(4)} + \text{DENB} & , \quad 0 \leqslant R \leqslant R9 \\ \text{DENB} & , \quad R9 \leqslant R \leqslant AM \end{cases}$$

(v) $$\begin{cases} B_p = 0 \\ B_p = ZJ\emptyset \; (\frac{0.2}{r}) \left\{1 - \left[1 - (\frac{r}{R9})^2\right] XI(1) + 1\right\}, \quad 0 \leqslant r \leqslant 9 \\ B_p = B_p \; (r - \Delta r) * (r - \Delta r)/r \quad , \quad r9 \leqslant r \leqslant AM \end{cases}$$

After an initial call to compute diffusion coefficients (subroutine CDC) the following equations are used:

(vi) $E(r) = \eta(r) \cdot j(r)$

(vii) $V(0) = 0$
$V(r > 0) = - TDP*(dN/dr)/N$

If INIT(1) > 0, initial profiles are read in from cards, according to the format (2F10.0, 2F10.3, 4PE10.2). The list of variables, for $1 \leqslant I \leqslant N$, is as follows:

TE(I) — electron temperature, eV
TI(I) — ion temperature, eV
ZJ(I) — plasma toroidal current density, a/cm^2
B(I) — poloidal magnetic field, G
E(I) — electric field (toroidal), V/cm
DEN(I) — plasma electron density, cm^{-3}
VEL(I) — plasma ion drift velocity, cm/sec
ZN1(I) — total neutral gas density, cm^{-3}

These variables are also the ones printed out in several of the profile summaries in subroutine PRTØUT. The variables can be punched out on cards to be read at this point for restarting or continuing a simulation run, if time limitations or other considerations require doing the simulation piecemeal. If the option of reading these profiles is used, the profile data must follow the data read by the NAMELIST feature.

The initial profiles required for variables associated with the neutral gas, impurity, and neutral beam injection modules will be generated by calls to the executive subroutines which maintain control over the computations done by each module. The arrays which must be initialized are:

1. Neutral gas module:

   TØ — neutral gas temperature profile

   ZN1 — total neutral gas density

2. Impurity module:

   DENP — ion density profile

   D1 — spatial profile of effective ionic charge; D1 > 1 with impurities

   HIZ — radiation from high-Z impurities

   QRD — radiation from low-Z impurities

   ZBR — Z bracket [Z], defined by:

$$[Z] \equiv \frac{\sum_{\text{species } k} \frac{m_{\text{proton}}}{m_k} n_k Z_k^2}{\sum_{\text{species } k} n_k Z_k}$$

   CARBØN — total density of carbon impurity, cm^{-3}

   ØXYGEN — total density of oxygen impurity, cm^{-3}

   CC(K,I) — density of (K-1)th charge state of carbon at I-th grid point

   CØ(K,I) — density of (K-1)th charge state of oxygen at I-th grid point

3. Neutral beam injection module:

   QE1 — injected neutral beam heat input to electrons

   QI1 — injected neutral beam heat input to ions

   DJT — fast ion current produced by injected neutral beam

Table 2 contains a list of the labeled common blocks used by GETSET, together with lists of the input and/or output variable and array names.

Called from:	MAIN
Subroutines called:	CDC, NEUTRA, IMPRTY, BMSTRT, INIØUT
Commons required:	BDYCØN, BEAMC, CDCLBX, CURENT, ELCTRN, FIELDS, GEØM, IMPURT, INDEX, IØNS, NEUTRL, NEWS, TEMP, TIME, TMINDX, XPØRT, ZRAD

Namelist names: MACHIN, TIMINT, PLADAT, NEGADA, IMPIDA, NBMIDA, IØDATA, TEMPDA

Input: See Section 1.5

Variables changed: See Table 2 and Appendix II.

Table 2. List of labeled CØMMØN blocks and elements in them used by subroutine GETSET

	Input		Output	
CØMMØN BLØCK	Variables	Arrays	Variables	Arrays
/BDYCØN/	DENB, DENØ, TEB, TEØ, TIØ			
/BEAMC/	DTCH, DTCGL, RC, RB, BHLF, TNJCT	CURBM, ENGY, INJ		DJT
/CURENT/	ZFI, ZPC			
/ELCTRN/	FQE			DEN
/FIELDS/	BT		ZJØ	B, E, ZJ
/GEØM/	AM, RØ, R9		AREA, HR, VØL	
/IMPURT/	PCIMP, XØX, XCAR, ZEFF, ZIMP			CARBØN, ØXYGEN, CC, CØ, DENP, ZBR
/INDEX/	N, NT, NR	NUTE, IMP, INIT	TIC, NM1	
/IØNS/	AMU			DEN
/NEUTRL/	ATT, FCDEN, FCF, PINJ			QE1 QI1, ZN1
/NEWS/			NEWS	
/TEMP/				TE, TI, TØ, VEL
/TIME/	TC, TSTAR, TMAX, TPØ		HT, TIM, TPR	
/TMINDX/	DTNJCT, N1, N3			
/XPØRT/		D11		D1
/ZRAD/				HIZ, QRD

SUBRØUTINE ØRMPWR

Subroutine ØRMPWR computes the plasma gross power balance, particle and energy confinement times, fluxes through surfaces, plasma volume sources and sinks, and a series of terms contributing to the gross power balance. This subroutine is the primary interface between the plasma simulation code and the user. Table 3 summarizes the basic physical processes considered and gives the names of the variables and arrays associated with them.

The following types of quantities are calculated:

1. Amount of power associated with the dominant plasma processes
2. Total amount of energy involved with each dominant process
3. Fluxes through the plasma surface
4. Confinement times
5. Safety factor
6. Number of microwave interferometer fringe shifts resulting from changes in the plasma density
7. Ratios of various plasma power loss terms to the power put into the plasma.

Section 2.3 gives the definitions of the B and C series arrays which are generated in ØRMPWR. The B series array is generated in the section of this subroutine defined by the alternate entry SAVPWR, which is executed at each time step. Time histories of quantities in the B series arrays are stored for 100 equal time intervals which span the entire duration of the simulated plasma discharge. The B serics arrays are printed at the end of the simulation run.

The C series arrays are generated in the section defined by the alternate entry NEWPWR, which is executed only for those time intervals for which printout of the plasma profiles has been specified. A summary of plasma variable values at the plasma center is given for the course of the simulation in the array RNEWS.

Table 4 shows which elements in the labeled common blocks supply information as input and which elements are defined or altered for output.

Table 3. Physical processes and names of variables and arrays associated with them in subroutine ØRMPWR

Process	Contribution to Plasma Power	Total Amount of Energy Involved Time Integral of Power	Flux Profiles	Flux Through Plasma Outer Boundary	Power Flow Throughout Plasma	Values of Integrals of Profiles	Power Densities
Ohmic heating	PØH	PØX, B1			C1	SUMØ	PIN
Poloidal magnetic field						SUMBT	
Line radiation	PLR	PLRX, B4				SUMLR	
Electron-ion collisions						SUMEI	
(For electrons only)							
Electron thermal conduction			PED B2	PEX, B2			ETE
Electron convection			PCVE, C3				EB
Electron energy, internal	ZE, B6					SUMEX, SUMP^a	
Electron energy, neutral beam injection	PER	PHE, B8			C9	SNJE	EEØ
Ionization loss due to electrons	PEIL				C11	SUMQE	PEØ

Table 3 (continued)

Process	Contribution to Plasma Power	Total Amount of Energy Involved Time Integral of Power	Flux Profiles	Flux Through Plasma Outer Boundary	Power Flow Throughout Plasma	Values of Integrals of Profiles	Power Densities
Electron-ion transfer	PEI				C5		EIØ
Synchrotron bremsstrahlung	PLR				C4		SHR
(For protons only)							
Ion thermal conduction			PID, C6	PIX, B3			ETI
Ion convection			PCIV, C7				PB
Ion energy, internal	ZSI, B7					SUMIX, SUZI	
Ion energy, neutral beam injection	PIR	PHI, B9			C10	SNJI	EE1
Ionization loss due to charge exchange	PIIL	PCXX, B5			C8	SUMQI	PIØ, PCX
Ion-electron transfer	PIE				C12		EI1

[a] SUMP is also used as a different type of quantity in computation of gross energy confinement time.

Table 4. List of labeled commons and elements within them used by subroutine ØRMPWR

CØMMØN BLØCK	Input		Output	
	Variables	Arrays	Variables	Arrays
/CDCLBM/		TPD		
/ERGBAL/				EB, EEØ, EIØ, ETE, ETI, EE1, EI1
/FIELDS/	BT	B, E, ZJ	BTHE, ELI, VAX	QX
/GEØM/	AM, AREA, HR, RØ, VØL			
/IMPURT/		DENP, ZBR		
/INDEX/	TIC, N, NM1			
/IØNLØS/		QEE, QII, FF		
/IØNS/		DEN		TES
/NEUTRL/	FCDEX	QE1, QI1		SIGV, ZN1
/ØUTPT/			GAMIN, GAMØUT, PLR	CTE, TAUE, TGRØS
/PØWER/			EDØT, PED, PEI, PEIL, PID, PIE, PIIL, PØH	PB, PED, PIN, PIØ
/RADLØS/		A1		SHR
/SUMS/			SUMBT, SUMC, SUMEI, SUMLR, SUMØ, SUMP, SUMQE, SUMQI, ZE, ZSI, SNJE, SNJI	
/TEMP/		TE, TI, TØ, VEL		
/THRMDF/	TDE, TDI			

Table 4 (continued)

	Input		Output	
CØMMØN BLØCK	Variables	Arrays	Variables	Arrays
/TIME/	HT, TSTRT, TIM, TMAX			
/XPØRT/		D1, D6		
/XTRA/		ALF		F
/ZRAD/		QRD		
/CXLØS/		QCX		
/HIST/				B1, B2, B3, B4, B5 B6, B7, B8, B9
/HIST1/			PØX, PEX, PIX, PLRX, PCXX, SUMEX, SUMIX, PHE, PHI, ICM	
/RATIØS/			PCVE, PCVI, RCUR, RCVE, RCVI, TAX, TAXZ, TAX1	
/PINT/				C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12
/RATIØ1/			REI, RTH, RPLR, RDØT, RTHE, RTHI	
/CXLØS/				PCX
/NEWS/			NEWS	RNEWS

Called from: RESET
Subroutines called: PC
Commons required: CDCLBM, CURENT, CXLØS, ERGBAL, FIELDS, GEØM, HIST, HIST1, IMPURT, INDEX, IØNLØS, NEWS, RATIØS, RATIØ1, SUMS, TEMP, THRMDF, TIME, XPØRT, XTRA, ZRAD
Variables required: IT. See also Table 3 and Appendix II.
Variables changed: See Tables 3 and 5 and Appendix II. In addition, there are:

GAMIN — incoming neutral flux, number of particles/($cm^2 \cdot sec$)

GAMØUT — outgoing flux of charged particles = N_pV, number/($cm^2 \cdot sec$)

CTE — stored energy, mJ/cm^3

TAVE — conduction energy confinement time profile, msec

TGRØS — gross energy confinement time profile, msec

QS — plasma safety factor profile

F — particle confinement time profile, msec

BTHE — poloidal $\beta \equiv [2\pi\int_o^a rdr(N_eT_i + N_pT_p)]/[\frac{1}{2\mu_o} B^2_{pol}(a)]$

ELI — plasma inductance, H

VAX — toroidal voltage of plasma, V

TAX — gross electron confinement time, msec

TAXZ — gross ion confinement time, msec

TAX1 — confinement time characteristic of plasma transport and radiation, msec

TNUEM — minimum value of ν_e*, $msec^{-1}$

TNUIM — minimum value of ν_i*, $msec^{-1}$

EDØT — time derivative of electron energy

XFRI — value of integral of the electron density

FRING — number of microwave fringe shifts with respect to the vacuum due to the presence of plasma electrons

Table 5. Variables representing ratios of power transported by various physical processes to total power input (from ohmic heating plus neutral beam injection)

Ratio Variable	Physical Process
RCVE	Electron convection
RCVI	Ion (proton) convection
REI	Electron-ion transfer
RTH	Total thermal conduction
RPLR	Line radiation
RDØT	Electron heating or cooling
RTHE	Electron thermal conduction
RTHI	Ion (proton) convection

FUNCTIØN PC(T)

Function PC(T) generates the program current, i.e. the driving current for the plasma. It is a function constructed to fit the driving current that actually produces the plasma discharge. The form of the function used in this subroutine is:

PC = ZPC + ZFI*(1. - EXP(-T/TC)) ,

where

ZPC = initial value of the total current

ZFI = additional amount of total current after time TC

T = value of current elapsed simulation time

TC = rise time of the current

Called from:	ØRMPWR, LBM
Commons required:	CURENT, TIME
Variables required:	T
Variables changed:	PC (the function name)

SUBRØUTINE PRTØUT

Subroutine PRTØUT handles only output from the simulation calculations. No calculations are done in this subroutine. Use of alternate entry points puts various format statements at the disposal of the user so the statements can be used any number of times for printing out different arrays. This organization also permits the user merely to insert a call to an alternate entry point in any subroutine he chooses, thus gaining access to several write and format statements by inserting a single line when output is desired.

The alternate entries and the subroutines calling them are:

Alternate Entry	Called From
INIØUT	GETSET
PRØFIL	RESET
SUMARY	RESET
NUPRIN	NEUTRA
IMPRIN	IMPRTY
HISTØR	MAIN

Commons required: CØUNT, CXLØS, ERGBAL, FIELDS, HIST, IMPURT, INDEX, IØNS, MEASUR, NEUTRL, NEWS, ØUTPT, PINT, PØWER, RADLØS, RATIØS, TEMP, THRMDF, TIME, TSTØR, XPØRT, XTRA, ZRAD

Quantities printed:

Single variables — TIM, PØH, PEI, PED, PEIL, TAX, BTHE, ELI, FRING, PIE, PID, PIIL, VAX, TSTRT, GAMØUT, GAMIN, TAXZ, TAX1, RET, RTH, RPLR, RDØT, RTHE, RTHI, RCUR, PCVE, PCVI, RCVE, RCVI, TX1, TX2

Arrays — TE, TI, ZJ, B, E, DEN, VEL, ZN1, PIN, ETE, EB, EIØ, SHR, PEØ, EEØ, RHS, DEDT, TDE, EI1, ETI, PB, PCX, PIØ, EE1, RHS, DEDT, TDI, TGRØS, F, QS, D1, ALF, D6, DENP, TAUE, DHI, ZHI, ZBR, C1, C2, C3, C4, C11, C9, C12, C6, C7, C8, C10, ZN1, TØ, ICT, SPT, CARBØN, ØXYGEN, CC, CØ, B1, B2, B3, B4, B5, B6, B7, B8, B9, RNEWS

SUBRØUTINE RESET (IT,K,*)

Subroutine RESET uses the solutions generated by the PLASMA module to set the new values of the plasma variable profiles, and to compute the new values of ionization loss. It also prints the new profiles, calls for the power balance calculations, and prints a summary of the power balance. Figure 2 shows the order of calls to other subroutines and the logic of the time sequence which is observed by RESET.

The electron and ion temperature and plasma density profiles are checked at each time step to be sure they are in a reasonable range of physical values. If the values go out of range, an attempt is made to recover by force. The profiles are set to their edge boundary values from the radial point where trouble occurs on out to the plasma edge.

Called from: MAIN

Subroutines called: SAVPWR, PRØFIL, BMPWR, NEWPWR, SUMARY

Commons required: BDYCØN, BEAMC, FIELDS, GEØM, IBTMN, INDEX, IØNS, MEASUR, NEUTRL, RADLØS, TEMP, TIME, XTRA, ZRAD

Variables required:

IT — DØ loop variable in the main program for number of simulation time intervals that have elapsed

K — number of equations in the two-fluid plasma model treated by the PLASMA module (currently K = 7).

From labeled common blocks (see Appendix II for definitions)

/BDYCON/ — DENB, TEØ, TIØ

/BEAMC/ — DJT(101), TNJCT

/GEØM/ — HR

/IBTMN/ — X(360)

/INDEX/ — N, NR

/IØNS/ — DEN(101)

/TIME/ — HT, TIM, TMAX, TPR

Variables and arrays changed:

Single variables — XFRI, TPR

Arrays — TE, TI, BZ, DEN, A3, ZJ, B, E, VEL, DJE, TE12. See Appendix II.

A3 — time derivative of the electron density

ORNL-DWG 76-2316

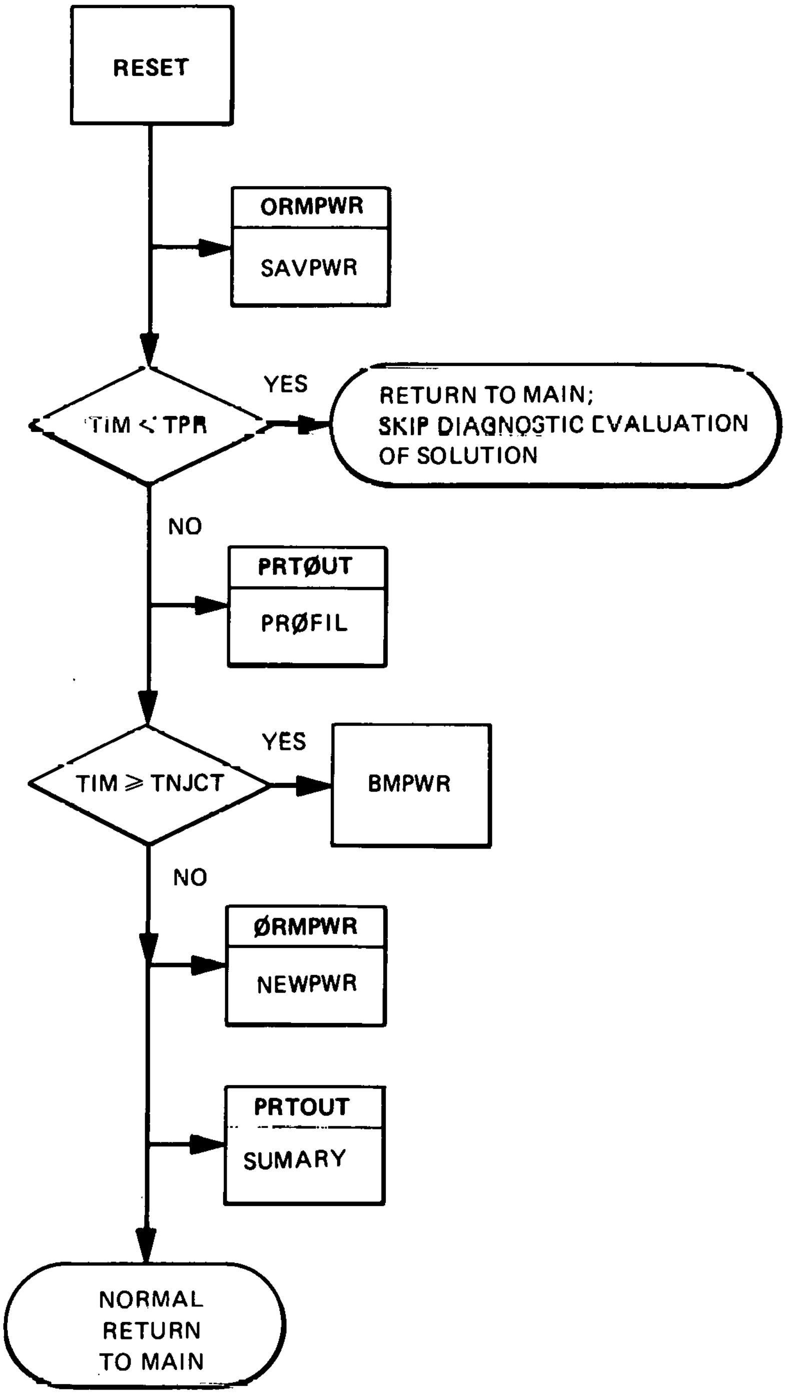


Fig. 2. Diagram of subroutine calls and logic of subroutine RESET.

SUBRØUTINE SIGMA (ZEO, ZTE, ZSIGCK, ZSIGI, ZSIGEL, NS)

Subroutine SIGMA calculates the cross sections for charge exchange σ_{cx}, ionization by ion impact σ_{ii}, and ionization by electron impact σ_{ie}. The cross sections for charge exchange and ionization by ion impact are taken from a paper by Riviere.[4] Cross sections for D and T are obtained by using the energy/nucleon to compute the velocity. The expressions used in the subroutine are:

$$\sigma_{cx} = \frac{6.937(10)^{-15}[1 - 0.155 \log_{10} T_i]^2}{1 = 1.112(10)^{-15} T_i^{3.3}} \text{ cm}^2 \quad , \quad T_i \text{ in eV}$$

$$\log_{10} \sigma_{ii} = - 0.8712(\log_{10} T_i)^2 + 8.156 \log_{10} T_i - 34.833 \quad .$$

The rate equation formula for ionization by electron impact was derived by R. Colchin[5] and is given by:

$$\sigma_{ie} = \begin{cases} (3.97 \ln \frac{T_e}{10} - 1.57) \dfrac{(10)^{-16}}{T_e} \text{ cm}^2 \quad , \quad T_e < 67.5 \text{ eV} \\ \\ (21.9 \ln T_e - 42.9) \dfrac{(10)^{-16}}{T_e} \text{ cm}^2 \quad , \quad T_e \geqslant 67.5 \text{ eV} \quad . \end{cases}$$

If the computed $\sigma_{ie} < 0$, then $\sigma_{ie} = 10^{-20}$ cm^2 .

Called from: CST

Variables required:

ZEØ — ion energy/nucleus, eV

ZTE — electron energy, eV

NS — switch with the following possible values:

= 1, compute σ_{cx} only

= 2, compute σ_{ii} only

= 3, compute σ_{ie} only

= 4, compute σ_{cx}, σ_{ii}, and σ_{ie}

Variables changed:

ZSIGCX — value of the charge-exchange cross section σ_{cx}, cm^2

ZSIGI — value of the cross section of ionization by ion impact σ_{ii}, cm^2

ZSIGEL — value of the cross section of ionization by electron impact σ_{ie}, cm^2

Limitations: The formulae given above represent fits to data and are valid in the following ranges:

σ_{cx}: 10 eV < T_i < 100 keV

σ_{ii}: 10 eV < T_i < 100 keV

σ_{ie}: 20 eV < T_e < 100 keV

SUBRØUTINE TIMSTP(IT)

Subroutine TIMSTP uses the timing information provided by the input to set up in a general way an array containing times when processes are turned on or off with respect to the beginning simulation time of the plasma discharge. The input switch arrays are used to select an initialization path for the timing and to determine the size of the time steps for the discharge. Small time increments can be chosen for the simulation calculation during periods of rapid changes due to fast current rise in the plasma and turn-on of neutral beam injection. The timing relationships are shown in Fig. 3.

Called from: MAIN

Commons required: BEAMC, INDEX, TMINDX, TIME

Variables required:

IT — DØ loop variable in the main program for number of simulation time intervals that have elapsed

From labeled common blocks (see Appendix II for definitions)

/BEAMC/	— TNJCT, INJ(10)
/INDEX/	— NT
/TMINDX/	— DTNJCT, N1, N3
/TIME/	— TC, TSTRT, TMAX

Variables and arrays changed:

First time the subroutine is called, the arrays TT, NN, HD.

First and all other times:

TIM — current value of elapsed time of the simulated plasma discharge, msec

HT — current time increment for the simulation, msec

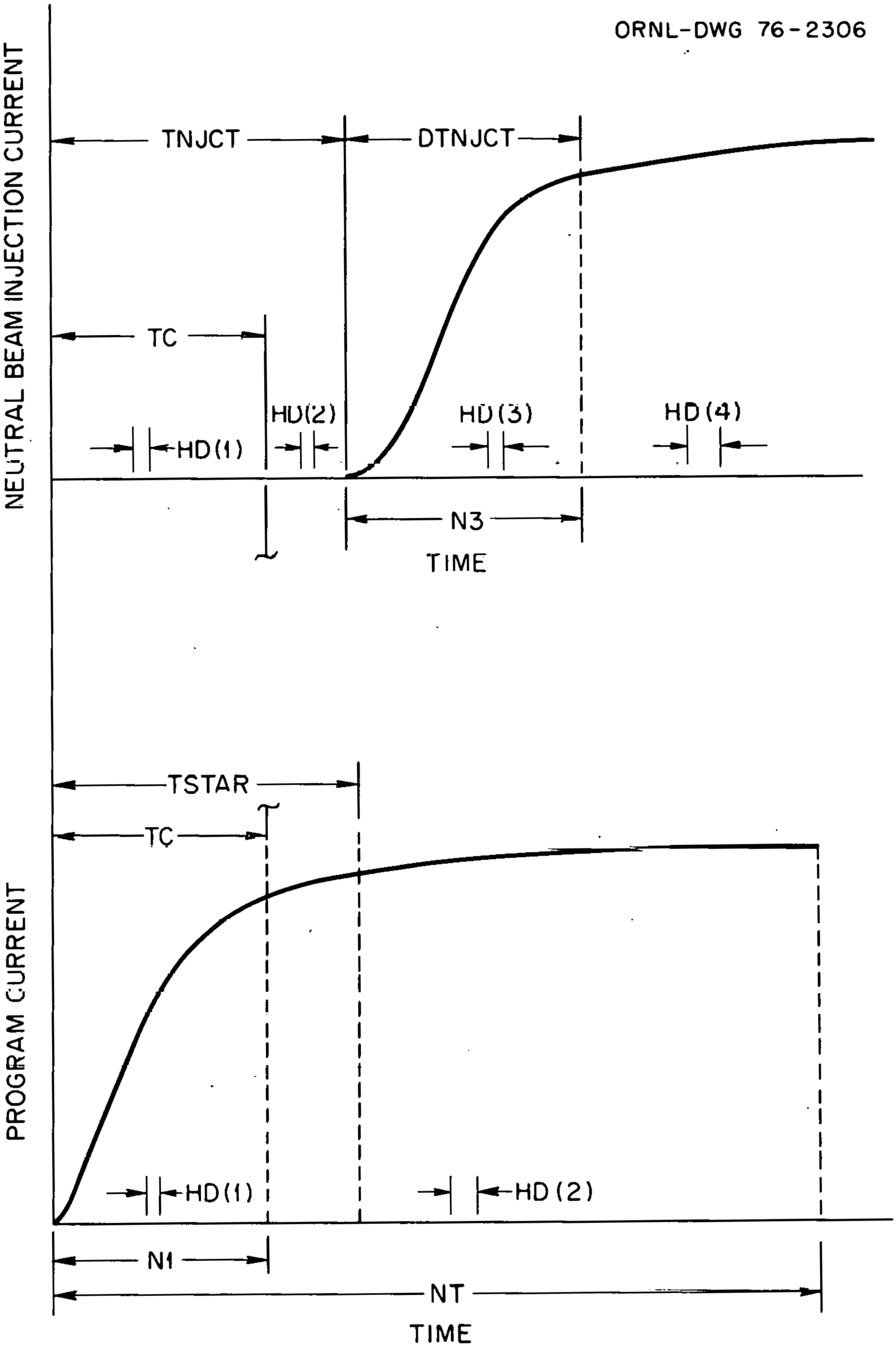


Fig. 3. Timing information used by plasma simulation code.

REFERENCES

1. C. W. Nestor, Jr., Kay C. Chandler, N. B. Gove, and J. D. McDowell, *User's Manual for the Graphics Package ØRGRAPH*, ORNL-4596 (November 1974); C. W. Nestor, Jr., *Addendum to User's Manual for the Graphic Package ØRGRAPH*, ORNL-4596 Addendum (January 1975).

2. "4010 and 4010-1 User's Manual," Tektronix, Inc., P.O. Box 500, Beaverton, Oregon 97005.

3. "Terminal Control System V.3 User's Manual" and "Advanced Graphing II User's Manual," Information Display Division, Tektronix, Inc., P.O. Box 500, Beaverton, Oregon 97005.

4. A. C. Riviere, *Nucl. Fusion* 11: 363 (1971).

5. R. J. Colchin, *Nucl. Fusion* 11: 329 (1971).

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

3. DESCRIPTION OF PLASMA MODULE

3.1 PURPOSE AND STRUCTURE OF MODULE

The plasma module contains the heart of the plasma simulation code. It is used to drive the evolution of the plasma with reference to the basic system of coupled partial differential equations describing the fluid model. This module makes no specific reference to plasma processes such as impurity diffusion effects, neutral gas transport, or the slowing down and trapping of an injected neutral beam. The influence of these processes on plasma evolution is provided through the rates of change of number, momentum, and energy of the fields and primary species which have been calculated in the pertinent module.

Subroutine CDC computes diffusion, particle, and energy conduction coefficients; resistivity; and associated derivatives with respect to the plasma variables. This information is then used to form the elements of the matrix needed to solve the system of equations. The matrix elements are computed and stored by subroutine LBM. The solutions are obtained by subroutine IBTM. The source terms and cross sections needed for the plasma model are computed by service subroutines CST and SIGMA, respectively.

The plasma module can be modified to include new physics or a different model for the plasma within the basic framework of the simulation code. The use of a different model for the diffusion coefficients will require changing some variables in subroutine CDC and perhaps making some corresponding changes in a few lines in subroutine LBM. The descriptions of the subroutines in this module are therefore written under the assumption that such changes will be made. The structure of the subroutines is presented in such a way as to aid the user in making his own changes.

3.2 BASIC INPUT/OUTPUT

The input required by this module includes machine geometry, toroidal magnetic field, plasma ion mass, poloidal magnetic field, electron and ion (proton) densities, temperatures (electron, ion, and neutral gas), Z_{eff}, transport and diffusion coefficients, electron and ion temperatures at the boundary, [Z], and current densities. The basic output is a new set of plasma variable profiles and resistivity.

3.3 SUMMARY OF FUNCTION OF EACH SUBROUTINE

SUBRØUTINE PLASIM

Subroutine PLASIM is the executive subroutine for the plasma module. It has the sole function of initializing plasma temperatures, densities, electromagnetic field profiles, particle diffusion velocity, and Z_{eff} at the beginning of the simulation. At subsequent times, it calls a series of subroutines which solve the set of coupled partial differential equations describing the basic plasma evolution. Figure 4 shows the subroutines which are called and the order of call.

Called from: CETSET [ENTRY IPPC(ITZ)] and MAIN [ENTRY PLASMA(ITZ)]

Subroutines called: CDC, CST, LBM, IBTM

Commons required: BDYCØN, CDCLBM, CDCLBX, CURENT, FIELDS, GEØM, IMPURT, INDEX, IØNS, TEMP, XPØRT

Variables required:

ITZ — time loop index variable

From labeled common blocks (see Appendix II for definitions)

/BDYCØN/ — DENB, DENØ, TEB, TEØ, T1Ø, X1(10)
/CURENT/ — ZPC
/GEØM/ — HR, R9
/IMPURT/ — ZEFF
/INDEX/ — N

Variables changed (initialized):

From labeled common blocks (see Appendix II for definitions)

/FIELDS/ — B(101), E(101), ZJ(101), ZJØ
/IØNS/ — DEN(101)
/TEMP/ — TE(101), TI(101), TØ(101), VEL(101)
/XPØRT/ — D1(101)

ORNL—DWG 76—2305

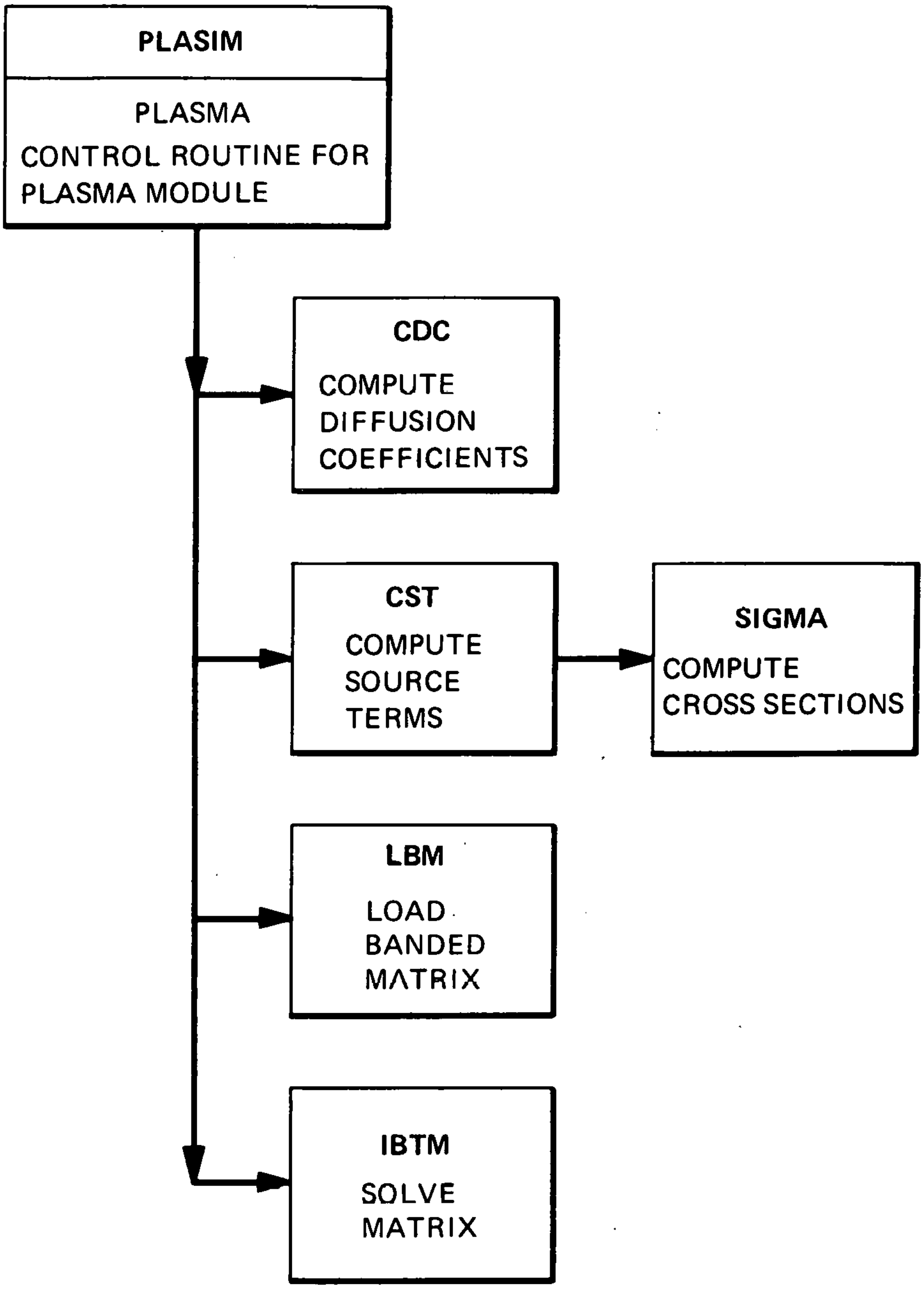


Fig. 4. Diagram of PLASMA module and subroutines called.

SUBRØUTINE CDC(IT)

Subroutine CDC computes diffusion coefficients, thermal conductivity profiles, resistivity, and associated derivatives with respect to the plasma variables used in the implicit space-time centered differencing scheme in subroutine LBM. The basic equations used in subroutine CDC and the models associated with them are summarized below.

The electron and ion collision frequencies are given by

$$\nu_e = \frac{4\sqrt{2\pi}\ e^4 Z^2 N_i\ \ell n\ \Lambda}{3\sqrt{m_e}\ T_e^{3/2}} = \frac{3.42(10)^{-8} N_e(cm^{-3})\ Z_{eff}}{T_e(eV)^{3/2}} \frac{\ell n\ \Lambda}{17.25}$$

$$\nu_i = \frac{4\sqrt{\pi}\ e^4 Z^4\ N_i\ \ell n\ \Lambda}{3\sqrt{m_i}\ T_i^{3/2}} = \frac{8(10)^{-10}\ N_i\ (cm^{-3})(1.41)\ Z_{eff}}{\sqrt{m_p(amu)}\ T_i(eV)^{3/2}} = \frac{\ell n\ \Lambda}{17.25}$$

where

$$\ell n\ \Lambda = 17.25 - \ell n \left\{ \frac{\sqrt{10^{-13}\ N_e\ (cm^{-3})}}{10^{-3}\ T_e\ (eV)} \right\} .$$

The ratios of the collision frequencies to the bounce frequencies are given by

$$\nu_e{}^* \equiv \frac{\nu_e}{\omega_{b,e}} = \nu_e\ (msec^{-1})\ R_o\ (cm)\ B_T\ (gauss) \sqrt{\frac{R_o(cm)}{r(cm)}}\ \frac{10^{-7}}{4.3\sqrt{T_e(eV)}\ B_p(gauss)}$$

$$\sim T_e{}^{-2} B_p{}^{-1} N_c$$

$$\nu_i{}^* \equiv \frac{\nu_i}{\omega_{b,i}} = 9.9\ \nu_i\ (msec^{-1})\ R_o(cm)\ B_T(gauss) \sqrt{\frac{R_o(cm)}{r(cm)}}\ \frac{10^{-7}}{\sqrt{T_i(eV)}\ B_p(gauss)}$$

$$\sim T_i{}^{-2} B_p{}^{-1} N_p$$

Let $\nu_e{}^{**}$ be the minimum value of $\hat{D}$ as a function of ν_e, and define

$$f = f(\nu_e{}^*) \equiv 1 + \left(\frac{\nu_e{}^{**}}{\nu_e{}^*} \right)^2 .$$

The basic pseudoclassical particle diffusion coefficient is

$$D = \nu_e \rho_e^{\,2} \left(\frac{B_T}{B_p} \right)^2 \gamma_3 \quad , \text{ in units of } \frac{\text{cm}^2}{\text{msec}} \quad ,$$

where γ_3 is a coefficient which is given in the input data and $\rho_e = (3.36 \sqrt{T_e})/B_T$ is the electron gyro-radius. This diffusion coefficient is modified further to give

$$\hat{D} = D\ f(\nu_e{*}) \sim T_e^{-1/2}\ B_p^{-2}\ N_e\ f(\nu_e{*})$$

for the particle diffusion coefficient of the model.

Three derivatives associated with this diffusion coefficient are required in subroutine LBM:

$$\frac{T_e}{\hat{D}} \frac{\partial \hat{D}}{\partial T_e} \, , \quad \frac{B_p}{\hat{D}} \frac{\partial \hat{D}}{\partial B_p} \, , \quad \text{and } \frac{N_e}{\hat{D}} \frac{\partial \hat{D}}{\partial N_e} \quad .$$

The first of these derivatives is evaluated to illustrate the form of these derivatives used in subroutine CDC:

$$\frac{T_e}{\hat{D}} \frac{\partial \hat{D}}{\partial T_e} = \frac{T_e}{\hat{D}} \left\{ \frac{\partial \hat{D}}{\partial D} \frac{\partial D}{\partial T_e} + \frac{\partial \hat{D}}{\partial f} \frac{\partial f}{\partial \nu_e{*}} \frac{\partial \nu_e{*}}{\partial T_e} \right\}$$

$$\frac{\partial \hat{D}}{\partial D} = f \qquad \frac{\partial D}{\partial T_e} = \tfrac{1}{2} \frac{D}{T_e}$$

$$\frac{\partial \hat{D}}{\partial f} = D \qquad \frac{\partial f}{\partial \nu_e{*}} = - \frac{2}{\nu_e{*}} \left(\frac{\nu_e{**}}{\nu_e{*}} \right)^2 \qquad \frac{\partial \nu_e{*}}{\partial T_e} = - \frac{\partial \nu_e{*}}{T_e}$$

so that

$$\frac{T_e}{\hat{D}} \frac{\partial \hat{D}}{\partial T_e} = \frac{T_e}{\hat{D}} \left\{ - \frac{D}{\partial T_e} + 4 \frac{D}{T_e} \left(\frac{\nu_e{**}}{\nu_e{*}} \right)^2 \right\}$$

$$= - \tfrac{1}{2} + \frac{4\ (\nu_e{**})^2}{(\nu_e{*})^2 + (\nu_e{*})^2} \quad .$$

In like manner,

$$\frac{B_p}{\hat{D}}\frac{\partial\hat{D}}{\partial B_p} = -2 + 2\,\frac{(\nu_e{**})^2}{(\nu_e{**})^2 + (\nu_e{*})^2}$$

and

$$\frac{N_e}{\hat{D}}\frac{\partial\hat{D}}{\partial N_e} = 1 - 2\,\frac{(\nu_e{**})^2}{(\nu_e{**})^2 + (\nu_e{*})^2} \quad .$$

The basic pseudo-classical electron thermal diffusion coefficient is

$$\chi_e = \nu_e\rho_e^{\,2}\left(\frac{B_T}{B_p}\right)^2 \gamma_1 \quad \text{, in units of } \left[\frac{\text{cm}^2}{\text{msec}}\right] \quad ,$$

where γ_1 is a coefficient which is given in the input data. The coefficient is modified further according to

$$\hat{\chi}_e = \chi_e f(\nu_e{*}) \sim T_e^{-1/2}\,B_p^{-2}\,N_e\,f(\nu_e{*}) \quad .$$

Three derivatives associated with this diffusion coefficient are also required for subroutine LBM and are given by

$$\frac{T_e}{\hat{\chi}_e}\frac{\partial\hat{\chi}_e}{\partial T_e} = -\tfrac{1}{2} + 4\,\frac{(\nu_e{**})^2}{(\nu_e{**})^2 + (\nu_e{*})^2}$$

$$\frac{B_p}{\hat{\chi}_e}\frac{\partial\hat{\chi}_e}{\partial B_p} = -2 + 2\,\frac{(\nu_e{**})^2}{(\nu_e{**})^2 + (\nu_e{*})^2}$$

$$\frac{N_e}{\hat{\chi}_e}\frac{\partial\hat{\chi}_e}{\partial N_e} = 1 - 2\,\frac{(\nu_e{**})^2}{(\nu_e{**})^2 + (\nu_e{*})^2} \quad .$$

The basic neoclassical ion (proton) thermal diffusion coefficient is

$$\chi_i = 1.41\ \nu_i\ \rho_i^2 \sqrt{\frac{r}{R_o}} \left(\frac{B_T}{B_p}\right)^2 \gamma_2 \quad \text{, in units of } \frac{\text{cm}^2}{\text{msec}} \quad ,$$

where γ_2 is a coefficient given in the input data and $\rho_i = \frac{\sqrt{2.09(10)^4 m_p T_i}}{B_T}$ is the proton gyro-radius. This diffusion coefficient is also modified further[1] to give

$$\hat{\chi}_i = \chi_i \frac{0.47\left[1 + 0.43\ \nu_i^*\left(\frac{r}{R_o}\right)^{3/2}\right]}{1 + 1.03\sqrt{\nu_i^*} + 0.18\ \nu_i^*} = \chi_i\ g(\nu_i^*) \sim T_i^{-1/2} B_p^{-2} N_i\ g(\nu_i^*).$$

The three derivatives associated with this diffusion coefficient are given by

$$\frac{T_i}{\hat{\chi}_i}\frac{\partial\hat{\chi}_i}{\partial T_i} = \frac{T_i}{\hat{\chi}_i}\left\{\frac{\partial\hat{\chi}_i}{\partial\chi_i}\frac{\partial\chi_i}{\partial T_i} + \frac{\partial\hat{\chi}_i}{\partial g}\frac{\partial g}{\partial\nu_i^*}\frac{\partial\nu_i^*}{\partial T_i}\right\}$$

where

$$\frac{\partial\hat{\chi}_i}{\partial\chi_i}\frac{\partial\chi_i}{\partial T_i} = g(\nu_i^*)\left[-\tfrac{1}{2}\frac{\chi_i}{T_i}\right] = -\tfrac{1}{2}\frac{\hat{\chi}_i}{T_i}$$

$$\frac{\partial g}{\partial\nu_i^*} = \frac{0.47\left\{0.43\left(\frac{r}{R_o}\right)^{3/2}\left(1 + 0.52\sqrt{\nu_i^*}\right) - \left(\frac{0.51}{\sqrt{\nu_i^*}} + 0.18\right)\right\}}{\left[1 + 1.03\sqrt{\nu_i^*} + 0.18\nu_i^*\right]^2}$$

$$\frac{\partial\hat{\chi}_i}{\partial g} = \chi_i \text{ and } \frac{\partial\nu_i^*}{\partial T_i} = -\frac{2\nu_i^*}{T_i} \quad ;$$

$$\frac{B_p}{\hat{\chi}_i}\frac{\partial\hat{\chi}_i}{\partial B_p} = \frac{B_p}{\hat{\chi}_i}\left\{\frac{\partial\hat{\chi}_i}{\partial\chi_i}\frac{\partial\chi_i}{\partial B_p} + \frac{\partial\hat{\chi}_i}{\partial g}\frac{\partial g}{\partial\nu_i^*}\frac{\partial\nu_i^*}{\partial B_p}\right\};$$

where

$$\frac{\partial\hat{\chi}_i}{\partial\chi_i}\,\frac{\partial\chi_i}{\partial B_p} = g\left[-\,\frac{2\chi_i}{B_p}\right] = -\,\frac{2\hat{\chi}_i}{B_p} \quad \text{and} \quad \frac{\partial\nu_i{}^*}{\partial B_p} = -\,\frac{\nu_i{}^*}{B_p} \quad ;$$

and

$$\frac{N_i}{\hat{\chi}_i}\,\frac{\partial\hat{\chi}_i}{\partial N_i} = \frac{N_i}{\hat{\chi}_i}\left\{\frac{\partial\hat{\chi}_i}{\partial\chi_i}\,\frac{\partial\chi_i}{\partial N_i} + \frac{\partial\hat{\chi}_i}{\partial g}\,\frac{\partial g}{\partial\nu_i{}^*}\,\frac{\partial\nu_i{}^*}{\partial N_i}\right\}$$

where

$$\frac{\partial\hat{\chi}_i}{\partial\chi_i}\,\frac{\partial\chi_i}{\partial N_i} = \frac{\hat{\chi}_i}{N_i} \quad \text{and} \quad \frac{\partial\nu_i{}^*}{\partial N_i} = \frac{\nu_i{}^*}{N_i} \quad .$$

The three diffusion coefficients and their associated derivatives are evaluated separately for their values at the origin:

$$\hat{D}(r=o) = \hat{D}(r=\Delta r) \qquad \frac{T_e}{\hat{D}}\,\frac{\partial\hat{D}}{\partial T_e} = 1 \qquad \frac{B_p}{\hat{D}}\,\frac{\partial\hat{D}}{\partial B_p} = 0 \qquad \frac{N_e}{\hat{D}}\,\frac{\partial\hat{D}}{\partial N_e} = 0$$

$$\hat{\chi}_e(r=o) = \hat{\chi}_e(r=\Delta r) \qquad \frac{T_e}{\hat{\chi}_e}\,\frac{\partial\hat{\chi}_e}{\partial T_e} = 1 \qquad \frac{B_p}{\hat{\chi}_e}\,\frac{\partial\hat{\chi}_e}{\partial B_p} = 0 \qquad \frac{N_e}{\hat{\chi}_e}\,\frac{\partial\hat{\chi}_e}{\partial N_e} = 0$$

$$\hat{\chi}_i(r=o) = \left[1.6 + \frac{(3.7)(980)j\,\sqrt{T_i}}{1.6\,B_T\nu_i}\right]\left(\frac{1.6\,B_T}{R_o j}\right)^2 \nu_i \left[\frac{2.09(10)^4 M_p T_i}{B_T^2}\right]$$

$$\left.\frac{T_i}{\hat{\chi}_i}\,\frac{\partial\hat{\chi}_i}{\partial T_i}\right\}_{r=o} = -\tfrac{1}{2} + \frac{(4.65)(980)j\sqrt{T_i}}{1.6\,B_T\nu_i}$$

$$\left.\frac{B_p}{\hat{\chi}_i}\frac{\partial\hat{\chi}_i}{\partial B_p}\right\}_{r=o} = -2 + \frac{(2.35)(980)j\sqrt{T_i}}{1.6\ B_T\nu_i}$$

$$\left.\frac{N_i}{\hat{\chi}_i}\frac{\partial\hat{\chi}_i}{\partial N_i}\right\}_{r=o} = 1 - \frac{(2.35)(980)j\sqrt{T_i}}{1.6\ B_T\nu_i}$$

Called from: PLASIM [ENTRY PLASMA]

Commons required: CDCLBM, CDCLBX, GEØM, FIELDS, IMPURT, INDEX, IØNS, TEMP, THRMDF, XPØRT, XTRA

Variables required: (see Table 6 and Appendix II)

IT — time loop index variable

Arrays changed:

RB = $1/B_0$, reciprocal of the poloidal magnetic field

RJ = $1/j$, reciprocal of the current density

RTE = $1/T_e$, reciprocal of the electron temperature

RTI = $1/T_i$, reciprocal of the ion temperature

RTE12 = $T_e^{-1/2}$

RTE32 = $T_e^{-3/2}$

RTE52 = $T_e^{-5/2}$

TE12 = $T_e^{1/2}$

ALF = ν_e*, ratio of the electron collision frequency to the electron bounce frequency

D6 = ν_i*, ratio of the ion collision frequency to the ion bounce frequency

TDP = $\hat{D}$, the particle diffusion coefficient (pseudoclassical)

WPT = $(T_e/\hat{D})(\partial\hat{D}/\partial T_e)$

WPB = $(B_p/\hat{D})(\partial\hat{D}/\partial B_p)$

WPN = $(N_e/\hat{D})(\partial\hat{D}/\partial N_e)$

TDE = $\hat{\chi}_e$, the electron thermal diffusion coefficient (pseudoclassical

WE = $(T_e/\hat{\chi}_e)(\partial\hat{\chi}_e/\partial T_e)$

EE = $(B/\hat{\chi}_e)(\partial\hat{\chi}_e/\partial B)$

WGE = $(N_e/\hat{\chi}_e)(\partial\hat{\chi}_e/\partial N_e)$

TDI = $\hat{\chi}_i$, the ion thermal diffusion coefficient (neoclassical)

WI $= (T_i/\hat{\chi}_i)(\partial\hat{\chi}_i/\partial T_i)$

EI $= (B/\hat{\chi}_i)(\partial\hat{\chi}_i/\partial B)$

WGI $= (N_i/\hat{\chi}_i)(\partial\hat{\chi}_i/\partial N_i)$

ETA $= \eta$, the plasma resistivity

Significant internal variables:

CZE $= \dfrac{4\sqrt{2\pi}\,e^4 N_p}{3\sqrt{m_e}} = 3.42(10)^{-8}\,N_p$, $N_p \equiv$ proton density

CZI $= \dfrac{4\sqrt{\pi}\,e^4\,N_p}{3\sqrt{m_p}} = 8(10)^{-10}\,\dfrac{N_p}{\sqrt{m_p}}$

ZL $= \dfrac{\ell n\,\Lambda}{17.25} = \dfrac{1}{17.25}\left\{1 - \ell n\left[\dfrac{\sqrt{10^{-13}\,N_e}}{10^{-3}\,T_e}\right]\right\}$, the Coulomb logarithm factor

ZNUE $= \nu_e \equiv \dfrac{1}{\tau_e} = 3.42(10)^{-8}\,N_p T_e^{-3/2}\,Z_{eff}\,\dfrac{\ell n\,\Lambda}{17.25}$, the electron collision frequency, msec^{-1}

ZNUI $= \nu_i \equiv \dfrac{1}{\tau_i} = 8(10)^{-10}\,\dfrac{N_p}{\sqrt{m_p}}\quad T_i^{-3/2}\,\sqrt{2}\,Z_{eff}\,\dfrac{\ell n\,\Lambda}{17.25}$, the ion collision frequency, msec^{-1}

QQ $= \dfrac{rB_T}{R_o B_p}$, the safety factor

RKI $= \dfrac{0.47\left[1 + 0.43\nu_i{*}\left(\dfrac{r}{R_o}\right)^{3/2}\right]}{1 + 1.03\sqrt{\nu_i{*}} + 0.18\nu_i{*}}$, the ion thermal diffusion modification factor

GKI $= \dfrac{\partial g(\nu_i{*})}{\partial \nu_i{*}} = \dfrac{0.47\left\{0.43\left(\dfrac{r}{R_o}\right)^{3/2}\left[1 + 0.52\sqrt{\nu_i{*}}\right] - \left[0.18 + \dfrac{0.51}{\sqrt{\nu_i{*}}}\right]\right\}}{\left[1 + 1.03\sqrt{\nu_i{*}} + 0.18\nu_i{*}\right]^2}$

$$\text{FACT} = f(\nu_e^*) = 1 + \left(\frac{\nu_e^{**}}{\nu_e^*}\right)^2$$

$$\text{TRAP} = \frac{f(\nu_e^*)-1}{f(\nu_e^*)} = \frac{(\nu_e^{**})^2}{(\nu_e^{**})^2 + (\nu_e^*)^2}$$

RHØE2 $= \rho_e{}^2 = \left(\frac{3.36}{B_T}\right)^2 T_e$, the square of the electron gyroradius

RHØI2 $= \rho_i{}^2 = \dfrac{2.09(10)^4\ m_p T_i}{B_T{}^2}$, the square of the ion gyroradius

Table 6. List of labeled commons and elements within them used by subroutine CDC

CØMMØN BLØCK	Input		Output	
	Variables	Arrays	Variables	Arrays
/CDCLBM/				EE, EI, RB, RJ, RTE, RTI, RTE12, RTE32, RTE52, TDP, WE, WI
/CDCLBX/				WPT, WPB, WPN, ETA, WGE, WGI
/GEØM/	HR, RØ			
/FIELDS/	BT	B, ZJ		
/INPURT/		DENP		
/INDEX/	N			
/IØNS/	AMU	DEN		
/TEMP/		TE, TI, TØ		
/THRMDF/				TDE, TDI
/XPØRT/		D1, D11		D6
/XTRA/				ALF, TE12

SUBRØUTINE CST

Subroutine CST computes the source terms: i.e., the particles and power delivered to the electrons and ions. Contributions to the electron power are a gain from the injected neutral beam heat input, losses due to heating newly ionized neutral hydrogen by electron impact, and radiation losses due to electron impact with impurities. Contributions to the proton power are a gain from the injected neutral beam heat input, losses due to heating newly created protons, and charge exchange. The rate R_{ie} for ionization of neutral hydrogen by electron impact is computed as

$$R_{ie} = N_p N_o <\sigma v>_{ionization} \quad , \; sec^{-1}$$

where $<\sigma v>_{ion}$ is the rate coefficient for neutral hydrogen ionization by electron impact.

Called from: PLASIM [ENTRY PLASMA]

Subroutines called: SIGMA

Commons required: CXLØS, INDEX, IØNLØS, IØNS, NEUTRL, QT, TEMP, ZRAD

Variables required:

From labeled common blocks (see Appendix II for definitions)

/INDEX/	— N
/IONS/	— DEN(101)
/NEUTRL/	— ZN1(101)
/TEMP/	— TE(101), TI(101), TØ(101)
/ZRAD/	— QRD(101)

Arrays changed:

FF — rate R_{ie} for ionization of neutral hydrogen by electron impact

QEE — electron power loss due to ionization of neutral hydrogen by electron impact

QII — proton power loss due to ionization of neutral hydrogen by electron impact

QE, QI — power gain to the plasma electrons and protons, the source terms

QCX — power loss to the plasma protons due to charge exchange with neutral hydrogen

SUBRØUTINE LBM(IT)

Subroutine LBM calculates the elements of the banded matrix which is used in subroutine IBTM to give the solution to the plasma variable profiles at each new time step. The method used here was originally devised by Widner and Dory.[2] The banded matrix is loaded according to a scheme which assigns a unique location MIJK according to the spatial grid number (I) used for the differencing, the model equation number (J), and the variable number (K). This scheme is summarized in Table 7.

The differencing scheme is straightforward conceptually, but can become tedious and time-consuming to work out in practice. The method is summarized so that if a user wants to change the model, he will have a reference for proceeding to derive the new matrix elements required. The general form of the equations is

$$\frac{\partial u}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}[rg\frac{\partial h}{\partial r}] + A \equiv \text{rhs} \quad .$$

Let the subscripts I-1, I, and I+1 be used to denote spatial grid points for r-Δr, r, and r+Δr. Let superscripts N, $N+\frac{1}{2}$, and N+1 denote time grid points for t, $t+\frac{\Delta t}{2}$, and t+Δt. The rules for evaluating the matrix elements are as follows:

1. Time derivatives: $\frac{\partial u}{\partial t} = \frac{1}{\Delta t}\left[u^{N+1} - u^{N}\right]$

2. First order spatial derivatives: $\frac{\partial h}{\partial r} = \frac{1}{2\Delta r}\left[h^{N}_{I+1} - h^{N}_{I-1}\right]$

3. Second order spatial derivatives: $\frac{\partial^2 h}{\partial r^2} = \left(\frac{1}{\Delta r}\right)^2\left[h^{N}_{I+1} - 2h^{N}_{I} + h^{N}_{I-1}\right]$

4. Each plasma variable φ on the right-hand side of the equation is evaluated at the midpoint of the interval (t, t+Δt) according to the prescription:

$$[\phi]^{N+1/2}_{I} = 1/2\left\{[\phi]^{N+1} + [\phi]^{N}\right\}$$

Table 7. Matrix element assignment scheme for banded matrix MIJK

I	Term Represented by Index I
1	j-1
2	j spatial grid point indices
3	j+1
4	"right-hand side" of equations in finite difference form

J,K	Variable (K)	Equation (J)
1	T_e:	$\frac{3}{2}\frac{\partial}{\partial t}(N_eT_e) = \frac{-1}{r}\frac{\partial}{\partial r}[r(q_e + \frac{3}{2}N_eVT_e)] - 3\frac{m_e}{m_p}\frac{T_e-T_i}{\tau_e}N_e[Z] + E \cdot J - Q_{rad} - Q_{HiZ}$ $+ QE1 + \frac{3}{2}T_oN_eN_o <\sigma v(T_e)>$
2	T_i:	$\frac{3}{2}\frac{\partial}{\partial t}(N_pT_i) = -\frac{1}{r}\frac{\partial}{\partial r}[r(q_i + \frac{3}{2}N_pVT_i)] + 3\frac{m_e}{m_p}\frac{T_e-T_i}{\tau_e}N_p - N_oN_pR_{cx}\frac{3}{2}(T_i-T_o)$ $+ QI1 + \frac{3}{2}T_oN_eN_o <\sigma v(T_e)>$
3	J:	$\frac{\partial j}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial E}{\partial r})$
4	B:	$\frac{\partial B}{\partial t} = \frac{\partial E}{\partial r}$
5	E	$E = \eta(j-j_B)$
6	N:	$\frac{\partial}{\partial t}N_e + \frac{1}{r}\frac{\partial}{\partial r}(rN_eV) = N_oN_eR_i(T_e) + \Sigma_{imp} + \Sigma_{\text{neutral beam injection}}$
7	V:	$N_eV = -D\frac{\partial N}{\partial r}$

where $q_e = -N_e\chi_e\frac{\partial T_e}{\partial r}$, $q_i = -N_p\chi_i\frac{\partial T_i}{\partial r}$, and j_B is the additional current density produced by the injected neutral beam. QE1 and QI1 are complicated source terms for which there are no single analytic expressions.

where

$$[\phi]^{N+1} = [\phi]^{N} + \Delta t \frac{\partial[\phi]}{\partial t} + [o(\Delta t)^2]$$

For the purposes of the computations, one is actually evaluating

$$\left[\frac{\partial u}{\partial t}\right]_I = R \left\{ \left[\phi_a\right]_I^{N+1/2}, \left[\phi_b\right]_I^{N+1/2}, \dots \left[\phi_n\right]_I^{N+1/2} \right\} .$$

Once all the equations have been treated in this fashion, they will be in the form

$$\{\alpha_1\} T_{e_{I-1}}^{N+1} + \{\alpha_2\} T_{e_I}^{N+1} + \{\alpha_3\} T_{e_{I+1}}^{N+1} + \{\beta_1\} T_{i_{I-1}}^{N+1} + \{\beta_2\} T_{i_I}^{N+1}$$

$$+ \dots = \text{RHS}$$

where the α_i, β_i, etc. are the matrix elements. Using the T_e equation as an example, the matrix elements MIJK would be

M111 = α_1	M112 = β_1	...	M411 = RHS
M211 = α_2	M212 = β_2	...	etc.
M311 = α_3	...	...	

Time derivatives involving the diffusion and transport coefficients should be handled in a general way using the chain rule and functional derivatives according to

$$\frac{\partial D}{\partial t} = \frac{\partial D}{\partial T_e} \frac{\partial T_e}{\partial t} , \dots$$

since subroutine CDC computes $\frac{T_e}{D} \frac{\partial D}{\partial T_e}$, ... (see the description of subroutine CDC, Arrays changed). The correspondence between the matrix elements in LBM and the terms in the partial differential equations is shown in Table 8.

Table 8. Correspondence between terms in plasma equations and matrix elements [IJK] in LBM.

T_e Equation

$$\frac{3}{2}\frac{\partial}{\partial t}(N_eT_e) = -\frac{1}{r}\frac{\partial}{\partial r}[r(q_e+\frac{3}{2}N_eVT_e)] - 3\frac{m_e}{m_p}\frac{T_e-T_i}{\tau_e}[Z]\,N_e + E\cdot j + \frac{3}{2}T_oN_eN_o<\sigma v(T_e)> + Q_{rem}$$

[411]	[111]	[111]	[211]	[212]	[213]	[411]	[411]
[211]	[311]	[311]	[411]	[411]	[215]		
	[211]	[217]	[216]		[411]		
	[114]	[117]					
	[214]	[317]					
	[314]	[116]					
	[116]	[216]					
	[216]	[316]					
	[316]	[411]					
	[411]						

where $Q_{rem} = QE1 - Q_{rad} - Q_{HiZ}$

T_i Equation

$$\frac{3}{2}\frac{\partial}{\partial t}(N_pT_i) = -\frac{1}{r}\frac{\partial}{\partial r}[r(q_i+\frac{3}{2}N_pVT_i)] + 3\frac{m_e}{m_p}\frac{T_e-T_i}{\tau_e}N_p - N_oN_pR_{cx}\frac{3}{2}(T_i-T_o) + \frac{3}{2}T_oN_eN_o<\sigma v(T_e)> + QI1$$

[421]	[122]	[122]	[221]	[222]	[222]	[222]	[421]
[222]	[322]	[322]	[421]	[421]		[421]	
	[124]	[227]					
	[224]	[421]					
	[324]	[127]					
	[421]	[327]					

Table 8 (continued)

J Equation

$$\frac{\partial j}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left(r\,\frac{\partial E}{\partial r}\right)$$

[233]	[135]
[431]	[235]
	[335]
	[431]

V Equation

$$N_e V = -D\,\frac{\partial N_e}{\partial r}$$

[277]	[176]
[471]	[276]
	[376]
	[274]
	[271]
	[471]

B Equation

$$\frac{\partial B}{\partial t} = \frac{\partial E}{\partial r}$$

[244]	[145]
[441]	[345]
	[441]

E Equation

$$E = \eta(j-j_B), \quad \eta = \eta(Z_{eff}, T_e, \ldots)$$

[255]	[253]
	[251]
	[451]

N Equation

$$\frac{\partial N_e}{\partial t} + \frac{1}{r}\frac{\partial}{\partial r}(rN_eV) = \underbrace{N_oN_eR_i(T_e) + \Sigma_{imp} + \Sigma_{NBI}}_{[461]}$$

[266]	[266]	[461]
[461]	[166]	
	[366]	
	[167]	
	[367]	
	[267]	
	[461]	

The matrix elements MIJK occur for each of the grid points in the spatial net, so the general matrix element is A(MX + MIJK), with MX = 0*MK, 1*MK, 2*MK, ..., N&MK where N is the number of radial grid points and MK = (3K+1)K, with K the number of equations (seven). The spatial net is divided into three intervals corresponding to MX = 0*MK, 1*MK through (N-1)*MK, and N*MK. The matrix elements for MX = 1*MK through (N-1)*MK all have the same general form while those for MX = 0*MK and N*MK are different since the boundary conditions must be incorporated at these two end points. At the plasma center (MX=0*MK), the boundary conditions are

$$\frac{dT_e}{dr} = \frac{dT_i}{dr} = \frac{dN}{dr} = 0 \quad ;$$

$$B(r=0) = 0 \quad ;$$

$$j = \nabla x B \quad ;$$

$$E = \eta j \quad ;$$

and at the plasma edge (MX = N*MK), the boundary conditions are

$$T_{e,i}(a) = T_{e,i}^{o} \quad ; \quad N(a) = N_e^{a} \quad ; \quad B(a) = .2I/a \quad ;$$

$j = \nabla x B$ (extrapolating from the values of B and $\frac{dB}{dr}$ at $r = a - \Delta r$);

$$E = \eta j.$$

A few matrix elements are evaluated in Appendix III to illustrate the method for a simple case and a complex one.

Called from:	PLASIM [ENTRY PLASMA]
Subroutines called:	PC
Commons required:	BDYCØN, BEAMC, CDCLBM, CDCLBX, GEØM, FIELDS, IMPURT, INDEX IØNLØS, IØNS, LBMIBT, NEUTRL, QT, TEMP, THRMDF, TIME, XPØRT
Variables required:	(See Table 9 and Appendix II)

IT — time loop index variable

Array changed:

A — matrix of coefficients of the system of differenced equations

Table 9. List of labeled commons and elements within them used by subroutine LBM

CØMMØN BLØCK	Input		Output	
	Variables	Arrays	Variables	Arrays
/BDYCØN/	TEØ, TIØ			
/BEAMC/		DJT		
/CDCLBM/		EE, EI, RB, RTE, RTI, RTE12, RTE32, RTE52, TDP, WE, WI		
/CDCLBX/		WPT, WPB, WPN, ETA		
/FIELDS/		B, E, ZJ		
/GEØM/	AM, HR			
/IMPURT/		DENP, ZBR		
/INDEX/	N, NM1			
/IØNLØS/		FF		
/IØNS/	AMU	DEN		
/LBMIBT/				A
/NEUTRL/		ZN1		
/QT/		ZE, QI		
/TEMP/		TE, TI, TØ, VEL		
/THRMDF/		TDE, TDI		
/TIME/	HT, TIM			
/XPØRT/		ZL		

SUBRØUTINE IBTM(IT)

Subroutine IBTM approximates the solution to the system of linear equations

FX=B

where F is a block tridiagonal matrix of order N and each block element of F is a K by K submatrix. The approximation is determined by Gaussian elimination without any pivoting. Let F(I,J) denote the (I,J) block element of the matrix F and let B(I) denote the associated K element subvector of the vector B. The 3K+1 by N*K array A in common LBMIBT contains on entry into IBTM the nontrivial information to describe the coefficient matrix F and the right-hand side B. The transpose of the array A is displayed since it portrays the information more naturally.

$$\text{transpose (A)} = \begin{pmatrix} 0 & F(1,1) & F(1,2) & B(1) \\ F(2,1) & F(2,2) & F(2,3) & B(2) \\ \vdots & \vdots & \vdots & \vdots \\ F(N-1,N-2) & F(N-1,N-1) & F(N-1,N) & B(N-1) \\ F(N,N-1) & F(N,N) & 0 & B(N) \end{pmatrix}$$

Upon exit from IBTM, the array X in common IBTMN contains the generated solution to the system of linear equations.

Called from: PLASIM [ENTRY PLASMA]

Variables required:

A — contains the nontrivial information of the block tridiagonal coefficient matrix for the linear system of differenced equations obtained from the set of partial differential equations; also contains the right-hand side

N — number of radial mesh points

K — number of partial differential equations describing the plasma

Arrays changed:

A — destroyed

X — solution vector, at the new time step, of the plasma variable profiles

REFERENCES

1. R. D. Hazeltine and F. L. Hinton, Phys. Fluids 16: 1883 (1973).

2. M. M. Widner and R. A. Dory, *An Implicit Numerical Method for Simulating Tokamak Plasma Discharges*, ORNL/TM-3498 (July 1971).

4. DESCRIPTION OF NEUTRAL GAS MODULE

4.1 PURPOSE AND STRUCTURE OF MODULE

The neutral gas module models the effects of neutral gas atoms in the plasma. Two distinct species of these atoms are present: hot and cold. The cold neutrals diffuse into the interior of the plasma from the area next to the container wall. When a cold neutral collides with a plasma ion, it can scatter elastically or charge exchange. The result of either process is production of a cold ion and a hot neutral. The purpose of this module is to determine the density distribution of hot and cold neutrals in the plasma. A calculation of the charge-exchange spectrum is included to allow comparison with experiment.

Four approaches are available for modeling the neutral gas effects: 1) an analytic expression based on a series of calculations using a detailed slab model, 2) a detailed slab model calculation without wall reflections,[1] 3) a slab model with wall reflections,[2] and 4) a Monte Carlo transport code calculation (NUTRLSN).[3] The analytic model options will always be available in the program. The remaining three options cannot be simultaneously linked together to form the object module of the entire simulation program. This restriction exists to reduce core memory requirements. One of options 2-4 must be selected and the subroutines for that option linked with the rest of the program. This linkage is accomplished on IBM 360/370 series computers by inserting an INCLUDE card with a reference to the name of the cataloged data set containing the subroutines for the desired option as input to the link step. Until options 3 and 4 have been implemented, option 2 will be assumed to be the only one available in the neutral gas module. Figure 5 shows the structure of the subroutine calls for this option. Figure 6 shows the slab geometry used in constructing the model. The user is reminded that the data required for the detailed slab model (DSM) calculations, as well as most of the data generated by the calculations, are communicated to the various subroutines through the labeled commons displayed in Table 10.

ORNL-DWG 76-2308

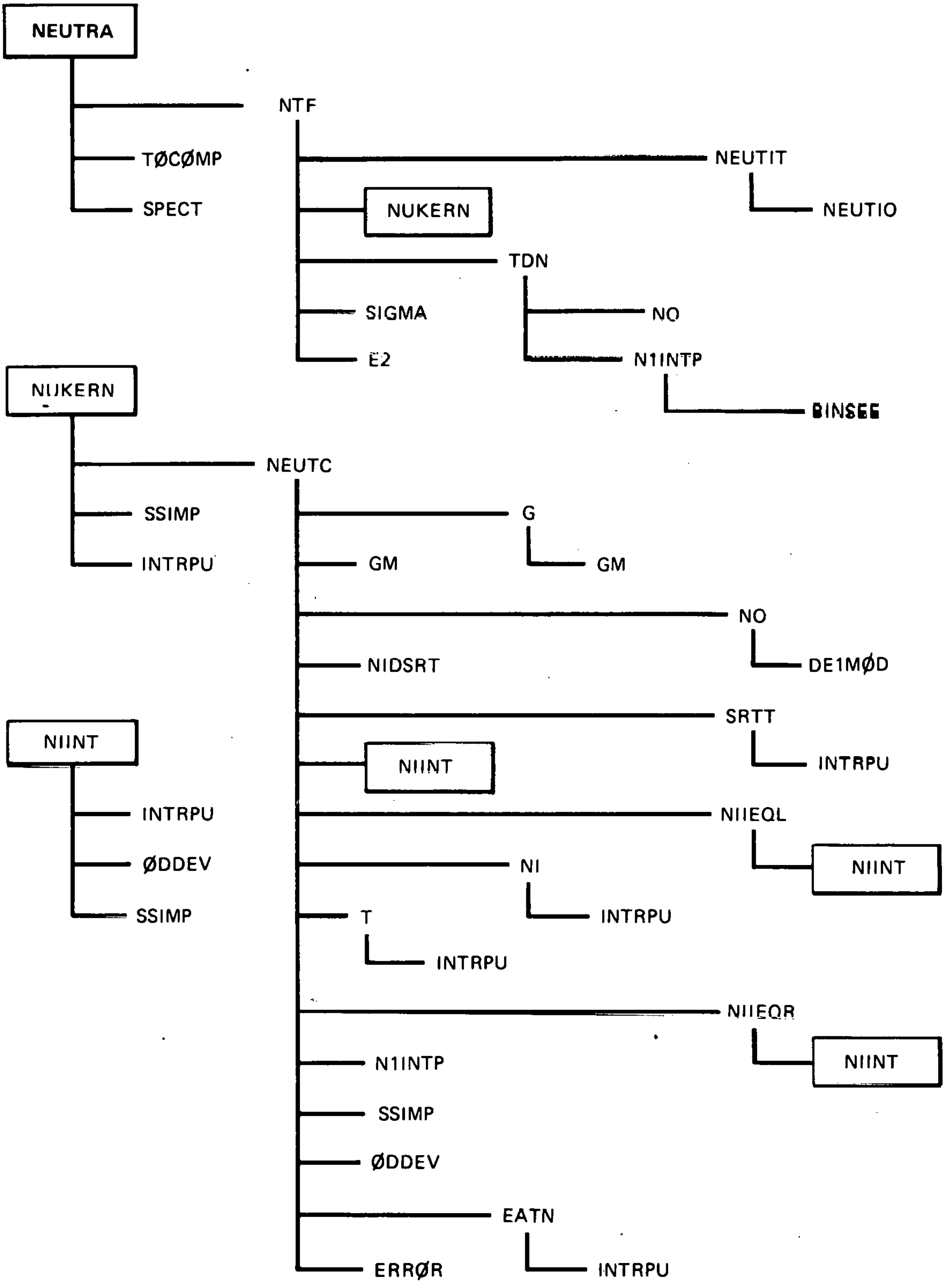


Fig. 5. Diagram of neutral gas module subroutine calls.

ORNL-DWG 76-2309

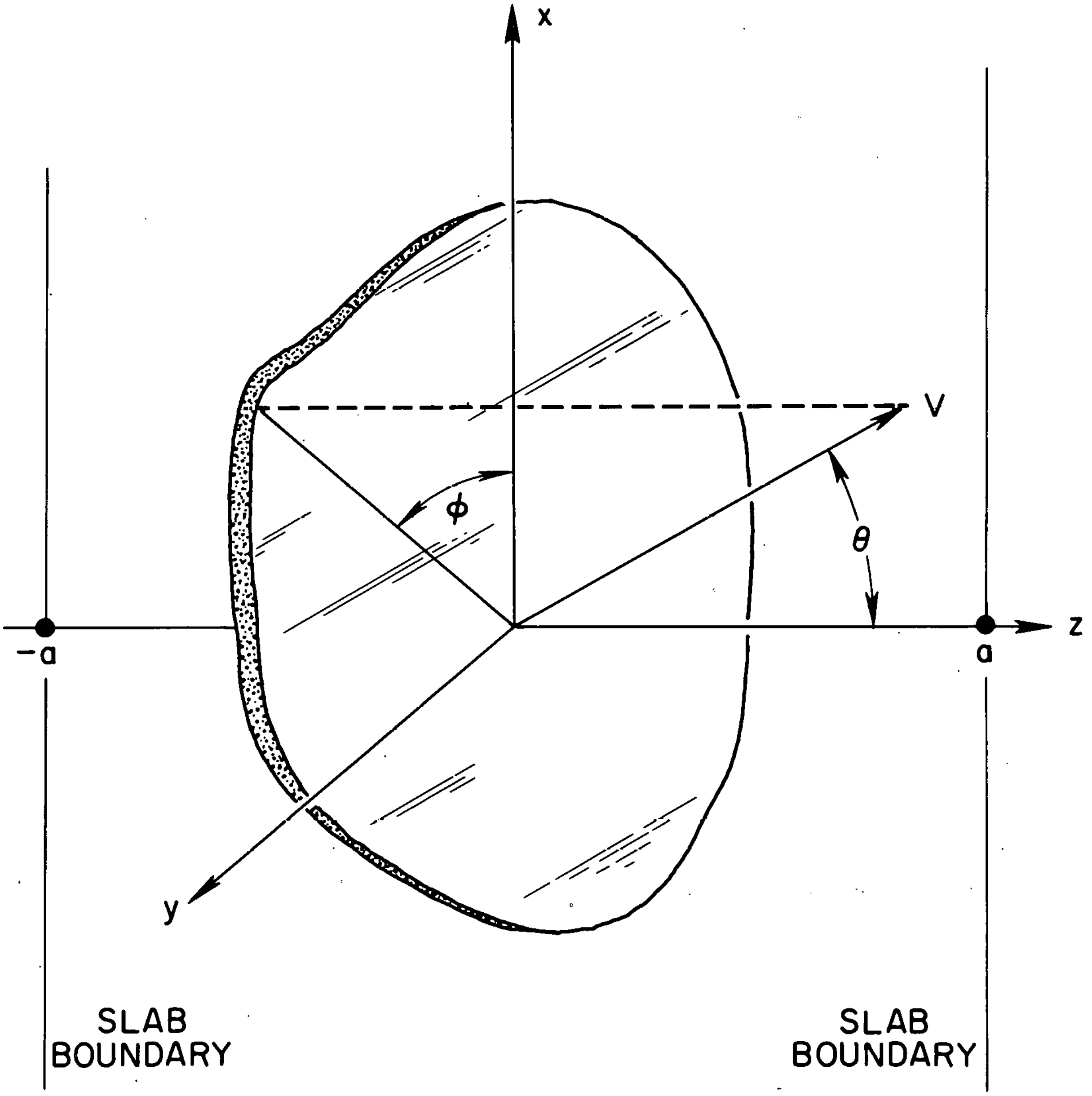


Fig. 6. Slab geometry used in detailed model of neutral gas diffusion and charge exchange.

Table 10. List of labeled commons and their contents which are used in the detailed slab model (DSM) calculation

```
CØMMØN/NUTRLP/ NØZ, NØX, NSIMP, NUNIF, QF, QC, DZ, DZD2, DZD3, EPSN1,
1SIMPM1, BDSQTX, TCINT, NØXM1, NØXP1, NØZP1, NSC, ININ, ITIN, IN1, IEATN,
2IINIOX, IINIX1, IX, IZ, IJL, IJH, IIJ, IRAT, NWDW1, MWDSC,
3N, ANM1, DSP, A, B, C, W1(1)
CØMMØN/INTRPS/ INTRPS, NSMO, NSMO21, IDEBUG
CØMMØN/NEUTPT/ contains pointers that partition the array WSC
CØMMØN/PRNT/ IPA, IPNO
CØMMØN/SLAB/ EØ, EPSNO, NWDWSC
```

4.2 BASIC INPUT/OUTPUT

The neutral gas module requires a great deal of input related to the numerical algorithms used to perform the integrations and to obtain the neutral profiles specified by the choice of model. Once set, most of this input should not need changing. This is the reason for setting many of these values in a BLØCK DATA section of the program rather than reading them in every time. The physics input includes the electron and ion temperatures and densities and the energy and density of the Franck-Condon neutrals at the plasma surface. The neutral gas module generates the profiles of the hot, cold, and total neutral densities and a total neutral density temperature. A subroutine SPECT computes a charge-exchange spectrum and any other quantities which can be related to experimental measurements and can be computed in terms of the variables appearing in the models used. Subroutine SPECT will most likely be modified by the user to reflect the type of experimental measurements and apparatus used for plasma diagnostics designed to obtain information on neutral gas effects.

4.3 DESCRIPTION OF THE ANALYTIC MODEL FOR THE TOTAL NEUTRAL DENSITY

The analytic model is based on the results of a number of cases run using the detailed slab model.[1] Let

$$\text{DMAX} = 10^{-13}\max\left\{10^{11},\ \max_{x}\{N_e(x)\}\right\}$$

and

$$\text{ATT} = \frac{85.4\ T_i(0)}{200(\text{DMAX*AM})^2}$$

where $T_i(0)$ is the ion temperature at the center, N_e the electron density, and AM the minor radius of the torus. Then the analytic approximation, $\overline{N}_T$, to the total neutral density is given by

$$\overline{N}_T(X) = .5*(\text{ATT}(1+2*X^{\text{NUTE}(2)}) + E_2(\text{ZZ}))$$

where

$$\text{ZZ} = .25*\text{DMAX*AM*}(1-X)(1-(1+X+X^2)/3)$$

and E_2 is the exponential integral of the second kind. NUTE(2) is a user-specified input value.

The scaled, total neutral density is obtained by multiplying $\overline{N}_T$ by the Franck-Condon boundary density N_{FC} where

$$N_{FC} = (\text{FCDEN} - \text{FCF})*\text{EXP}(-\text{TIM}/\text{TP}\emptyset)+\text{FCF} \quad .$$

TIM is the current time and FCDEN, FCF, and TP∅ are input parameters.

4.4. SLAB MODEL APPROXIMATION TO THE NEUTRAL GAS DENSITY WITHOUT WALL REFLECTION

4.4.1 Neutral Gas Density as the Solution to an Integral Equation

The total neutral gas density, N_T, will be taken as the solution to the integral equation

$$N_T(X) = N_o(X) + A\int_{-1}^{1} dz\,\frac{\tilde{N}_i(z)}{\sqrt{\ddot{T}_i(z)}}\,N_T(z)G(q(z,x)) \qquad (1)$$

where N_i is the proton density, N_e the electron density, T_i the ion temperature, and T_e the electron temperature, and where

$$\tilde{N}_i(z) = N_i(z)/N_i(0) \quad ,$$

$$\tilde{T}_i(z) = T_i(z)/T_i(0) \quad ,$$

$$q(z,x) = \frac{B}{\sqrt{\tilde{T}_i(z)}} \left| \int_z^x dz' \; \mathrm{EATN}(z') \right| \quad ,$$

$$G(q) = \int_0^\infty dy \; ye^{-y^2} E_1(q/y) \quad ,$$

$$N_o(x) = .5(E_2(\xi_1) + E_2(\xi_2)) \quad ,$$

$$\xi_1 = C \int_{-1}^{x} dz' \; \mathrm{EATN}(z') \quad ,$$

$$\xi_2 = C \int_x^1 dz' \; \mathrm{EATN}(z') \quad , \qquad (2)$$

$$\mathrm{EATN}(x) = \frac{N_i(x)\sigma_{cx}(x)\;\sqrt{T_i(x)} + N_e(x)\sigma_{EL}(x)\;\sqrt{1836.15152T_e(x)}}{N_i(0)\sigma_{cx}(0)\;\sqrt{T_i(0)}} \quad ,$$

$$A = 1.13(AM)\sigma_{cx}(0)N_i(0) \quad ,$$

$$B = A \quad ,$$

$$C = A\sqrt{T_i(0)/E\emptyset} \quad .$$

Moreover, σ_{cx} is the cross section for charge exchange between atomic neutral hydrogen and protons; σ_{EL} is the cross section for ionization of atomic neutral hydrogen by electron impact; AM is the minor radius of the torus; and E∅ is the external gas energy.

The computational procedure actually approximates the hot neutral density N_1. N_1 is equal to $N_T - N_o$ and N_1 satisfies the integral equation

$$N_1(x) = A \int_{-1}^{1} dz \frac{\tilde{N}_i(z)}{\sqrt{\tilde{T}_i(z)}} [N_o(z) + N_1(z)]G[q(z,x)] \quad . \qquad (3)$$

4.4.2 Computational Procedure for the Determination of an Approximation to the Hot Neutral Density, N_1

For x in [0,1]

$$\gamma(x) \equiv A \int_{-1}^{1} dz \frac{\tilde{N}_i(z)}{\sqrt{\tilde{T}_i(z)}} G(q(z,x)) \quad . \tag{4}$$

If

$$\bar{\gamma} \equiv \max\{\gamma(x)\} < 1 \quad , \tag{5}$$

then the method of functional iteration will yield a sequence of functions $\{N_1{}^{(i)}\}$ that converges to the function N_1. In fact, let $N_1{}^{(0)}$ be an arbitrary continuous function. Then for x in [0,1] and for i = 1,2, ..., define

$$N_1{}^{(i+1)}(x) \equiv \hat{I}(N_1{}^{(i)},x) \equiv A \int_{-1}^{1} dz \frac{\tilde{N}_i(z)}{\sqrt{\tilde{T}_i(z)}} [N_o(z) + N_1{}^{(i)}(z)] G(q(z,x)) \quad . \tag{6}$$

Then

$$\max_x \{| N_1(x) - N_1{}^{(i)}(x)| \} \leqslant \left[\frac{\bar{\gamma}}{1-\bar{\gamma}}\right] \max_x \{| N_1{}^{(i)}(x) - N_1^{(i-1)}(x)| \}$$
$$\leqslant \left[\frac{\bar{\gamma}}{1-\bar{\gamma}}\right] \max_x \{| N_1{}^{(1)}(x) - N_1{}^{(0)}(x)| \} \quad . \tag{7}$$

If $\bar{\gamma} \geqslant 1$, then the method of functional iteration is, in general, inappropriate. This does not necessarily mean that the integral equation has no solution.

The discrete analogue of the continuous problem expressed by Eq. (3) employs a finite partition of the unit interval [0,1]. Let

$$0 = x_1 < x_2 < \ldots < x_{N\emptyset X} = 1 \tag{8}$$

be the NØX nodes at which an approximation to the hot neutral density, $N_1(x_j)$, is desired. The approximation to $N_1(x_j)$ is taken to be an approximation to $\hat{N}_1(x_j)$ where

$$\hat{N}_1(x_j) = A \int_{-1}^{1} dz \, \frac{\tilde{N}_i(z)}{\sqrt{\tilde{T}_i(z)}} \, [N_o(z) + \hat{N}_1(z)] G[q(z,x_j)] \tag{9}$$

and $\hat{N}_1$ is a continuous, piecewise linear function on [0,1] with nodes at x_j. The definition of $\hat{N}_1$ is extended to [-1,0] by the requirement that $\hat{N}_1$ be symmetric with respect to x=0.

The computation defines

$$\hat{N}_1^{(0)}(x_j) = 0 \qquad j = 1, \ldots, N\emptyset X \tag{10}$$

and

$$\hat{N}_1^{(i+1)}(x_j) = \hat{I}(\hat{N}_1^{(i)}, x_j) \qquad j = 1, \ldots, N\emptyset X \tag{11}$$

where $\hat{I}$ is defined in Eq. (6). The definition of $\hat{N}_1^{(i)}$ for x not equal to x_j is determined by the requirement that $\hat{N}_1^{(i)}$ be a continuous, piecewise linear function on [0,1] with nodes at x_j. The inequality given in Eq. (7) is valid for the discrete case if $\hat{N}_1$ is identified with N_1 and if

$$\hat{\gamma} = \max \{\gamma(x_j)\} \tag{12}$$

is identified with $\bar{\gamma}$. For fixed j the sequence $\{\hat{N}_1^{(i)}(x_j)\}$ converges monotonically from below to $\hat{N}_1(x_j)$ whenever $\hat{\gamma} < 1$.

It can be shown ($\hat{\gamma} < 1$) that

$$|N_1(x_j) - \hat{N}_1(x_j)| \leqslant \gamma\varepsilon\left(1 + \frac{\hat{\gamma}}{1-\gamma}\right) \tag{13}$$

where

$$\varepsilon = \max_{x} \{|N_1(x) - N_1^*(x)|\} \tag{14}$$

and N_1^* is the continuous, piecewise linear function with nodes at x_j that satisfies

$$N_1^*(x_j) = N_1(x_j) \quad . \tag{15}$$

4.4.3 Computational Description of the Approximation to $\hat{I}(g,x_j)$

The basic computational problem is to make an adequate approximation of the value of the integral representation for $\hat{I}(g,x_j)$ as expressed in Eq. (6) where g is a continuous, piecewise linear function on [0,1] with nodes $\{x_j\}$. The definition of g is extended to [-1,0] by the requirement of symmetry with respect to x = 0. A straightforward quadrature formula is not adequate since the function G in Eq. (2) possesses a logarithmic singularity at q = 0; G may be represented by

$$G(q) = -\frac{\ln q}{2} + GM(q) \tag{16}$$

where GM is continuous for $q \geqslant 0$. Let

$$U(z,x) = \left| \int_z^x dz' \; EATN(z') \right| \;. \tag{17}$$

Then G can be expressed by

$$G[q(z,x)] = -\frac{\ln U(z,x)}{2} + \left\{ 0.25 \ln\left(\frac{\tilde{T}_i(z)}{B2} \right) + GM[q(z,x)] \right\} \;. \tag{18}$$

The integrand in the integral representation for $\hat{I}(g,x_j)$ possesses a logarithmic singularity at $z = x_j$. The situation may be handled by partitioning the integration interval as follows:

$$\int_{-1}^{1} = \int_{-1}^{m^-(QC)} + \int_{m^-(QC)}^{n^-(QF)} + \int_{n^-(QF)}^{x_j} + \int_{x_j}^{n^+(QF)} + \int_{n^+(QF)}^{m^+(QC)} + \int_{m^+(QC)}^{1} \tag{19}$$

Let

$$-1 = z_1 < z_2 < \ldots < z_{N\emptyset Z} = 1 \tag{20}$$

be a uniform partition of the interval [-1,1] and let NØZ be odd. Moreover, let

$$0 \leqslant QF < QC \tag{21}$$

be two fixed constants. Then define for each x_j

$$m^-(QC) = \max\{z_i \text{ such that } z_i < x_j \text{ and } q(z_i,x_j) \geqslant QC\},$$

$$n^-(QF) = z \text{ such that } z < x_j \text{ and } q(z,x_j) = QF,$$

$$n^+(QF) = z \text{ such that } z > x_j \text{ and } q(x_j,z) = QF,$$

$$m^+(QC) = \min\{z_i \text{ such that } z_1 > x_j \text{ and } q(x_j,z_i) \geqslant QC\}. \tag{22}$$

Undefined intervals are determined and handled correctly. The first three subintervals and the last three are qualitatively the same. If an odd number of z_i nodes are in $[-1, m^-(QC)]$, then a Simpson's approximation to the integral over $[-1, m^-(QC)]$ is calculated with these nodes. If there are an even number of z_i nodes, then a trapezoidal approximation is made with the first two nodes and a Simpson's approximation on $[z_2, m^-(QC)]$. A Simpson's approximation to the integral over $[m(QC), n^-(QF)]$ is calculated using NSIMP uniformly distributed nodes. To handle the third integral on the right-hand side of Eq. (19) the integrand is decomposed into a sum. The derivative of U with respect to z is given by [see Eq. (17)]

$$\begin{aligned} U'(z,x) &= -\,\mathrm{EATN}(z) \quad , \quad z < x \\ U'(x,z) &= \mathrm{EATN}(z) \quad , \quad z > x \end{aligned} \tag{23}$$

Using this and the representation for G expressed in Eq. (18), the third integral may be written as

$$A \int_{n^-(QF)}^{x_j} dz \, \frac{\tilde{N}_i(z)}{\sqrt{\tilde{T}_i(z)}} \, [N_o(z) + g(z)] \left\{0.25 \ln \left(\frac{\tilde{T}(z)}{B2}\right) + GM[q(z,x_j)]\right\}$$

$$+ 0.5A \int_{n^-(QF)}^{x_j} d\{U(z,x_j)[\ln U(z,x_j)-1]\}h(z) \tag{24}$$

where

$$h(z) = \frac{N_o(z) + g(z)}{\sqrt{\tilde{T}_i(z)}} \; \frac{\tilde{N}_i(z)}{\mathrm{EATN}(z)} \; . \tag{25}$$

Let

$$n^{-}(QF) = y_1 < y_2 < \ldots < y_{NSMO} = x_j \qquad (26)$$

be a uniform partition of the interval $[n^{-}(QF), x_j]$ where NSMO is odd and $\geqslant 3$. Then the first integral in Eq. (24) is approximated by Simpson's rule with the nodes y_i; the second integral is approximated by

$$0.25A \sum_{k=1}^{NSMO-1} [h(y_k) + h(y_{k+1})][s_{k+1}(\ln s_{k+1}-1)-s_k(\ln s_k-1)] \qquad (27)$$

where

$$s_k = U(y_k, x_j) \quad . \qquad (28)$$

4.4.4 Required Input, Space Allocation, and Calling Sequence for Computer Program

With the input specified and with sufficient memory space in the arrays W1 and WSC, the calls

```
CALL NEUTIT
CALL NUKERN(WSC,NPRØFI)                                    (29)
```

will yield, if $\hat{\gamma} < 1$, an approximation to $\hat{N}_1(x_j)$. In fact

$$W1(IN1+J) = \hat{N}_1^{(i)}(x_j) \qquad (30)$$

and the FØRTRAN function TDN may be used to give the approximation

$$N_o(x) + \hat{N}_1^{(i)}(x) \qquad (31)$$

to $N_T(x)$ where $\hat{N}_1^{(i)}$ is the i-th continuous, piecewise linear function defined by Eq. (11) and x is in [-1,1]. The value of i is determined by the accuracy desired in the approximation; i is the least i for which

$$\max_{x_j} \left\{ | \hat{N}_1(x_j) - \hat{N}_1^{(i)}(x_j) | \right\} \leqslant \left[\frac{\hat{\gamma}}{1-\hat{\gamma}} \right] \max_{x_j} \left\{ | \hat{N}_1^{(i)}(x_j) - \hat{N}_1^{(i-1)}(x_j) | \right\} \leqslant \text{EPSN1} \qquad (32)$$

where EPSN1 is user specified. NUKERN sets NPRØFI to 0 if a new profile has been generated and to 1 if no new profile has been generated.

Prior to the CALL NEUTIT the values for the variables N, NØX, NØZ, NSIMP, NSMO, NUNIF, QF, QC, EPSN1, INTRPS, and IDEBUG must be specified. However, QF, QC, EPSN1, INTRPS, and IDEBUG may be changed subsequently without issuing another CALL NEUTIT. Moreover, subsequent N_T profiles may be generated without another CALL NEUTIT unless a value of N, NØX, NØZ, NSIMP, NSMO, or NUNIF has changed.

Let

$$y_L = (L-1)/(N-1) \qquad L=1, \ldots, N \quad .$$

Prior to the CALL NUKERN the following variables must be defined [see Eq. (2)]:

$$\begin{aligned} &\mathrm{W1(ININ+L)} = \tilde{N}_i(y_L), \\ &\mathrm{W1(ITIN+L)} = \tilde{T}_i(y_L), \qquad L+1, \ldots, N \\ &\mathrm{W1(IEATN+L)} = \mathrm{EATN}(y_L), \end{aligned} \tag{33}$$

and A, B, and C.

Except for WSC and NPRØFI all the FØRTRAN variables and arrays appear in the labeled commons listed in Table 10.

The array W1 must have a minimum of

$$5N + 4N\text{Ø}X + 3N\text{Ø}Z \tag{34}$$

elements in its definition. Moreover, the number of elements in the array W1 must be specified in the variable NWDW1 in common NUTRLP. It is convenient to dimension W1 and define NWDW1 in a BLØCK DATA subprogram. In this way the size of W1 can easily be set to the value in Eq. (34).

The first argument in the parameter list for NUKERN is a scratch array to be used by NUKERN. This array must have at least (QF > 0)

$$\mathrm{N\text{Ø}Z(N\text{Ø}Z + 10NSMO + 5) + NSIMP(8N\text{Ø}X + 1) + 4(N\text{Ø}Z + NSMO) - 2} \tag{35}$$

elements in its definition. NEUTIT places the required number of elements in WSC in MWDSC in common NUTRLP. The user should check that WSC has at least MWDSC elements prior to the call to NUKERN.

A fuller understanding of the input parameters may be gained by reviewing the previous section and Sect. 1.5.

4.5 SUMMARY OF FUNCTION OF EACH SUBROUTINE

The descriptions of the subroutines appearing in this section are grouped as much as possible according to the model with which they are associated. Subroutines used in common by two or more of the models are in the group which appears first and which includes the executive subroutines for this module.

SUBRØUTINE NEUTRA

A call to NTF returns to NEUTRA the estimate of the total neutral density in the local array XTR. This is then scaled by the Franck-Condon boundary density FCDEX and placed in the array ZN1 for export through a labeled common. In fact

FCDEX = (FCDEN-FCF)*EXP(-TIM/TPØ)+FCF

and

ZN1 = FCDEX*XTR

where TIM is the current time and FCDEN, FCF, and TPØ are user-specified parameters.

The call to TØCØMP returns the estimate for the neutral gas atom temperature profile T_o in the array TØ.

Called from: GETSET, MAIN

Subroutines called: NTF, TØCØMP, SPECT

Commons required: IMPURT, INDEX, IØNS, LBMIBT, NEUTRL, TEMP, TIME

Variables required: FCDEN, FCF, TPØ, TIM, N

Arrays required: DEN, DENP, TE, TI, TØ

Variables and arrays changed: (See Subroutine NTF)

XTR total neutral density. May not be altered if NUTE(1) = 2 or 3.

ZN1 — total neutral density multiplicatively scaled by the Franck-Condon boundary density.

FCDEX — = NFC = Franck-Condon boundary density

ØNIDAI — contains $N_i/\sqrt{T_i}$ where N_i and T_i were used in the calculation of the profile in XTR. Applies only when NUTE(1) = 2 or 3. See Eq. (2) in Sect. 4.4

SIG — temporary storage for NTF. Used only if NUTE(1) = 2 or 3.

SUBRØUTINE NTF(NT, TE, TI, NE, NI, ØNIDAI, NTNEW, WSC)

Subroutine NTF is structured to accommodate and/or drive four models for estimating the total neutral density. The switch NUTE(1) is used to select the model desired. Currently only two models are available. If NUTE(1) = 1, then the analytic model described in Sect. 4.3 is selected. If NUTE(1) = 2, then the slab model approximation (without wall reflection) described in Sect. 4.4 is selected.

Suppose NUTE(1) = 2. In order to conserve computing time, the following criterion has been included to determine whether a new neutral density is to be calculated. When a neutral density is determined, the values $N_i/\sqrt{T_i}$ are stored in the array ØNIDAI and the scalar $\sigma_{cx}(0)\sqrt{T_i(0)}$ in the variable C1. On successive calls the range of the relative error between the values C1*ØNIDAI(I) and the associated current values is determined. If the magnitude of the relative error for any I exceeds the user-specified parameter EPSNO, then a new neutral profile is requested. However, if NPRØFI = 1 (see Sect. 4.4.4), then the most recently generated neutral profile will be used. In the case where NPRØFI = 1 on the first call to NUKERN, the analytic model is used to generate a neutral profile.

The variables described in Eq. (33) are defined within NTF whenever NUTE(1) = 2.

Called from: NEUTRA

Subroutines called: NEUTIT, NUKERN, TDN, SIGMA, E2

Commons required: GEØM, INDEX, NUTRLP, PRNT, SLAB, TIME

Variables and arrays changed:

NT — neutral gas density to use with given profiles

ØNIDAI* = NI(I)/SQRT(TI(I)) for the most recently calculated neutral profile

NTNEW* = scratch array with N elements

WSC* = scratch array with at least the number of elements as given in Eq. (35).

C1* = $\sigma_{cx}(0)\sqrt{T_i(0)}$ for the most recently calculated neutral profile.

*Applicable only when NUTE(1) = 2.

SUBRØUTINE TØCØMP(TØ, TI)

The various models for estimating the neutral gas atom temperature T_o should be contained in subroutine TØCØMP or driven from it. Presently the only model is to take T_o to be one tenth of T_i where T_i is the ion temperature.

```
TØ(I) = .1*TI(I)      I = 1,...,N
```

Called from: NEUTRA

Commons required: INDEX

Array changed:

TØ — upon exit TØ contains the estimate for T_o at the usual grid points.

SUBRØUTINE SPECT

The subroutine must presently be user supplied since it compares simulated results with measured values and thus depends on the particular experimental apparatus and instrumentation used to obtain the measured values (see Sect. 4.2).

SUBRØUTINE NEUTIT

Subroutine NEUTIT defines the pointers that logically partition the arrays W1 in common NUTRLP and in WSC, the scratch array that will be passed to NUKERN. Prints most of the level 0 data and checks that

$$\text{NWDW1} \geqslant 5\text{N} + 4\text{NØX} + 3\text{NØZ}.$$

If this inequality is not satisfied, a message is printed and a CALL EXIT is issued. If the inequality is satisfied, then NEUTIO is called before returning.

Called from: NTF

Subroutines called: NEUTIO, possibly EXIT

Commons required: NUTRLP, NEUTPT, INTRPS

Variables required: Level 0 data, i.e., N, NØX, NØZ, NSIMP, NSMO, NUNIF, QF, QC, EPSN1, INTRPS, IDEBUG. If NUNIF $\neq 0$, values to use for x_j must be in W1(J). Level 0 data is stored in commons NUTRLP and INTRPS.

SUBRØUTINE NEUTIO

Subroutine NEUTIO basically defines arrays that expedite the calculation of $\hat{N}_1^{(i)}(z_j)$ where the z_j is defined in Eq. (20) and in arrays of Z subscripts to aid in the determination of $m^-(QC)$ and $m^+(QC)$ in Eq. (22).

Called from:	NEUTIT
Subroutines called:	EXIT (whenever X(1) ≠ 0 or X(NØX) ≠ 1)
Commons required:	NUTRLP
Variables required:	NØX, NØZ, NSIMP, NUNIF, QF, QC, N

Variables defined:	SIMPM1 = NSIMP-1, ANM1 = N-1, DSP = 1./ANM1,
	NØXM1 = NØX-1, NØXP1, NØZP1

Arrays defined:	Z, X, JL, JH, IJ, RAT

SUBRØUTINE NUKERN(WSC, NPRØFI)

Subroutine NUKERN first generates the tables

$$\mathrm{W1(IINIOX + I)} = \int_0^{y_I} dZ\ \mathrm{EATN(Z)} \quad ,$$

$$\mathrm{W1(IINIX1 + I)} = \int_{y_I}^{1} dZ\ \mathrm{EATN(Z)}$$

for I = 1, ..., N where $y_I = (I-1)/(N-1)$. These are used in NIINT. Then two calls to NEUTC are issued. The first returns the set of values given in Eq. (12). Then NUKERN computes the Lipschitz constant $\hat{\gamma}$ in Eq. (12). If $\hat{\gamma} < 1$ and the number of estimated functional iterations required to obtain the accuracy expressed by EPSN1 is less than MAXIT(-250), then a second call to NEUTC is issued to obtain the desired approximations to $\hat{N}_1(x_j)$. The user should note that if $\hat{\gamma} \geqslant 1$, the computational procedure may be inappropriate, even though the integral equation may have a solution. If a problem is encountered, the user should review Sect. 4.4.2 on the computational procedure for the determination of an approximation to the neutral density. Facilities to print information from NUKERN are provided through the parameter IDEBUG, discussed in Sect. 1.5.

Called from: NTF

Subroutines called: NEUTC, SSIMP, INTRPU, ALØG

Commons required: NUTRLP, NEUTPT, INTRPS

Variables required:

W1(ININ+I) = $\tilde{N}_i(y_I)$, the normalized ion density

W1(ITIN+I) = $\tilde{T}_i(y_I)$, the normalized ion temperature

W1(IEATN+I) = EATN(y_I), estimate of the total neutral gas attenuation coefficient at the radial point y_I.

A = $1.13*AM*\sigma_{cx}(0)*N_i(0)$

B = A

C = $A*\sqrt{T_i(0)/E\emptyset}$ and a scratch array, WSC, with at least MWDSC elements.

SUBRØUTINE NEUTC

If NSC (in common NUTRLP) = 1, then

$$\gamma(x_j) = A \int_{-1}^{1} dz \frac{\tilde{N}_i(z)}{\sqrt{\tilde{T}_i(z)}} G(q(z,x_j))$$

is approximated for j = 1, ..., NØX. If NSC = 0, then approximations to $\hat{N}_1(x_j)$ are obtained. See Sect. 4.4.2.

Called from: NUKERN

Subroutines called: G, GM, NO, NIDSRT, SRTT, NIINT, NIIEQL, NI, T, NIIEQR, N1INTP, SSIMP, ØDDEV, EATN, ALØG, AMAX1, ERRØR (do not expect ERRØR to be called)

Commons required: NUTRLP, INTRPS

FUNCTIØN TDN(X)

In function TDN(X),

$$\mathrm{TDN}(X) = N_o(X) + \hat{N}_1^{(i)}(X)$$

where the functional iteration [see Eq. (11)] performed in NEUTC terminated with the i-th iterate. It is assumed that X is in [-1, 1].

Called from:	NTF
Subroutines called:	NO, N1INTP
Commons required:	NUTRLP

FUNCTIØN NO(X)

If NSC $\neq$ 0, then NO = 1. This is required in the approximations to the function γ at the nodal points x_j.

If NSC = 0, then NO is the approximate value of the source term in the integral equation for the total neutral density;

$$NO = .5\ (E_2(XI1) + E_2(XI2))$$

where

$$XI1 = C \int_{-1}^{X} dz\ EATN(z)$$

and

$$XI2 = C \int_{X}^{1} dz\ EATN(z) \quad .$$

Double precision is used for the evaluation of

(1-X*DEXP(X)*E1(X))/DEXP(X)

as the representation for the second exponential integral E_2.

Called from:	NEUTC
Subroutines called:	INTRPU, DE1MØD, DEXP
Commons required:	NUTRLP

<u>SUBRØUTINE DE1MØD(X, ZEXPX, E1XM)</u>

This subroutine returns E1XM = X*EXP(X)*E_1(X).

<u>Called from</u>: NO

<u>Subroutines called</u>: DLØG

<u>Variables required</u>: X, ZEXPX = DEXP(X)

FUNCTIØN N1INTP(NIL, Z, X)

The coordinates

(X(J), N1L(J)) J=1,...,NØX

completely describe the continuous, piecewise linear function N1L. N1INTP returns N1L(Z). This is simply linear interpolation applied to a table of values.

Called from: NEUTC, TDN

Subroutines called: BINSEE

SUBRØUTINE BINSEE (L, Z, X)

X is the array of nodes in [0, 1] at which the hot neutral density, N_1, is to be approximated. These are assumed to increase in value as the index increases. BINSEE determines, by bisection, an L such that

$$X(L) \leqslant |Z| \leqslant X(L+1).$$

It is assumed that $|Z| \leqslant 1$.

Called from: N1INTP

Commons required: NUTRLP

FUNCTIØN E2(X)

Function E2 returns the value of the exponential integral of the second kind

$$E_2(x) = \int_1^{\infty} \frac{e^{-xt}}{t^2} \, dt \quad .$$

Called from: NTF

Subroutines called: E1

Variables required:

X — the argument of the function

Variable changed:

E2 — the value of the integral

FUNCTIØN E1(X)

Function E1 returns the value of the exponential integral of the first kind

$$E_1(x) = \int_x^\infty \frac{e^{-t}}{t} dt \quad .$$

This routine was obtained from the Argonne Code Center.[4] The routine assumes x > 0.

Called from: E2

Variables required:

X — the argument of the function, the lower limit of the integral

Variables changed:

E1 — the value of the integral

FUNCTIØN G(Q)

This function returns an estimate of $G(Q) = \int_0^\infty dy\, ye^{-y^2} E_1(Q/y)$. For $Q > 10$ the function G returns 0 as the estimate for G(Q). This may need to be modified. If there exists a generated $Q > 10$, then this is reported from subroutine NUKERN. There should be about four significant figures in G(Q) for $Q \leqslant 10$. The user should refer to the comment cards within FØRTRAN function G.

Called from: NEUTC

Subroutines called: GM, ALØG

FUNCTIØN GM(Q)

This function returns an estimate of $GM(Q) = G(Q) - \frac{\ln Q}{2}$. GM requires, but does not check, that Q satisfies $Q \leqslant 1$. An appraisal of the accuracy of the estimate is contained within the function as comment cards.

Called from: NEUTC, G

FUNCTIØN NI(Z)

Function NI(Z) estimates $\tilde{N}_i(Z)$ at the radial point Z. $\tilde{N}_i(Z)$, the normalized ion density at Z, is estimated according to the interpolation scheme selected by the switch INTRPS.

Called from:	NEUTC
Subroutine called:	INTRPU
Commons required:	NUTRLP

FUNCTIØN T(Z)

Function T(Z) estimates $\tilde{T}_i(Z)$ at the radial point Z. $\tilde{T}_i(Z)$, the normalized ion density at Z, is estimated according to the interpolation scheme selected by the switch INTRPS.

Called from: NEUTC

Subroutines called: INTRPU

Commons required: NUTRLP

FUNCTIØN NIDSRT(Z)

This function approximates $\tilde{N}_i(Z)/\sqrt{\tilde{T}_i(Z)}$ where $\tilde{N}_i$ and $\tilde{T}_i$ are the normalized ion density and temperature. If INTRPS = 1, 2, or 3, then linear, quadratic, or cubic interpolation is used to approximate $\tilde{N}_i(Z)$ and $\tilde{T}_i(Z)$.

Called from:	NEUTC
Functions called:	INTRPU, SQRT
Commons required:	NUTRLP
Variables changed:	NIDSRT

FUNCTIØN SRTT(Z)

Function SRTT(Z) estimates $\sqrt{\tilde{T}_i(Z)}$ at the radial point Z. $\tilde{T}_i(Z)$, the normalized ion temperature at Z, is estimated according to the interpolation scheme selected by the switch INTRPS.

Called from:	NEUTC
Subroutines called:	INTRPU, SQRT
Commons required:	NUTRLP

FUNCTIØN NIINT (Z,X)

This function estimates $\left| \int_{z}^{x} dz'\mathrm{EATN}(z') \right|$.

Called from: NEUTC, NIIEQL, NIIEQR

Subroutines called: INTRPU, ØDDEV, SSIMP

Commons required: INTRPU, ØDDEV, SSIMP

Variables required: Z, X, ANM1, DSP, (W1(IEATN+I), W1(IINIOX+I), W1(IINIX1+I), I=1,...,N)

SUBRØUTINE NIIEQL(ZM, BL, BH, VAL)

This subroutine returns in ZM an approximate solution of

$$\frac{B}{\sqrt{\tilde{T}_i(z)}} \int_{z}^{BH} dz' \; EATN(z') = VAL$$

that is known *a priori* to lie in the interval [BL, BH]. Uniqueness of solution is not guaranteed.

Called from: NEUTC, SRTT

Subroutines called: NIINT

Variables required: BL, BH, VAL

SUBRØUTINE NIIEQR(ZM, BL, BH, VAL)

This subroutine returns in ZM an approximate solution of

$$\frac{B}{\sqrt{\tilde{T}_i(z)}} \int_{BL}^{z} dz'EATN(z') = VAL$$

that is known *a priori* to lie in the interval [BL, BH]. Uniqueness of solution is not guaranteed.

Called from:	NEUTC
Subroutines called:	NIINT, SRTT
Variables required:	BL, BM, VAL

FUNCTIØN EATN(Z)

This function estimates the total neutral gas attenuation coefficient at the radial point Z according to the interpolation scheme selected by the switch INTRPS. The analytic expression for the EATN function may be found in Eq. (2).

Called from:	NEUTC
Subroutine called:	INTRPU
Commons required:	NUTRLP

SUBRØUTINE SSIMP(S, F, N, HD3)

Subroutine SSIMP returns in S the Simpson's approximation to $\int_{r_1}^{r_N} dtf(t)$ where $r_i = r_1 + (i-1)*h$, HD3 = h/3, F(I) = $f(r_i)$, and N is odd. If N < 3, then S = 0.

Called from: NUKERN, NEUTC, NIINT

FUNCTIØN ØDDEV(N)

Function ØDDEV returns value 0 if N is even and 1 if N is odd. ØDDEV is an integer-valued function.

Called from: NEUTC, NIINT

FUNCTIØN INTRPU(Y, X, Y, ANM1)

Y(I) is an approximation to the function Y evaluated at (I-1)/(N-1) where I = 1, ..., N and N is the number of spatial grid points. ANM1 = N-1. Function INTRPU approximates Y at the argument X by means of linear, quadratic, or cubic interpolation whenever INTRPS = 1, 2, or 3 respectively. It is assumed that $|X| \leqslant 1$.

Called from:	NUKERN, EATN, SRRT, NIDSRT, NO, NIINT
Commons required:	INTRPS

SUBRØUTINE ERRØR

Subroutine ERRØR calls for a dump of the contents of core memory containing the job which references it and terminates the job. An ASSEMBLER language version of the subroutine is available for users having access to the IBM 360/370 series computers.

Called from: NEUTC

REFERENCES

1. J. T. Hogan and J. F. Clarke, J. Nucl. Mat. 53: 1 (1974).

2. J. F. Clarke and D. J. Sigmar, Proc. 7th European Conf. on Controlled Fusion and Plasma Physics, p. 134 (1975).

3. L. M. Petrie and O. W. Herrman, private communication (February 1976).

4. Donald F. Jordan, *ANL Subroutine Library Documentation Vol. 2*, Sect. C. Polynomials and Special Functions, ANL C369S: DEØNE (January 1973).

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

5. DESCRIPTION OF THE IMPURITY MODULE

5.1 PURPOSE AND STRUCTURE OF MODULE

The impurity module calculates the properties of nonhydrogenic ions in the plasma. Subroutine IMPRTY directs the choice of models for stripping of the multicharged ions, for spatial diffusion according to the transport model selected, and for radiative loss. The module requires the choice of a simple description [IMP(3) = 1] or a detailed model [IMP(3) = 2], based on one of the options selected with the switch IMP(2), for carbon, oxygen, and a pseudo "high-Z" or iron impurity.

Impurity stripping is currently modeled using corona equilibrium tables [IMP(1) = 1] for carbon, oxygen, and iron if the detailed model choice IMP(3) = 2 is made. A rate equation approach is also possible through the use of the subroutine DYNAM [IMP(1) = 2] but is not currently implemented. A description of DYNAM is included in this section.

Diffusion is modeled by using formulae, by a numerical calculation using Pfirsch-Schlüter diffusion coefficients, or by a combination of these two approaches. If the formulae are selected, the code arbitrarily sets the overall carbon, oxygen and/or "high-Z" spatial densities by the functions FØT, FØX, FCT, AND FCARX where:

FØT gives the time evolution of the total number of oxygen ions in the discharge [FØT(0) = 1];

FCT does the same for carbon;

FØX(X) gives the spatial distribution of the total number of oxygen ions

$(2\int_0^1 dX*X*FØX(X) = 1)$; and

FCARX(X) does the same for carbon.

The detailed model options are as follows:

IMP(2) = 1, corona equilibrium tables and formulae for carbon and oxygen. A crude model is used for high-Z impurities.

IMP(2) = 2, corona equilibrium tables and formulae are used for carbon, oxygen, and high-Z (iron).

IMP(2) = 3, corona equilibrium tables and Pfirsch-Schlüter diffusion for carbon and oxygen. A crude model is used for high-Z impurities.

IMP(2) = 4, corona equilibrium tables and Pfirsch-Schlüter diffusion are used for carbon, oxygen, and high-Z (iron).

5.2 BASIC INPUT/OUTPUT

The basic input to this module consists of the total carbon, oxygen, and electron density profiles; the electron temperature profile; the profiles of the charge states of carbon; some timing and geometry values; edge density values for carbon and iron; and an array of switch values.

The basic output consists of a set of new total carbon and oxygen density profiles, the effective charge of the plasma and the high-Z impurity, power losses due to the impurities, and synchrotron-bremsstrahlung radiation. A plasma ion density profile is also generated.

5.3 SUMMARY OF MODELS AVAILABLE

The available models include use of corona equilibrium tables, solution of a diffusion equation using Pfirsch-Schlüter diffusion coefficients, and a rate equation calculation which is not currently referenced by the program logic. (To employ the dynamic calculation, the call to TABLE should be replaced by a call to DYNAM.)

5.4 BRIEF DESCRIPTION OF EACH SUBROUTINE IN THE IMPURITY MODULE

The subroutines described in this section will probably undergo more changes than those of any other module. Therefore, more attention has been given to defining some of the significant internal variables in the module. This should ease the task of making desired modifications.

SUBRØUTINE IMPRTY

Subroutine IMPRTY is the main driving program for the impurity module. Figure 7 shows which subroutines are called, their order of call, and the flow of the logic associated with the switches which determine the combination of models used. This subroutine does some of the basic bookkeeping and the calculations of the impurity profiles using the formulae.

Called from:	MAIN
Subroutines called:	FCT, FØT, FCARX, FØX, ZDIFFU, TABLE, HIGHZ, ZCALC, IMPRIN
Commons required:	GEØM, IMPURT, INDEX, IØNS, RADLØS, SURF, TEMP, TIME, XPØRT, ZRAD
Variables required:	See Table 11 and Appendix II.
Variables changed:	See Table 11 and Appendix II.

Significant internal variables:

ICLK — a "clock" setting used to determine the frequency of printout of the density and effective charge profiles for the iron impurity. The density profiles of the various charge states of carbon and oxygen are also printed.

IST — a "timing" variable which keeps track of the number of times the impurity module is called, and is compared with the value of ICLK. When IST = ICLK, the impurity profiles are printed and the value of IST is reset to zero.

ICR — a "zero cross-over" switch used to initialize the values of total carbon and oxygen impurities on the first call to IMPRTY only

IICCT — a counter switch setting used to program logic control for the frequency with which low-Z diffusion is computed

IICC — counter variable for frequency of low-Z diffusion computation

IIFET — a counter switch setting, used for program logic control, which determines how often high-Z (Fe) impurity diffusion is computed

IIFE — counter variable for frequency of high-Z diffusion computation

FEO — central density of iron impurity, cm^{-3}

HTC — elapsed time variable used for the simulation of the low-Z impurity diffusion

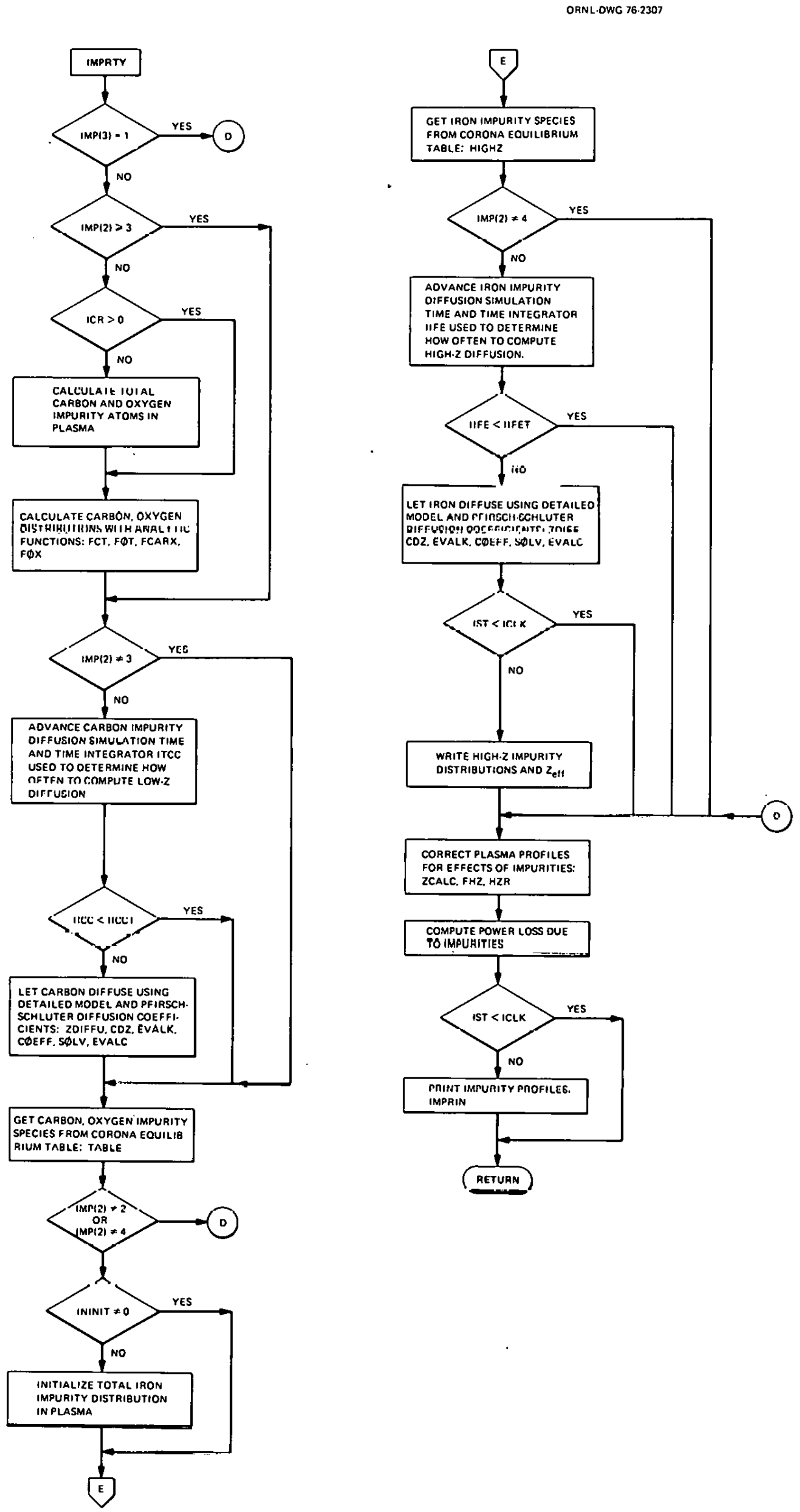

ORNL-DWG 76-2307
IMPRTY
IMP(3) = 1
YES
D
NO
IMP(2) ≥ 3
YES
NO
ICR > 0
YES
NO
CALCULATE TOTAL CARBON AND OXYGEN IMPURITY ATOMS IN PLASMA
CALCULATE CARBON, OXYGEN DISTRIBUTIONS WITH ANALYTIC FUNCTIONS: FCT, FØT, FCARX, FØX
IMP(2) ≠ 3
YES
NO
ADVANCE CARBON IMPURITY DIFFUSION SIMULATION TIME AND TIME INTEGRATOR ITCC USED TO DETERMINE HOW OFTEN TO COMPUTE LOW-Z DIFFUSION
ITCC < ITCCT
YES
NO
LET CARBON DIFFUSE USING DETAILED MODEL AND PFIRSCH-SCHLUTER DIFFUSION COEFFICIENTS: ZDIFFU, CDZ, EVALK, CØEFF, SØLV, EVALC
GET CARBON, OXYGEN IMPURITY SPECIES FROM CORONA EQUILIBRIUM TABLE: TABLE
IMP(2) ≠ 2 OR IMP(2) ≠ 4
D
ININIT ≠ 0
YES
NO
INITIALIZE TOTAL IRON IMPURITY DISTRIBUTION IN PLASMA
E
E
GET IRON IMPURITY SPECIES FROM CORONA EQUILIBRIUM TABLE: HIGHZ
IMP(2) ≠ 4
YES
NO
ADVANCE IRON IMPURITY DIFFUSION SIMULATION TIME AND TIME INTEGRATOR IIFE USED TO DETERMINE HOW OFTEN TO COMPUTE HIGH-Z DIFFUSION.
IIFE < IIFET
YES
NO
LET IRON DIFFUSE USING DETAILED MODEL AND PFIRSCH-SCHLUTER DIFFUSION COEFFICIENTS: ZDIFF, CDZ, EVALK, CØEFF, SØLV, EVALC
IST < ICLK
YES
NO
WRITE HIGH-Z IMPURITY DISTRIBUTIONS AND Z_{eff}
D
CORRECT PLASMA PROFILES FOR EFFECTS OF IMPURITIES: ZCALC, FHZ, HZR
COMPUTE POWER LOSS DUE TO IMPURITIES
IST < ICLK
YES
NO
PRINT IMPURITY PROFILES. IMPRIN
RETURN

Fig. 7. Diagram of IMPRTY module and subroutines called.

Table 11. List of labeled commons and elements in them which are referenced by subroutine IMPRTY

CØMMØN BLØCK	Input		Øutput	
	Variables	Arrays	Variables	Arrays
/GEØM/	VØL			
/IMPURT/	PCIMP	CARBØN, ØXYGEN, CC		CARBØN, ØXYGEN
/INDEX/	N, NM1	IMP		
/IØNS/		DEN		
/RADLØS/				A1
/SURF/	CARN, FENS		ISRØN	
/TEMP/		TE		
/TIME/	HT, TIM, TMAX			
/ZRAD/				QRD
/XPØRT/				D1
/ZRAD/				HIZ

HTFE — elapsed time variable used for the high-Z impurity diffusion simulation

ININIT — a "zero cross-over" switch used to initialize the total iron impurity profile and high-Z impurity (Fe) value of Z_{eff}

FUNCTIØN FCT(T)

Function FCT allows for programming the carbon impurity content of the plasma discharge throughout the simulation time. This subroutine currently sets the value of FCT to unity for all times.

Called from: IMPRTY

Variable required:

T — fraction of elapsed simulation time

Variable changed:

FCT — value of factor expressing time dependence of total carbon content of the plasma

FUNCTIØN FØT(T)

Function FØT allows for programming the oxygen impurity content of the plasma discharge throughout the simulation. The value of this function is currently set to unity for all times.

Called from: IMPRTY

Variable required:

T — fraction of elapsed simulation time

Variable changed:

FØT — value of factor expressing time dependence of total oxygen content of the plasma.

FUNCTIØN FCARX(X)

Function FCARX is used to program the profile of the total combined carbon species present at each point X in the plasma. The value of this function is currently set to unity for all radial values.

Called from: IMPRTY

Variable required:

X — radial position in the plasma, scaled to the torus minor radius

Variable changed:

FCARX — value of total carbon profile form factor at position X

FUNCTIØN FØX(X)

Function FØX is used to program the profile of the total combined oxygen species present at each point X in the plasma. The value of this function is currently set to unity for all radial values.

Called from: IMPRTY

Variable required:

X — radial position in the plasma, scaled to the torus minor radius

Variable changed:

FØX — value of total oxygen profile form factor at position X

SUBRØUTINE TABLE (TE,N)

Subroutine TABLE contains tables of values of the fractions of total density of the elements carbon and oxygen in each ionization state as a function of temperature. These fractions were computed on the basis of a corona equilibrium model of electron impact ionization. The fractions are given for temperature intervals of 10 eV and a linear interpolation is used to compute the fractions for an arbitrary value of the temperature.

Called from: IMPRTY

Commons required: IMPURT

Variables required: TE, N

Arrays changed:

CC(K,I) — density of the (K-1)th charge state of carbon at the i-th grid point

CØ(K,T) — density of the (K-1)th charge state of oxygen at the i-th grid point

Significant internal arrays:

C1 — table of fractions of total density of carbon in the ground state as a function of temperature

C2 — same as C1, except for first ionization state

C3 — same as C1, except for second ionization state

... ...

C7 — same as C1, except for sixth ionization state

Ø1 — table of fractions of total density of oxygen in the ground state as a function of temperature

Ø2 — same as Ø1, except for first ionization state

... ...

Ø9 — same as Ø1, except for eighth ionization state

SUBRØUTINE ZCALC (TE, D1, DEN, ZFE, ZIRØN, IIRØN)

Subroutine ZCALC computes the following profiles:

D1 $Z_{eff} \equiv \sum_k (n_k Z_k{}^2)/n_e$

DENP $n_p \equiv n_e - \sum_k n_k Z_k$

ZBR $[Z] \equiv \left[\sum_k (n_k Z_k{}^2)/A_k\right]/n_e$

QRD power loss due to line radiation from low-Z elements, eV/msec/cm^3

HIZ power loss due to line radiation from high-Z elements, eV/msec/cm^3

These profiles can be computed using a detailed model [IMP(3) = 1] or an analytic model [IMP(3) = 2].

In the detailed model calculation, the profiles are calculated by considering the impurity species

i) carbon — charge states 1 through 6,

ii) oxygen — charge states 1 through 8, and

iii) "high-Z" — stripped according to the formula Z = $\sqrt{\text{TE}/13.6}$ * ZIMP, where ZIMP is the atomic charge number.

The auxiliary values

$$\text{XØ1} = \sum_J \text{CØ}\ (\text{J,I}) * Z_J/16$$

$$\text{XC1} = \sum_J \text{CC}\ (\text{J,I}) * Z_J/12$$

$$\text{XCØ1} = \sum_J \text{CØ}\ (\text{J,I}) * Z_J/\text{DEN(I)}$$

$$\text{XCC1} = \sum_J \text{CC}\ (\text{J,I}) * Z_J/\text{DEN(I)}$$

$$\text{XCØ2} = \sum_J \text{CØ}\ (\text{J,I}) * Z_J{}^2/\text{DEN(I)}$$

$$\text{XCC2} = \sum_J \text{CC}\ (\text{J,I}) * Z_J{}^2/\text{DEN(I)}$$

are defined for each grid point I. The charge state for the "high-Z" element is modeled by assuming the charge to be

$$\text{ZI} = \begin{cases} 0 & \text{TE} < 13.6 \\ \sqrt{\text{TE}/13.6} & 13.6 < \text{TE} < 13.6*(\text{ZIMP})^2 \\ \text{ZIMP} & \text{TE} > 13.6*(\text{ZIMP})^2 \end{cases}$$

The "high-Z" impurity density is assumed to be

$$\mathrm{DENZ} = \mathrm{PCIMP} * .01 * \mathrm{DEN}(1) * \mathrm{DHZ}(X),$$

where DHZ is a chosen function of $X(=r/a)$.

Thus, for Z_{eff}

$$\mathrm{D1} = 1. + \mathrm{XCØ2} + \mathrm{XCC2} + \mathrm{DENZ}*(\mathrm{ZI})^2/\mathrm{DEN}(\mathrm{I}) \quad ,$$

for n_p

$$\mathrm{DENP} = 1. - \mathrm{XCØ1} - \mathrm{XCC1} - \mathrm{DENZ}*\mathrm{ZI}/\mathrm{DEN}(\mathrm{I}) \quad ,$$

for [Z]

$$\mathrm{ZBR} = 1. + \mathrm{XØ2} + \mathrm{XC2}$$

(the "high-Z" impurity is assumed to have infinite mass), for low-Z radiation power loss

$$\begin{aligned}\mathrm{QRD} = 9.18(10)^{-10} * \sqrt{13.6/\mathrm{TE}} * N_e * \Big[& N_2^O\, e^{-16/\mathrm{TE}} \\ & + N_3^O\, e^{-18/\mathrm{TE}} + N_4^O\, e^{-19/\mathrm{TE}} + N_5^O\, e^{-20/\mathrm{TE}} \\ & + N_6^O\, e^{-12/\mathrm{TE}} + 2N_7^O\, e^{-575/\mathrm{TE}} + 2N_8^O\, e^{-655/\mathrm{TE}} \\ & + N_8^O\, e^{-83/\mathrm{TE}} \Big] \\ & + 2.04(10)^{-10} \sqrt{13.6/\mathrm{TE}} * N_e * \Big[N_3^C\, e^{-12/\mathrm{TE}} \\ & + N_4^C\, e^{-8/\mathrm{TE}} + N_5^C\, e^{-308/\mathrm{TE}} + N_6^C\, e^{-370/\mathrm{TE}} \\ & + N_5^C\, e^{-40/\mathrm{TE}} \Big] \quad ,\end{aligned}$$

and for high-Z radiation power loss

$$\mathrm{HIZ} = \{.4 + 4.6*\mathrm{EXP}[-10^{-6}*(\mathrm{TE})^2]\} * \mathrm{DEN} * \mathrm{DENZ} * 4.17(10)^{-11}$$

For the analytic model, the profiles are given by simple analytic expressions:

Z_{eff}	D1 = ZEFF
n_p	DENP = DEN
[Z]	ZBR = 1
low-Z radiation power loss	QRD = 0.0
high-Z radiation power loss	HIZ = $4.17(10)^{18}$ * PHZ*HZR(X)/VØL,

where PHZ is the assumed power loss in kilowatts, specified in a DATA statement, and HZR is a preselected function of X(=r/a). [Note that

$$2\int_0^1 dX * HZR(X) * X = 1.]$$

Z_{eff} is represented by ZEFF, an input parameter read in at the beginning of the simulation.

Called from: IMPRTY

Common required: IMPURT

Variables and arrays required (see also Table 12):

TE — electron temperature profile, eV

D1 — profile of effective charge Z_{eff}

N — number of radial points in the spatial grid

DEN — electron density profile, cm^{-3}

IMP(3) — switch for type of model
= 1, detailed calculation
= 2, analytic model formulae

ZFE — effective charge profile of the iron impurity

ZIRØN — density profile of the iron impurity, cm ^{3}

IIRØN — switch used to determine whether high-Z effects are to be modeled using iron or a general "high-Z" impurity species

Arrays changed:

DENP — plasma proton density profile, cm^{-3}

ZBR — the profile of [Z]

D1 — effective charge Z_{eff} profile

QRD — low-Z impurity radiation contribution to the power loss, eV/(msec • cm^3)

HIZ — high-Z impurity radiation contribution to the power loss, eV/(msec • cm^3)

Table 12. List of labeled commons and elements in them which are referenced by subroutine ZCALC

CØMMØN BLØCK	Input		Output	
	Variables	Arrays	Variables	Arrays
/GEØM	VØL			
/INDEX/	N	IMP		
/IMPURT/	PCIMP, ZEFF, ZIMP	CØ, CC		DENP, ZBR
/ZRAD/		DHI, ZHI		QRD, HIZ

SUBRØUTINE ZDIFFU(VAR, SIV, HT)

Subroutine ZDIFFU advances the impurity atom population profile VAR(I) having charge distribution ZIV(I). ZDIFFU uses subroutine CDZ to compute the needed diffusion coefficients. The following diffusion equation was derived by Rutherford[1] for the Pfirsch-Schlüter regime.

$$\frac{\partial}{\partial t} \text{VAR}(r,t) = \frac{1}{r}\frac{\partial}{\partial r}\left[r{*}\text{TDZZ}{*}\ \frac{\partial}{\partial r}\text{VAR}(r,t)\right]$$

$$-\frac{1}{r}\frac{\partial}{\partial r}\left\{r{*}\text{VAR}(r,t){*}\text{ZIV}(r,t){*}\text{TDZZ}{*}\ \left(L_T^{-1}\left[C_1\left(1-\frac{1}{\text{ZIV}(r,t)}\right)\right.\right.\right.$$

$$\left.\left.\left.-C_2\right]+C_1L_P^{-1}\right)\right\}$$

where

$$L_T^{-1} = \frac{dT_i}{dr}\frac{1}{T_i}\quad,\ L_P^{-1} = \frac{dN_p}{dr}\frac{1}{N_p}\quad,$$

$$C_1 = 0.47 + \frac{0.35}{0.66+\alpha}\quad,$$

$$C_2 = 0.30 + \frac{0.41}{0.58+\alpha}\quad,$$

$$\alpha = \text{VAR}{*}(\text{ZIV})^2/N_p\quad,$$

TDZZ is computed in CDZ (Note: it is scaled so that $0 \leqslant r \leqslant 1$), and

$$\text{ZGRAD} = a\ L_T^{\ 1}\left[(1-\frac{1}{\text{ZIV}(r,t)}) - \frac{C_2}{C_1}\right] + a\ L_p^{-1}$$

is also computed in CDZ.

The numerical method for solving the diffusion equation is predictor-corrector and is contained in the subroutines EVALK, CØEFF, and SØLVE. These routines solve the general one-dimensional, time-dependent equation:

$$\frac{\partial u}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left[rk(u)\ \frac{\partial u}{\partial r}\right] - \frac{1}{r}\frac{\partial}{\partial r}\ [rf]$$

where f is a given inhomogeneous term.

Subroutine IMPRTY prepares the diffusion step by loading the appropriate profiles into the arrays VAR and ZIV and be selecting the value of the time step HT to be advanced (this may be different from the time step used for the plasma evolution in the main part of the simulation code). The differencing scheme in the immediately preceding equation is based on the Crank-Nicholson representation

$$\frac{u_i^N - u_i}{\Delta t} = \frac{1}{r}\frac{1}{2}\left\{\frac{[rk\,\frac{\partial u}{\partial r}]^N_{i+1/2} - [rk\,\frac{\partial u}{\partial r}]^N_{i-1/2}}{\Delta r} + \frac{[rk\,\frac{\partial u}{\partial r}]_{i+1/2} - [rk\,\frac{\partial u}{\partial r}]_{i-1/2}}{\Delta r}\right\} - \frac{1}{r}\frac{\tilde{f}_{i+1} - \tilde{f}_{i-1}}{2\Delta r}$$

where

$$\left.\frac{\partial u}{\partial r}\right|_{i+1/2} = \frac{u_{i+1} - u_i}{\Delta r} \;, \quad \left.\frac{\partial u}{\partial r}\right|_{i+1/2} = \frac{u_i - u_{i-1}}{\Delta r} \;, \text{ and } \tilde{f} = rf \;.$$

The steps used in the predictor-corrector cycle are:

1) predict — compute $u_i^{t + \frac{\Delta t}{2}}$ using $k[u_i(t)]$

2) compute $u_i^{t+\Delta t}$ using $k[u_i(t + \frac{\Delta t}{2})]$

3) correct — compute $u_i^{t + \frac{\Delta t}{2}}$ using

$$u_i^{t + \frac{\Delta t}{2}} = \frac{1}{2}\,[u_i^{old} + u_i^{t+\Delta t}]$$

and compare this with the prediction. If the prediction is within a specified tolerance, continue with the simulation. Otherwise

4) iterate the sequence

$$u_i^{(m)}\,(t + \tfrac{\Delta t}{2}) = \tfrac{1}{2}\,[u_i^{old}(t) + u_i^{(m)}(t + \Delta t)] \;.$$

Each new $u_i^{(m)}(t + \frac{\Delta t}{2})$ is gotten from a $u_i^{(m)}(t + \Delta t)$ in Step 2 using $k[u_i^{m-1}(t + \frac{\Delta t}{2})]$ until the tolerance criterion is satisfied.

When the iteration has converged (or after one correction step in the current code), the array VAR is set equal to the array X1, the value of the dependent variable at the new time.

Called from: IMPRTY

Subroutines called: EVALC, EVALK, CØEFF, SØLVE

Commons required: GEØM, INDEX, ZDFPS

Variables required:

VAR(101) — impurity species density profile, cm^{-3}

ZIV(101) — effective charge profile of impurity species

HT — time increment for the impurity diffusion, msec

From labeled common blocks (see Appendix II for definitions)

/GEØM/ — AM

/INDEX/ — N, NM1

/ZDFPS/ — TDZZ(101), ZGRAD(101)

Variables changed: VAR(101)

Significant internal variables:

X1(101) — a temporary (local) array for the impurity species density profile

ICY — subscript variable used to determine the number of corrector iteration cycles for the predictor-corrector method of solution

SUBRØUTINE EVALC (ZK,ZF,V,Z,AM,N)

Subroutine EVALC computes the profiles of the diffusion coefficient k(u) and the inhomogeneous term f of the diffusion equation referred to in the description of subroutine ZDIFFU (p. 124). These profiles must be recomputed for each new time in the diffusion process.

Called from:	ZDIFFU
Subroutines called:	CDZ
Common required:	ZDFPS

Variables required:

V(101) — total density of an impurity species of a given atomic number, cm^{-3}

Z(101) — effective charge profile of the given impurity species

AM — torus minor radius, cm

N — number of radial grid points

Variables changed:

ZK(101) — profile of k(u)

ZF(101) — profile of f

SUBRØUTINE CDZ(VAR,ZIV)

Subroutine CDZ computes the impurity ion diffusion coefficients for the Pfitsch-Schlüter regime. They are given by the expressions

$$\mathrm{TDZZ(I)} = \nu_i \, \rho^2 \left[\frac{r}{R_o} \frac{B_T}{B} \right]^2 c_1 \, (10)^{-3}$$

and

$$\mathrm{ZGRAD(I)} = \frac{a_m}{N_p} \frac{dN_p}{dr} + \frac{a_m}{T_i} \frac{dT_i}{dr} \left\{ [1. - \frac{1}{\mathrm{ZIV(I)}}] - \frac{c_2}{c_1} \right\}$$

where

$$\nu_i = \frac{1.13(10)^{-6}}{\sqrt{M}} N_e \left[1. - \frac{\ln(N_e/T_e)}{17.25} \right] Z_{eff} \, T_e^{-3/2} \quad ,$$

$$\rho^2 = 2.09(10)^4 \, MT_i/B_T^2 \quad ,$$

$$c_1 = \frac{0.47 + 0.35}{0.66 + \alpha} \quad , \quad c_2 = \frac{0.3 + 0.41}{0.58 + \alpha} \quad ,$$

and

$$\alpha = \mathrm{VAR(I)*ZIV(I)*ZIV(I)}/N_e \quad .$$

Subroutine CDZ prints the arrays VAR, ZIV, TDZZ, and ZGRAD the first time it is called.

Called from: EVALC

Commons required: FIELDS, GEØM, IMPURT, INDEX, IØNS, TEMP, XPØRT, ZDFPS

Variables required:

VAR(101) — total density of an impurity species of a given atomic number

ZIV(101) — effective charge profile of the given impurity species

From labeled common blocks (see Appendix II for definitions)

/FIELDS/ — B(101), BT

/GEØM/ — AM, HR, RØ

/IMPURT/ — DENP(101)

/INDEX/ — N

/IONS/ — DEN(101), AMU

/TEMP/ — TE(101), TI(101)

/XPØRT/ — D1(101)

Variables changed:

TDZZ(101) — impurity species diffusion coefficient

ZGRAD(101) — a collection of ion temperature and density gradient terms

Significant internal variables:

ZNUI — ion collision frequency

RHØ2 — ion gyroradius

SUBRØUTINE EVALK(ZK, N, ZKP, ZKM)

Subroutine EVALK calculates the arrays

$$\left.\begin{array}{l}\text{ZKP(I)} = .5*(\text{ZK(I)} + \text{ZK}(\text{I} + 1)) \\ \text{ZKM(I)} = .5*(\text{ZK(I)} + \text{ZK}(\text{I} - 1))\end{array}\right\} \quad \text{I} = 2, \ldots, \text{N} - 1$$

for the function k(u) in the predictor-corrector algorithm for obtaining the solution to

$$\frac{\partial u}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left[rk(u)\frac{\partial u}{\partial r}\right] - \frac{1}{r}\frac{\partial}{\partial r}[rf] \quad ,$$

set up in subroutine ZDIFFU.

Called from: ZDIFFU

Variables required:

ZK(I) — TDZZ(I)/AM/AM = $D(r)/a_m^2$, the impurity particle diffusion coefficient scaled by the torus minor radius

N — the number of radial nodes

Variables changed:

ZKP — forward difference of ZK

ZKM — backward difference of ZK

SUBRØUTINE CØEFF(ZKP, ZKM, A, B, C, D, E, F, S1, N, V, ZF)

Subroutine CØEFF computes the coefficients of the values of the impurity atom density u at the new time step in the set of linear difference equations which are obtained from the Crank-Nicholson representation of

$$\frac{\partial u}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\,[rk(u)\,\frac{\partial}{\partial r}] - \frac{1}{r}\frac{\partial}{\partial r}\,[rg]$$

where g(r) is a given inhomogeneous term. The Crank-Nicholson representation is

$$\frac{u_i^N - u_i}{\Delta t} = \frac{1}{r}\,\frac{1}{2}\,\{\delta^N + \delta\} - \frac{1}{r}\,\frac{f_{i+1} - f_{i-1}}{2\Delta r}$$

where

$$\delta^N = \frac{1}{\Delta r}\left\{[rk(u)\,\frac{\partial u}{\partial r}]^N_{i+1/2} - [rk(u)\,\frac{\partial u}{\partial r}]^N_{i-1/2}\right\} \quad ,$$

$$\delta = \frac{1}{\Delta r}\left\{[rk(u)\,\frac{\partial u}{\partial r}]_{i+1/2} - [rk(u)\,\frac{\partial u}{\partial r}]_{i-1/2}\right\} \quad ,$$

$$\left.\frac{\partial u}{\partial r}\right|_{i+1/2} = \frac{u_{i+1} - u_i}{r} \quad , \qquad \left.\frac{\partial u}{\partial r}\right|_{i-1/2} = \frac{u_i - u_{i-1}}{\Delta r} \quad , \quad f=rg$$

and

$$[rk(u)\,\frac{\partial u}{\partial r}]_{i+1/2} = \frac{1}{\Delta r}\,(r + \frac{\Delta r}{2})\,\frac{1}{2}\,(k_i + k_{i+1})(u_{i+1} - u_i) \quad .$$

The Crank-Nicholson representation can be rearranged to the form

$$-A_i\,u^N_{i+1} + B_i\,u^N_i - C_i\,u^N_{i-1} = D_i \quad .$$

Subroutine CØEFF also evaluates the arrays E and F which are used in the Gauss elimination procedure to solve for u:

$$E_i = \frac{A_i}{B_i - C_i E_{i-1}} \quad \text{and} \quad F_i = \frac{D_i + C_i F_{i-1}}{B_i - C_i E_{i-1}} .$$

Called from: ZDIFFU

Variables required:

ZKP(101) — forward difference of the diffusion coefficient, k(u)

ZKM(101) — backward difference of the diffusion coefficient, k(u)

S1 — reciprocal of the time step for the evolution of the impurity profile, msec^{-1}

N — the number of spatial nodes

U(101) — impurity species density profile at the current value of elapsed time, cm^{-3}

ZF(101) — profile of function appearing in the inhomogeneous term of the diffusion equation

Variables changed:

A(101)
B(101)
C(101) — coefficients appearing in the set of linear equations for the value of the density profile at the new time. The set of equations arises from the Crank-Nicholson finite differencing scheme for solving the diffusion equation.

D(101) — inhomogeneous term in the set of linear equations obtained from the finite differencing scheme

E(101)
F(101) — arrays used in the Gaussian elimination procedure to solve for the impurity species density profile at the new time step

SUBRØUTINE SØLVE(U, E, F, N)

Subroutine SØLVE generates the solution of the impurity atom diffusion equation for the new time value. It actually performs the Gaussian elimination using the E and F arrays generated by subroutine CØEFF. The routine sweeps

$$u_i = E_i u_{i+1} + F_i$$

from $r = a^-$ $(i = N-1)$ to $r = 0$ $(i = 1)$.

Called from: ZDIFFU

Variables required:

U(101) — impurity species density profile at the old time value (upon entry into this subroutine)

E(101), F(101) — coefficients computed in setting up the Gaussian elimination procedure

N — number of radial nodes in the spatial grid

Variables changed: U(101)

FUNCTIØN HZR(I,X)

Function HZR returns a value for the high-Z impurity radial profile form factor using the formula

HZR = (I + 1) * (1. - X*X)**I .

Called from: ZCALC

Variables required:

I — value of the exponent in the form factor formula

X — radial variable, normalized to the torus minor radius

Variables changed:

HZR — value of the form factor at X

Note: The user is free to choose any formula that seems appropriate for this form factor.

SUBRØUTINE HIGHZ (TE, ZIRØN, ZFE)

Subroutine HIGHZ contains tables of fractional values of total iron density in each ionization state as a function of temperature. These fractions were computed on the basis of a corona equilibrium model of electron impact ionization. The fractions are given for temperature intervals of 0.1 keV starting at 5.0 keV at the lowest energy considered and extending to 9.0 keV. Three separate cases are treated:

i) If T_e < TS(=5.0 keV), all the iron ions are in the lowest charge state (in the current version of this subroutine, the lowest charge state is 3+).

ii) If TS $\leqslant T_e \leqslant$ TTØP(=9.0 keV), a linear interpolation is used to compute the fractions for a given value of T_e.

iii) If TTØP < T_e, all the iron ions are in the highest charge state (in current version of the subroutine, the highest charge state is 26+).

A printout of the TE array and selected parts of the RFE array is given the first time subroutine HIGHZ is called.

Called from: IMPRTY

Common required: INDEX

Variables required:

TE(101) — electron temperature profile, eV

ZIRØN(101) — ion impurity (all charge states included) density profile

From labeled common block (see Appendix II for definitions)

/INDEX/ — N

Variables changed:

ZFE(101) — effective charge profile of iron impurities

Significant internal variables:

AFE(45,30) — array of exponents for the fraction of iron atoms in each charge state (K + 3) at a given temperature (index L)

RFE(101,30) — profiles of fractions of iron impurities in a particular charge state K + 3

SUBRØUTINE DYNAM (TE, DEN, DT, N)

Subroutine DYNAM calculates impurity stripping dynamically using electron impact ionization and recombination coefficients, determined analytically in a rate equation, to determine the densities of the various charge states of carbon and oxygen as a function of temperature. The ionization and recombination coefficients are supplied by subroutines SVCARB and SVØXYG.

The plasma simulation time interval is subdivided into ten intervals. The rate equation has the general form

$$\frac{dN_Z}{dt} = N_e \left\{ <\sigma v>^{Z-1}_{ion} N_{Z-1} + <\sigma v>^{Z}_{rec} N_{Z+1} - \left[<\sigma v>^{Z}_{ion} + <\sigma v>^{Z-1}_{rec} \right] N_Z \right\} .$$

This equation is integrated over the sub-intervals of time using the general expression

$$N_Z^{new} = \frac{N_Z^{old} + N_e \Delta t \left\{ <\sigma v>^{Z-1}_{ion} N^{old}_{Z-1} + <\sigma v>^{Z}_{rec} N^{old}_{Z+1} - \frac{1}{2} \left[<\sigma v>^{Z}_{ion} + <\sigma v>^{Z-1}_{rec} \right] N^{old}_Z \right.}{1 + \frac{1}{2} N_e \Delta t \left[<\sigma v>^{Z}_{ion} + <\sigma v>^{Z-1}_{rec} \right]}$$

to generate the density of each charge state at each step of the time integration.

The general expression above is modified appropriately for the neutral and completely ionized species. The neutral state cannot be fed from ion species below it by ionization since it is the lowest state and cannot feed a lower state by recombination. The completely ionized state cannot be fed from above by recombination nor can it feed anything higher by ionization. The integrations for these two states are thus done explicitly by themselves.

The total densities of carbon and oxygen impurities are computed from the densities of the various charge states of each element.

Called from:	Not currently used.
Subroutines called:	SVCARB, SVØXYG
Commons required:	IMPURT
Variables and arrays required:	TE, DEN, DT, N

Arrays changed:

CC(K,I) — density of the (K-1)th charge state of carbon at the I-th grid point

CØ(K,I) — density of the (K-1)th charge state of oxygen at the I-th grid point

CARBON(I) — total density of carbon impurity at the I-th grid point

ØXYGEN(I) — total density of oxygen impurity at the I-th grid point

SUBRØUTINE SVØXYG (TE, SVIØN, SVREC)

Subroutine SVØXYG generates ionization $<\sigma v>^{Z}_{ion}$ and recombination $<\sigma v>^{Z}_{rec}$ coefficients for the different charge states of oxygen. These coefficients are required by subroutine DYNAM for calculating the densities of the various oxygen impurity charge states. The coefficients are generated using the analytic expressions[2]

$$<\sigma v>^{Z}_{ion} = \begin{cases} A^{ion}_{Z} \; e^{-\beta} \dfrac{\sqrt{\beta}}{\beta + X^{ion}_{Z}} , & I_Z \leqslant 100\ T_e \\ 0 , & I_Z > 100\ T_e \end{cases}$$

and

$$<\sigma v>^{Z}_{rec} = \frac{A^{rec}_{Z} \; \beta^{3/2}}{\beta + X^{rec}_{X}} ,$$

where $\beta \; \frac{I_Z}{T_e}$. The values of the coefficients A^{ion}_{Z}, A^{rec}_{Z} and parameters X^{ion}_{Z} and X^{rec}_{Z} for each charge state are supplied in DATA statements in the subroutine. These values were obtained by R. A. Dory,[3] who made adjustments in the results given by Beigman et al.,[2] and Lotz.[4]

Called from: DYNAM

Variable required:

TE — value of the electron temperature, not to be confused with the electron temperature profile of the same name used in most of the remainder of the simulation code.

Arrays changed:

SVION — values of the ionization coefficient $<\sigma v>^{Z}_{ion}$ for each charge state Z

SVREC — values of the recombination coefficient $<\sigma v>^{Z}_{rec}$ for each charge state Z

SUBRØUTINE SVCARB (TE, SVIØN, SVREC)

Subroutine SVCARB generates ionization $<\sigma v>^{Z}_{ion}$ and recombination $<\sigma v>^{Z}_{rec}$ coefficients for the different charge states of carbon. These coefficients are required by subroutine DYNAM for calculating the densities of the various carbon impurity charge states. The coefficients are generated using the analytic expressions[2]

$$<\sigma v>^{Z}_{ion} = \begin{cases} A^{ion}_{Z} \, e^{-\beta} \dfrac{\sqrt{\beta}}{\beta + X^{ion}_{Z}} & , \; I_{Z} \leqslant 100 \; T_{e} \\ 0 & , \; I_{Z} > 100 \; T_{e} \end{cases}$$

and

$$<\sigma v>^{Z}_{rec} = \frac{A^{rec}_{Z} \; \beta^{3/2}}{\beta + X^{rec}_{Z}} \quad ,$$

where $\beta = \frac{I_{Z}}{T_{e}}$. The values of the coefficients A^{ion}_{Z}, Z^{rec}_{Z} and parameters X^{ion}_{Z} and X^{rec}_{Z} for each charge state are supplied in DATA statements in the subroutine. These values were obtained by R. A. Dory,[3] who made adjustments in the results given by Beigman et al.,[2] and Lotz.[4]

Called from: DYNAM

Variable required:

TE — value of the electron temperature, not to be confused with the electron temperature profile of the same name used in most of the remainder of the simulation code

Arrays changed:

SVIØN — values of the ionization coefficient $<\sigma v>^{Z}_{ion}$ for each charge state Z

SVREC — values of the recombination coefficient $<\sigma v>^{Z}_{rec}$ for each charge state Z

REFERENCES

1. P. H. Rutherford, *Phys. Fluids* 17: 1782 (1974).

2. I. L. Beigman, L. A. Vainshtein, and A. Vinogradov, Astron. Zh. [Sov. Astron. — AJ] 13: 775 (1970).

3. R. A. Dory, private communication.

4. W. Lotz, *Astrophys. J. Suppl. Ser. 128* 14: 207 (1967).

6. DESCRIPTION OF NEUTRAL BEAM INJECTION MODULE

6.1 PURPOSE, STRUCTURE, AND BASIC INPUT/OUTPUT OF MODULE

The neutral beam module treats the slowing down and charge-exchange interactions between the injected neutral beam and the plasma target. The module calculates the fast-ion deposition profile, the power transferred to electrons and ions in the plasma, and the change in the plasma current density. Heating and current perturbation profiles and smooth injection profiles are also obtained.

The fast-ion deposition profile $H(\xi = \frac{r}{a})$ is calculated when the simulation time (TIM) first exceeds the time of injection turn-on (TNJCT), as shown in Fig. 3. Subroutine SUBH calculates the profile at ten radial positions (ξ = 0, 0.1, ..., 0.9), and the results are stored in the array HTMP.

In order to assure a smooth beam turn-on, it is assumed that the beam has already slowed for a given time DTCGL in the simulated plasma existing at time TNJCT. This assumption is used in calculating the fractions of beam power transferred to the ions (G_i) and to the electrons (G_e) in the plasma, and in determining the change in the plasma current density (ΔJ). These values are obtained at ten radial positions by a call to subroutine FPMØME and are stored in the arrays GETMP, GITMP, and DJTMP.

Heating and current perturbation profiles are then obtained at the ten radial positions for the time TNJCT + DTCCL by using

$$Q_{e,i} = \sum_{\text{beam components, } J} \frac{I_B E_B(J)\ H_J(r)\ G_{e,i}(J)}{e \times \text{volume}}$$

and

$$\Delta J = \sum_{\text{beam components, } J} \Delta J(J) \quad .$$

The profiles are stored in arrays QETMP, QITMP, and DJTTMP. At time TNJCT, the injection profiles are all assumed to be zero.

Smooth injection profiles are then obtained over the entire number N of spatial grid points used in the simulation code by a fourth order spatial interpolation between the eleven calculated values for the points $r = 0.0, 0.1, ..., 0.9, 1.0$, where the heating and current perturbation are assumed to be zero. These smooth profiles, calculated at the times t_1 = TNJCT and t_2 = TNJCT + DTNJCT, are stored in QES, QIS, and DJS. As the simulation progresses from t_1 to t_2, the heating (QE1 and QI1) and current perturbation (DJT) profiles for each time are assumed to be obtained from a linear interpolation at each radial point between the values at t_1 and t_2.

When the simulation time $t = t_2$, subroutine FPMØME is called to give values of G_e, G_i, etc. for t_3 = TNJCT + 2*DTCGL, and a new set of profiles is generated for this time. The temporal interpolation is now between the old values of the injection profiles calculated for t_2 and the values just calculated at t_3 as the simulation proceeds from t_2 to t_3. This procedure repeats every DTCGL milliseconds until the end of the simulation. Every DTCH milliseconds, starting at t = TNJCT, subroutine SUBH is called to generate a new fast-ion deposition profile $H(\xi)$ which is used for each injection profile until the next call to SUBH. Figure 8 shows the order of subroutine calls in this module.

6.2 SUMMARY OF FUNCTION OF EACH SUBROUTINE

BLØCK DATA (FOR THE NEUTRAL BEAM INJECTION MODULE)

The block data subprogram for the neutral beam injection module contains initial values for the loss cone correction arrays. The array names and subscripts have the general form STX (I, J, L, M), where the values of X, I, J, L, and M are given in Table 13 with a brief description of each case represented by the values of these characters. The numbers contained in the loss cone correction arrays were generated using a Fokker-Planck code.[1]

ORNL-DWG 76-2310

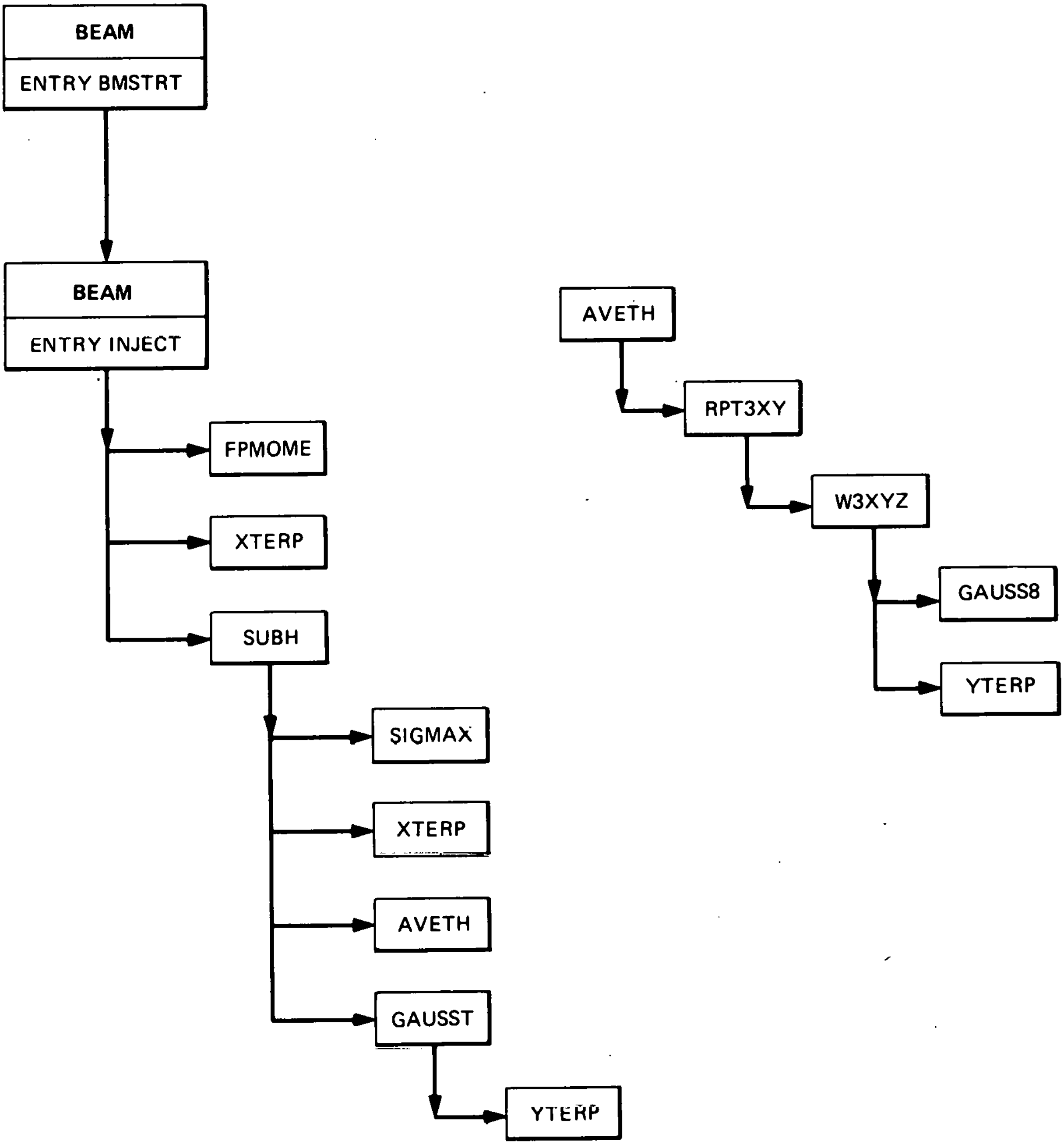


Fig. 8. Order of primary subroutine calls in the neutral beam injection module.

Table 13. Identification of characters used to label loss cone correction arrays STX(I, J, L, M)

X[a]	K	N	Description
1	1	1	G_e, no loss cone correction
2	1	2	G_i, no loss cone correction
3	1	3	KE, no loss cone correction
4	2	1	G_e, with loss cone correction
5	2	2	G_i, with loss cone correction
6	2	3	KE, with loss cone correction

Subscript[a]	Range of Values	Identification
I	1, 10	Ten values of the radius (normalized): 0.0, 0.1, ..., 0.9
J	1, 4	Energy components of injected neutral beam: E_o, $E_o/2$, $E_o/3$, and the sum of all three
K	1, 2	Loss cone corrections: without, with
L	1, 2	Injection direction: co-, counter-
M	1, 2	Impurity type: low-, high-Z
N	1, 2	Power component: G_e, G_i, KE

[a] Data is stored in the arrays in BLØCK DATA as STX(I,J,L,M)

SUBRØUTINE BEAM

Subroutine BEAM contains the primary logical control for the neutral beam injection module. Figure 9 shows this primary logical control and indicates where some of the computations described in Sect. 6.1 are done. Switch variables are also listed with the options which may be chosen according to the values assigned to these variables. The subscripts II, J, and L used in the output arrays refer to the radial position, energy component, and type of neutral beam injection, whether co-injection (L = 1) or counter-injection (L = 2).

In the alternate ENTRY BMSTRT section of the subroutine, many of the variables used in the moments calculation of G_e, G_i, G_e^E, G_i^E, etc., are initialized. Variables and arrays used to calculate the fast-ion deposition profile are initialized next, followed by adjustments (if necessary) to the loss cone correction tables. Logic near the beginning of this alternate entry section routes control to the remaining subsections.

The labeled common blocks associated primarily with the moments calculation are INPUT and ØUTPUT. Commons FPPP and INTRP are used primarily in the H(r) calculation.

Alternate ENTRY INJECT contains the logic and the interpolation and numerical procedures for calculating the heating and current perturbation profiles from the results of the moments and H(r) calculations for each simulation time step.

Due to the relatively long time required to do the injection computations, the complete heating and current perturbation profiles are computed at time intervals much longer than the simulation code time step. Subroutine BEAM interpolates between calculated values of the injection profiles for spatial and time grids which are coarser than the grids required by the rest of the simulation code.

In the call to alternate entry BMSTRT, the switch setting INJ(1) = 2 can be used to initialize the injection parameters quickly, but with only approximate values. The profiles for the electron and ion neutral beam heat input are assumed to have a decreasing exponential shape. This feature is useful for debugging the program.

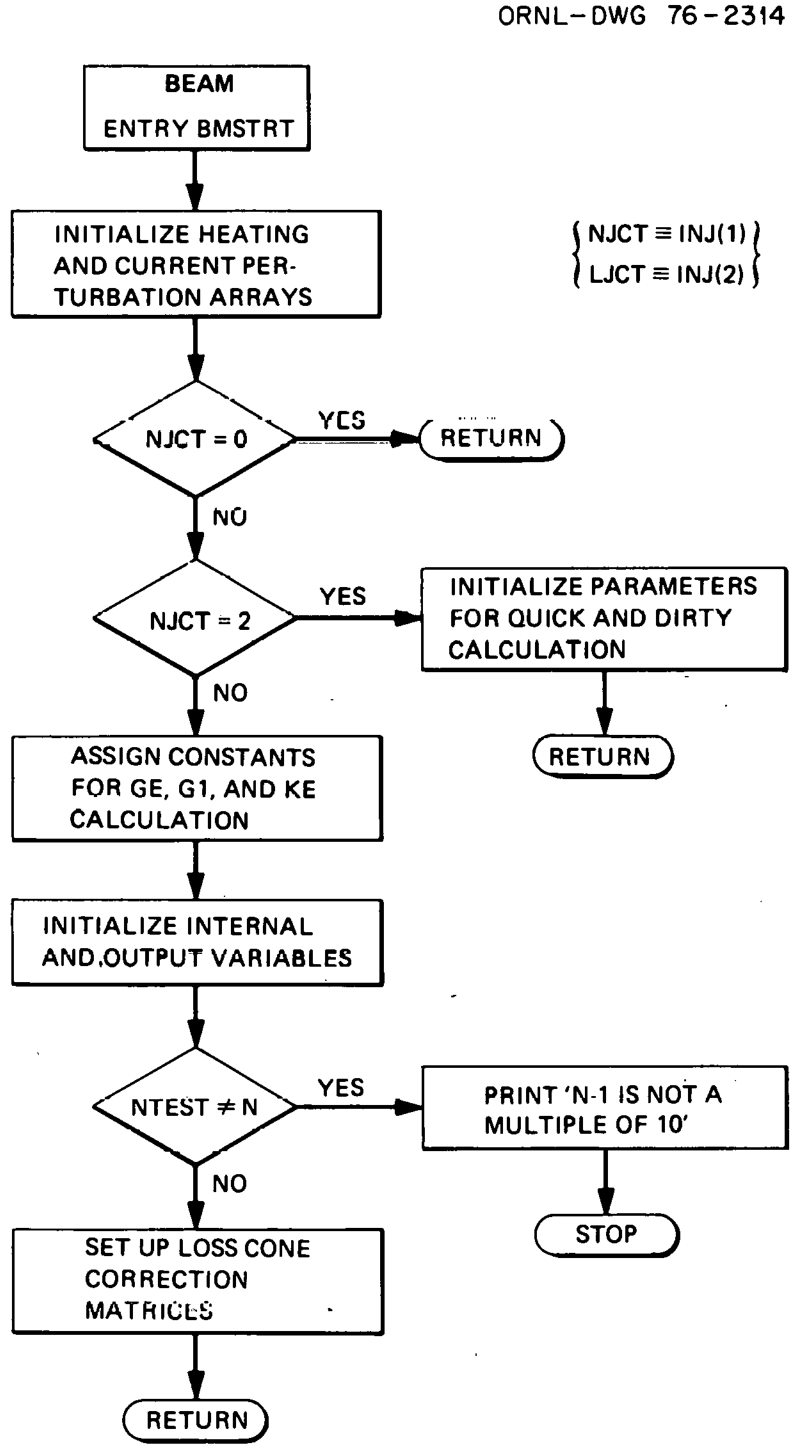

ORNL-DWG 76-2314
BEAM
ENTRY BMSTRT
INITIALIZE HEATING AND CURRENT PERTURBATION ARRAYS
NJCT ≡ INJ(1)
LJCT ≡ INJ(2)
NJCT = 0
YES
RETURN
NO
NJCT = 2
YES
INITIALIZE PARAMETERS FOR QUICK AND DIRTY CALCULATION
RETURN
NO
ASSIGN CONSTANTS FOR GE, G1, AND KE CALCULATION
INITIALIZE INTERNAL AND OUTPUT VARIABLES
NTEST ≠ N
YES
PRINT 'N-1 IS NOT A MULTIPLE OF 10'
STOP
NO
SET UP LOSS CONE CORRECTION MATRICES
RETURN

Fig. 9a. Primary logic for subroutine BEAM, ENTRY BMSTRT (initialization of arrays for neutral beam injection simulation).

ORNL-DWG 76-2315

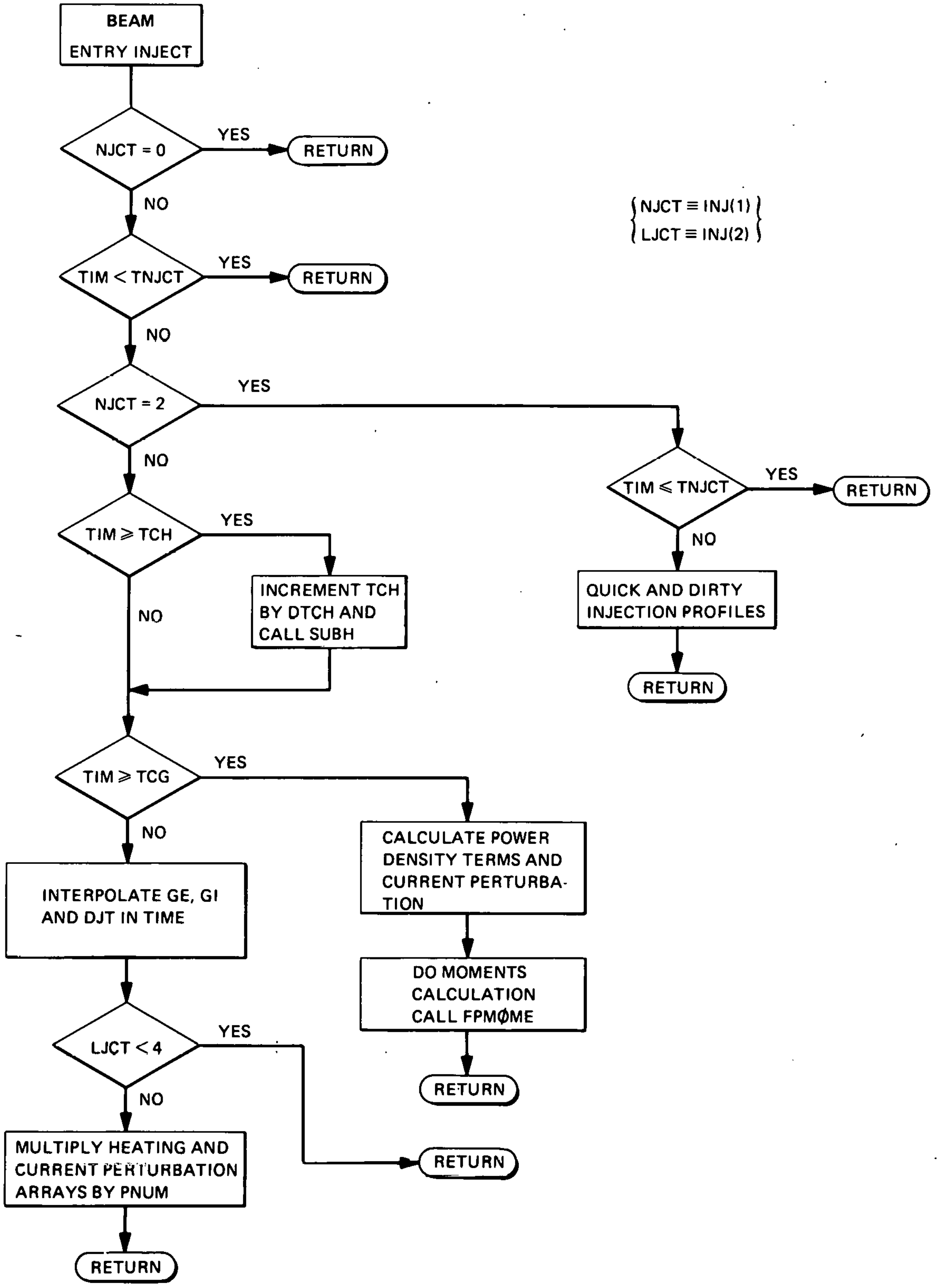

BEAM
ENTRY INJECT
NJCT = 0
YES
RETURN
NO
TIM < TNJCT
YES
RETURN
NO
NJCT ≡ INJ(1)
LJCT ≡ INJ(2)
NJCT = 2
YES
NO
TIM ≤ TNJCT
YES
RETURN
NO
QUICK AND DIRTY INJECTION PROFILES
RETURN
TIM ≥ TCH
YES
INCREMENT TCH BY DTCH AND CALL SUBH
NO
TIM ≥ TCG
YES
CALCULATE POWER DENSITY TERMS AND CURRENT PERTURBATION
DO MOMENTS CALCULATION CALL FPMØME
RETURN
NO
INTERPOLATE GE, GI AND DJT IN TIME
LJCT < 4
YES
RETURN
NO
MULTIPLY HEATING AND CURRENT PERTURBATION ARRAYS BY PNUM
RETURN

Fig. 9b. Primary logic for subroutine BEAM, ENTRY INJECT (main part of the calculation of the neutral beam injection module).

Called from:	GETSET (BMSTRT), MAIN (INJECT)
Subroutines called:	FPMØME, XTERP, SUBH
Commons required:	BEAMC, BEAMP, ELCTRN, FIELDS, FPPP, FUDGE, GEØM, IMPURT, INDEX, INPUT, INTRP, IØNS, NEUTRL, ØUTPUT, TEMP, TIME, TMINDX, XPØRT
Variables and arrays required:	See Table 14 and Appendix II
Variables and arrays changed:	See Table 14 and Appendix II

Table 14. List of labeled commons and elements in them used by subroutine BEAM

CØMMØN BLØCK	Input		Øutput	
	Variables	Arrays	Variables	Arrays
/BEAMC/	ENGY(1), DTCH, DTCGL, RC, TNJCT, NJCT, LJCT	CURBM	ENGY(2), ENGY(3)	DJT
/BEAMP/	TCGS(2)		TCGS(1), PNUM, TCHS, MM	HTMP, GETMP, GITMP, DJTMP, QETMP, QITMP, DJTTMP, SDT, ECRS, DJSUM, HX
/ELCTRN/	FQE			
/FIELDS/		E		
/FPPP/			NCGLM	
/FUDGE/		STL, ST2, ST3 ST4, ST5, ST6		ST4, ST5, ST6
/GEØM/	AM, RØ, VØL			
/IMPURT/		ZBR		
/INDEX/	N, NT	INJ		
/INPUT/			T, EØ, TEE, TII, DNE, DNO, AMF, AMI, BRAZ, SQBRZ, ZF, ETO, EPA3, EPA4, EPA5, M3, M4, M5, ICTX, INO	

Table 14 (continued)

CØMMØN BLØCK	Input		Øutput	
	Variables	Arrays	Variables	Arrays
/INTRP/	TØL	XXX		RTMP
/IONS/		DEN		
/NEUTRL/		ZN1		QE1, QI1
/OUTPUT/	ECØ, TSØ, TFØ, TCXVØ, VOØ, GII, GEE, BKE, GIE, GEEE, BKEE			
/TEMP/		TE, TI		
/TIME/	HT, TIM, TMAX			
/TMINDX/	DTNJCT, N1, N3			
/XPØRT/		D1		

SUBRØUTINE BMPWR

Subroutine BMPWR uses information calculated in the neutral beam injection module to generate profiles showing the power balance for the injected neutral beam after it has interacted with the target plasma. Table 15 shows the variables and arrays required as input and generated as output.

In all cases, $1 \leqslant J \leqslant 3$ refers to the energy component (E_o, $E_o/2$, or $E_o/3$) of the beam and L = 1, 2 to whether co- or counter-injection is used. The subroutine calculates total power by summing beam components and fractions of the total deposited power which go to electrons, ions, and the wall. The component and total current perturbations are also calculated. This information is printed by the subroutine when the printout for the rest of the simulation code is requested.

Called from:	MAIN
Subroutines called:	QUADR
Commons required:	BEAMC, BEAMP, FPPP, GEØM, INDEX, INTRP, NEUTRL, TIME, XTRA

Arrays changed:

FNBA — fraction of injected neutral beam deposited in the plasma (neutral beam absorption factor)

FPRT — fractions of total injected neutral beam power deposited in the plasma electrons and ions, lost to charge exchange, and deposited directly into the liner

DIT — sum of contributions to current perturbation resulting from co- and counter-injected neutral beams

DITT — sum of current perturbation contributions resulting from the three injected beam energy components

Variables representing amounts of power, W/cm^3, for each energy component and injection configuration.

QTB — supplied by the neutral beam injector

QTD — deposited in the plasma

QEB — deposited in the plasma electrons

QIB — deposited in the plasma ions

QCX — lost to charge exchange

QBL — deposited directly into the liner

Variables representing total power, W/cm^3, for the sum of the three injected beam energy components

QTBT — supplied by the neutral beam injector

QTDT — deposited in the plasma

QEET — deposited in the plasma electrons

QIIT — deposited in the plasma ions

QCXT — lost to charge exchange

QBLT — deposited directly into the liner

QINT — a working array containing intermediate quantities related to neutral beam power loss or transfer and current perturbation

SS — value of integral of profile contained in QINT array, computed by call to subrourine QUADR

Table 15. List of labeled commons and elements in them used by subroutine BMPWR

CØMMØN BLØCK	Input		Øutput	
	Variables	Arrays	Variables	Arrays
/BEAMC/	NJCT, LJCT	CUREM, ENGY		
/BEAMP/	PNUM, TCHS[a], MM[a]	HTMP, GETMP, GITMP, DJTMP, DJTIMP,[a] TCGS,[a] SDT,[a] ECRS,[a] DJSUM,[a] XPRT[a]		FNBA, QEB, QIB, QCX, QTB, QTD, QBL, QTBT, QTDT, QEET, QIIT, QCXT, QBLT, DIT, DITT, FPRT
/FPPP/		FPCX,[a] FPI,[a] FPE,[a] FP[a]		
/GEØM /	AM			
/INDEX/	N			
/INTRP/		RTMP		
/NEUTRL/		QE1,[a] QI1[a]		
/TIME/	TIM[a]			
/XTRA/		TE12[a]		

[a] Variables and arrays used only for printout.

SUBRØUTINE QUADR(Z,N,H,S,IER)

Subroutine QUADR integrates a function by numerical quadrature using a basic five-point formula if there are at least five function values. The routine assumes that the values of the independent variable (function argument) are equally spaced. Since the five-point formula uses successive groups of five function values, there will be function values remaining at one end of the integration interval when $N \pmod{4} \neq 0$. The integration is completed by a 2-, 3-, or 4-point formula according to whether $N \pmod{4}$ is 1, 2, or 3.

Called from: BMPWR

Variables required:

Z(50) — array of values of function to be integrated

N — number of points for which function values exist

H — spacing between successive values of the independent variable

Variables changed:

S — value of the integral

IER — diagnostic and error variable

= 0, normal return

= 1, only a single value or no value of the function to be integrated was passed from the calling program

= 2, the interval between successive values of the independent variable vanishes

SUBRØUTINE SUBH

Subroutine SUBH, when it is first called, initializes the variables and arrays required by the subroutines which generate the fast-ion deposition profile H(r).[2] The values of these variables and arrays are stored in the common block areas CN and D. These initializations are skipped on subsequent calls.

For each neutral beam energy component this subroutine calculates the mean free paths of the injected ions due to charge exchange and ionization. It sets up the plasma density profile (normalized to the central density) used in the calculation of H(r) and calls the subroutines to compute

$$H(r) = \frac{a^2}{I_T \pi r_B^2} \int_0^{\pi} d\theta \int_0^{\min\left\{\begin{matrix}\rho \\ r_B\end{matrix}\right\}} \frac{dZ_B}{\sqrt{\rho^2 - Z_B{}^2}} \; [I^+ + I^-] \quad ,$$

where

$$I^+ = \int_{R_c - \sqrt{r_B{}^2 - Z_B{}^2}}^{\min\left\{\begin{matrix} R_o + x_s + \sqrt{\rho^2 - Z_B{}^2} \\ R_c + \sqrt{r_B{}^2 - Z_B{}^2}\end{matrix}\right\}} I_B(r) \left.\frac{dN_f}{dR}\right|_{R = R_o + x_s - \sqrt{\rho^2 - Z_B{}^2}} dR_B$$

$$I^- = \int_{R_c - \sqrt{r_B{}^2 - Z_B{}^2}}^{\min\left\{\begin{matrix} R_o + x_s - \sqrt{\rho^2 - Z_B{}^2} \\ R_c + \sqrt{r_B{}^2 - Z_B{}^2}\end{matrix}\right\}} I_B(r) \left.\frac{dN_f}{dR}\right|_{R = R_o + x_s - \sqrt{\rho^2 - Z_B{}^2}} dR_B$$

$$\frac{dN_f}{dR} = \frac{1}{\tilde{\lambda}_o} \frac{\tilde{n}(r)}{\tilde{n}(o)} \frac{R}{\sqrt{R^2 - R_B{}^2}} \exp\left\{\frac{1}{\tilde{\lambda}_o}\left[\int_{R_o + \sqrt{a^2 - Z_B{}^2}}^{R} + \int_R^{R_B} - \int_{R_B}^{R_o + \sqrt{a^2 + Z_B{}^2}}\right] \frac{\tilde{n}(r)}{\tilde{n}(o)} \frac{R' dR'}{\sqrt{R'^2 - R_B{}^2}}\right\},$$

$$\rho = \sqrt{r^2 + x_s{}^2 - 2rx_s\cos\theta} \leqslant a - x_s \quad ,$$

and

$$x_s = 229.581\ [1\text{-}2(L\text{-}1)]\ \frac{\sqrt{\mathrm{ENGY(J)}}}{R_o\mathrm{*ZJ(1)}} \quad , \quad \left\{ \begin{array}{l} L = 1,\ \text{co-injection} \\ L = 2,\ \text{counter-injection} \\ J = 1,\ 2,\ 3\ \text{for the three} \\ \quad\ \text{energy components of the} \\ \quad\ \text{injected neutral beam.} \end{array} \right.$$

Subroutine SUBH calculates the plasma density profile used in the expression for H(r) at the N grid points used by the rest of the simulation code. The subroutine then calls a routine XTERP to do a fourth order interpolation to generate a table of values for 101 equally spaced radial points. However, the spatial meshes of the integrations required to generate the profile H(r) and used in the simulation code are independent of each other.

The quantity $\frac{1}{\tilde{\lambda}_o}\ \frac{\tilde{n}(r)}{\tilde{n}(o)}$ used here is different in form and kind from that used by Colchin and Wooten.[2] Some additional physics, the effect of impurities, has been added by defining a cross section for impurities $\sigma_I = Z_I^2\ \sigma_\rho$ for each impurity species I. The cross section σ_ρ is defined in terms of the proton impact ionization reciprocal mean free path $<\sigma v> = \sigma_\rho v_o$, where v_o is the initial velocity of the injected neutral beam.

Using the definition for Z_{eff}, the collision frequency,

$$\nu_T = n_e<\sigma v>_e + n_i(<\sigma v>_\rho + <\sigma v>_{cx}) + \sum_I n_I Z_i{}^2<\sigma v>_\rho \quad ;$$

can be written as

$$\nu_T = v_o n_i(o)\sigma_{cx} \left[\frac{n_e(r)}{n_i(o)}\ \frac{<\sigma v>_e}{\sigma_{cx}v_o} + \frac{n_i(r)}{n_i(o)} + \frac{n_e(r)}{n_i(o)}\ \frac{\sigma_p Z_{eff}}{\sigma_{cx}} \right] \quad .$$

Letting

$$\lambda_{cx} = \frac{1}{n_i(o)\sigma_{cx}} \quad , \quad \lambda_i = \frac{1}{n_i(o)\sigma_\rho} \quad , \quad \text{and } \lambda_e = \frac{v_o}{n_e(o)<\sigma v>_e} \quad ,$$

then

$$\frac{1}{\lambda_T} = \frac{1}{\lambda_{cx}} \left\{ n_i(r) + n_e(r) \left(\frac{\lambda_{cx}}{\lambda_i} Z_{eff} + \frac{n_i(o)}{n_e(o)} \frac{\lambda_{cx}}{\lambda_e} \right) \right\} \frac{1}{n_e(o)}$$

and the absorption decrement can be written as

$$D(S) = -\int_{S_o}^{S} \frac{ds}{\lambda_T} = \frac{1}{\lambda_{cx}(o)} \int_{S_o}^{S} \frac{\lambda_{cx}(o)}{\lambda_T} ds \quad .$$

Thus $\tilde{\lambda}_o = \lambda_{cx}(o)$, $\tilde{n}(o) = n_e(o)$

and

$$\tilde{n}(r) = n_i(r) + n_e(r) \left(\frac{\lambda_{cx}}{\lambda_i} Z_{eff} + \frac{n_i(o)}{n_e(o)} \frac{\lambda_{cx}}{\lambda_e} \right) .$$

The stagnation distance x_s (assumed constant) is also calculated for each neutral beam energy component. Subroutine SUBH then calls subroutine AVETH to get the H(r) profile, which is symmetric about the stagnation point. Subroutine GAUSST averages $H(\rho,\theta)$ on a flux surface to give H(r). A separate H(r) profile is calculated for each of the three beam energy components for co- and/or counter-injection. Thus, as many as six H(r) profiles can be returned for each call to subroutine SUBH; these profiles are stored in the array HTMP.

Called from:	BEAM (alternate ENTRY INJECT)
Subroutines called:	SIGMAX, XTERP, AVETH, GAUSST
Commons required:	BEAMC, BEAMP, CN, D, E, FIELDS, FPPP, GEØM, IMPURT, INDEX, INTRP, IØNS, TA, TEMP, XPØRT
Variables and arrays required:	See Table 16 and Appendix II
Variables and arrays changed:	See Table 16 and Appendix II

Table 16. List of labeled commons and elements in them used by subroutine SUBH

CØMMØN BLØCK	Input		Øutput	
	Variables	Arrays	Variables	Arrays
/BEAMC/	R7, R8, HALF, LJCT	ENGY		
/BEAMP/				HTMP, XPRT
/CN/			S1, NP	S
/D/			A, XS, RB, RHØ RC, RD, ØFP, NZ	
/FIELDS/		ZJ		
/FPPP/	NCGLM			FPCX, FPI, FPE, FP, FPIMP
/GEØM/	AM, RØ			
/IMPURT/		DENP		
/INDEX/	N			
/INTRP/		XXX	TØL	PIDEN
/IØNS/		DEN		
/TA/		HTT	RHØMAX, DELH DELN, NGLP1	HRHØ, RV
/TEMP/		TE		
/XPØRT/		D1		

SUBRØUTINE AVETH (R, HR)

Subroutine AVETH generates the profile H[ρ(θ)] using a 12-point Gauss-Chebyshev quadrature formula:

$$H[\rho(\theta)] = \frac{\pi A^2}{9.8695 r_B^2} * \text{ANSZ} * \text{BSHAPE}$$

where

$$\text{BSHAPE} = \left(\frac{r_B}{\text{HALF}}\right)^2 \left\{1 - \exp\left[-\left(\frac{r_B}{\text{HALF}}\right)^2\right]\right\}^{-1}$$

and

$$\text{ANSZ} = \begin{cases} 0 \text{ , if } \rho \geqslant a - |\text{XSS}| \\ \displaystyle\int_0^{\min\left\{\begin{matrix}\rho \\ r_B\end{matrix}\right\}} \frac{dz_B}{\sqrt{\rho^2 - z_B^2}} \; I^{\pm} \text{ , if } 0 < \rho < a - |\text{XSS}| \\ \frac{\pi}{2} I^{\pm} \text{ , if } \rho = 0 \text{ .} \end{cases}$$

The integrals I^+ and I^- are computed in subroutine RPT3XY according to the value of the switch NZ (=1 and =2, respectively):

$$I^{\pm} = \int_{R_c - \sqrt{r_B^2 - z_B^2}}^{\min\left\{\begin{matrix} R_o + x_s \pm \sqrt{\rho^2 - z_B^2} \\ R_c + \sqrt{r_B^2 - z_B^2} \end{matrix}\right\}} I_B(\tilde{r}) \frac{dN_f}{dR} \Bigg|_{R=R_o+x_s\pm\sqrt{\rho^2-z_B^2}} dR_B$$

An explanation of the modifications required to get the integral over z_B into the form required for applying the Gauss-Chebyshev quadrature formula is given in the description of subroutine RPT3XY. The constraint $\rho \geqslant a - |\text{XSS}|$ takes into account the excluded volume shown in Fig. 10.

Called from:	SUBH
Subroutine called:	RPT3XY
Commons required:	CN, D, E

ORNL-DWG 76-2312

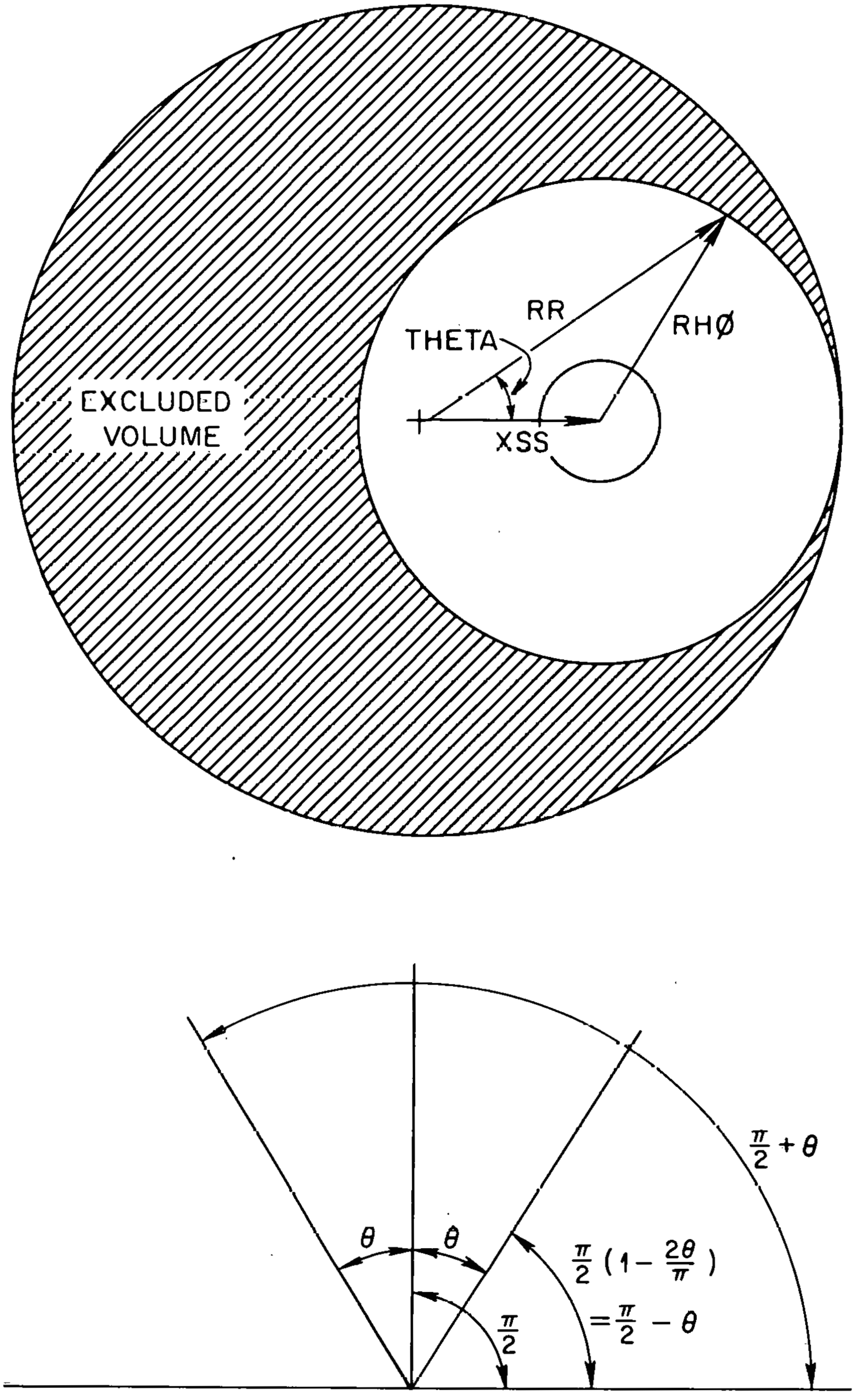


Fig. 10. Geometry and conventions used for computations in subroutines GAUSST, AVETH, and RPT3XY.

Variables required:

R — distance from the stagnation point to points in the plasma defined by $R \leqslant A - |XSS|$, where A is the torus minor radius

S — array of zeros of the Chebyshev polynominals of degree NP

S1 — Christoffel numbers β_j for the Gauss-Chebyshev formula, the coefficient π/n for the sum on the right-hand side (see description of subroutine RPT3XY)

NP — the number of points used by the quadrature formula

A — torus minor radius, cm

XSS — displacement between the magnetic axis and the stagnation point (see Fig. 10), cm

RB — radius r_B of the injected neutral beam, cm

ØFP — reciprocal of the absorption mean free path of the neutral beam, cm^{-1}

BSHAPE — injected neutral beam profile factor

Variables changed:

HR — the value of $H[\rho(\theta)]$ for a given value of $\rho(\theta) = R$

RHØ — the value of $\rho(\theta)$

Significant internal variables:

X — the successive values of the integration variable corresponding to the zeros of the Chebyshev polynominals of degree NP

UP — upper limit of integration

XL — lower limit of integration

RES1 — values of the integrals $I^{\pm}$

SUBRØUTINE RPT3XY(ZB, XINT3)

Subroutine RPT3XYZ uses a 12-point Gauss-Chebyshev quadrature formula to evaluate the integrals

$$I^{+} = \text{XINT3} = \int_{R_c - \sqrt{r_B{}^2 - z_B{}^2}}^{\min\left\{\begin{matrix} R_o + x_s + \sqrt{\rho^2 - z_B{}^2} \\ R_c + \sqrt{r_B{}^2 - z_B{}^2} \end{matrix}\right\}} I_B(\tilde{r}) \left.\frac{dN_f}{dR}\right|_{R = R_o + x_s + \sqrt{\rho^2 - z_B{}^2}} dR_B$$

when NZ = 1 and

$$I^{-} = \text{XINT3} = \int_{R_c - \sqrt{r_B{}^2 - z_B{}^2}}^{\min\left\{\begin{matrix} R_o + x_s - \sqrt{\rho^2 - z_B{}^2} \\ R_c + \sqrt{r_B{}^2 - z_B{}^2} \end{matrix}\right\}} I_B(\tilde{r}) \left.\frac{dN_f}{dR}\right|_{R = R_o + x_s - \sqrt{\rho^2 - z_B{}^2}} dR_B$$

when NZ = 2. The variable $\tilde{r}$ is the radial distance from the center of the injected neutral beam to the pencil beam axis. If

$$\min\left\{\begin{matrix} R_o + x_s \pm \sqrt{\rho^2 - z_B{}^2} \\ \\ R_c + \sqrt{r_B{}^2 - z_B{}^2} \end{matrix}\right\} \leqslant R_c - \sqrt{r_B{}^2 - z_B{}^2} \quad ,$$

then XINT3 = 0. This condition takes into account the cases when the injected neutral beam does not cross a particular flux surface and hence does not contribute any fast ions to that surface.

The integrals I^{+} and I^{-} are of the form $\int_b^a f(x)dx$ and must be modified to obtain the form required for the Gauss-Chebyshev quadrature in the following way:

$$\int_a^b w(x)\,\frac{f(x)}{w(x)}\,dx = \int_a^b \frac{F(x)}{\sqrt{1-x^2}}\,dx \quad ,$$

where $w(x) = \frac{1}{\sqrt{1-x^2}}$ is the weight function. The Gauss-Chebyshev formula is^3

$$\int_{-1}^{1} \frac{F(z)}{\sqrt{1-z^2}} \, dz \doteq \frac{\pi}{n} \sum_{j-1}^{n} F(z_j)$$

where

$$F(z_j) = \sqrt{1-z_j^2}\; f\left(\frac{a+b}{2} + z_j \frac{b-a}{2}\right) = \sqrt{1-z_j^2}\; f(\zeta_j) \quad .$$

The z_j are the zeros of the Chebyshev polynominals of degree n.

Called from: AVETH

Subroutines called: W3XYZ

Commons required: CN, D, E

Variables required:

ZB — distance z_B from torus plane of bilateral symmetry to axis of pencil beam, cm

S — array of zeros z_j of the Chebyshev polynominals of degree NP

S1 — Christoffel numbers β_j for the Gauss-Chebyshev formula, the coefficient π/n for the sum on the right-hand side

NP — the number of points used by the quadrature formula

XSS — displacement between the magnetic axis and the stagnation point (see Fig. 10), cm

RB — radius r_B ot the injected neutral beam, cm

RHØ — distance from the stagnation point to points in the plasma defined by RHØ $\leqslant$ AM - | XSS | , where AM is the torus minor radius

RC — radius of center of injected neutral beam measured from torus vertical axis, cm

RO — major radius of the torus, cm

NZ — switch used to determine which integral, I^+ or I^-, is to be evaluated

Variables changed:

XINT3 — the value of the integral I^+ or I^-

XUP1 — variable containing values of upper limits of the integrals, one of the values XUP1 = $R_o + x_s \pm \sqrt{\rho^2 - z_B^2}$ according to the value of NZ (=1, 2).

Significant internal variables:

X — array of successive values of ζ_j used to compute the value of the function $f(\zeta_i)$

FX — array of values of the integrands

$$I_B(\tilde{r})\,\frac{dN_f}{dR}\bigg|_{R=R_o+x_s\pm\sqrt{\rho^2-z_B{}^2}}$$

TXY — value of upper limit of integration

PXY — value of lower limit of integration

SUBRØUTINE W3XYZ(ZRHØ, ZB, ACRB, WXYZ)

Subroutine W3XYZ calculates the number of fast ions produced per unit radius dN_f/dR at the four points where the pencil beam passes through a given flux surface. The geometry upon which the ordering of the computations is based is shown in Fig. 11. The outer points p_1 and p_4 are treated separately from the inner points p_2 and p_3. The switch variable NZ distinguishes between these two cases. Values of dN_f/dR for points p_1 and p_4 are computed when NZ = 1 and for points p_2 and p_3 when NZ = 2.

The general equation upon which this subroutine is based is[2]

$$I_B(\tilde{r})\,\frac{dN_f}{dR} = I_B(\tilde{r})\,\frac{\tilde{n}(r)}{\tilde{n}(o)}\,\frac{R}{\sqrt{R^2 - R_B^2}}\left\{\exp\left[\frac{1}{\tilde{\lambda}_o}\int_{R_o + \sqrt{a^2-z_B^2}}^{R} + \int_{R}^{R_B} - \int_{R_B}^{R_o + \sqrt{a^2 - z_B^2}}\right]\frac{\tilde{n}(r)}{\tilde{n}(o)}\,\frac{R'dR'}{\sqrt{R'^2-R_B^2}}\right\}$$

where r is the radial distance from the torus minor axis to the pencil beam axis, $\tilde{r}$ the radial distance from the center of the injected neutral beam to the pencil beam axis and $\tilde{\lambda}_o = \tilde{\lambda}_{cx}(o)$ the charge-exchange mean free path (the beam absorption mean free path).

Called from:	RPT3XY
Subroutines called:	GAUSS8, YTERP
Commons required:	CN, D, E, HZ, INTRP

Variables required:

ARHØ — distance from the stagnation point x_s to points in the plasma, i.e., radius of a flux surface

ZB — vertical displacement z_B of pencil beam from plane of symmetry of the torus, cm

ACRB — an array of current values of R_B (there are nested integrals)

NP — number of points used by the Gauss-Chebyshev quadrature formula

A — torus minor radius, cm

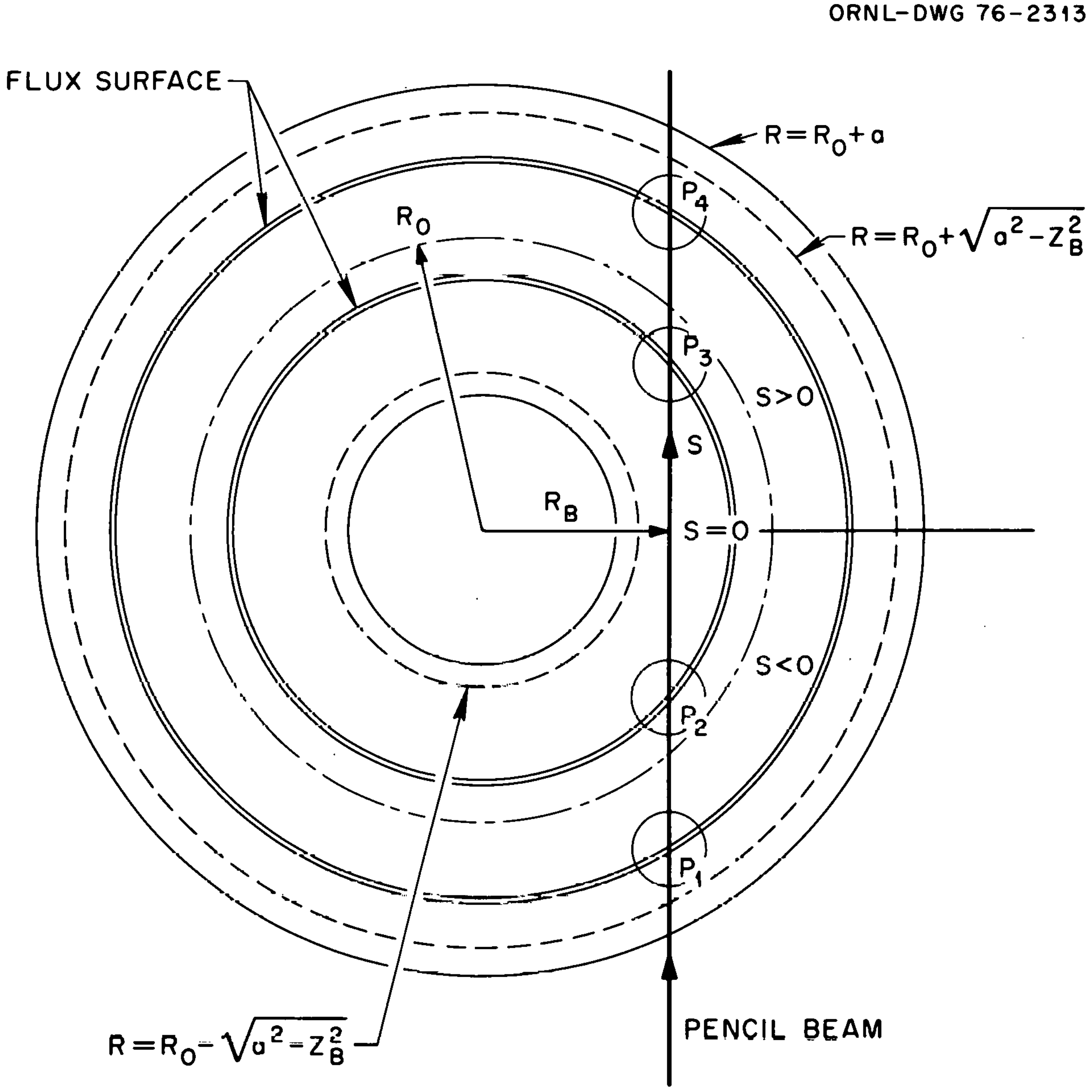


Fig. 11. Pencil beam geometry for neutral beam injection.

RC — radius of center of injected neutral beam measured from torus vertical axis, cm

RO — major radius of the torus, cm

NZ — switch used to determine at which pair of points to compute dN_f/dR
= 1, p_1 and p_4
= 2, p_2 and p_3

XUP1 — a variable containing one of the limits for the integrals, with one of the values XUP1 = $R_o + x_s \pm \sqrt{\rho^2 - z_B{}^2}$ according to the value of NZ (=1, 2)

BEAM — square of the half width of the assumed Gaussian profile shape of the injected neutral beam

DXX — spacing interval for the XXX array

XXX — independent variable array containing values of radius for which the plasma density is known

PIDEN — array containing plasma density profile normalized to the central density (see description of subroutine SUBH)

Variables changed:

WXYZ — array of values of the quantity $\left.\frac{dN_f}{dR}\right|_{R = R_o + x_s \pm \sqrt{\rho^2 - z_B{}^2}}$

CRB — current value of R_B, required by a nested integral

RHØ — value of radius of a particular flux surface, cm

XZB — a particular value of z_B, used in a nested integral, cm

Significant internal variables:

A1 — attenuation factor for beam from entrance point of plasma to p_1

A2 — attenuation factor for beam from entrance point of plasma to p_4

A3 — attenuation factor for beam from entrance point of plasma to p_2

A4 — attenuation factor for beam from entrance point of plasma to p_3

A7 — value of neutral beam current profile $I_B(\tilde{r})$ at the location of the pencil beam

XL, XL1 — values of lower limits of integration

XU, XUP1— values of upper limits of integration

SUBRØUTINE GAUSS8(A, B, ANS)

Subroutine GAUSS8 uses a 12-point Gauss-Chebyshev quadrature formula to evaluate integrals of the form

$$\mathrm{ANS} = \frac{1}{\tilde{\lambda}_o} \int_A^B \frac{\tilde{n}(r)R'dR'}{\sqrt{R'^2 - R_B^2}} \quad , \; r = R' - R_o$$

which appear in the general form of the absorption decrement[2]

$$D(s) = \int_{-\infty}^{S} \frac{ds'}{\lambda(s')}$$

for a pencil beam component of an injected neutral beam. Let $R_B = R'\xi$; then

$$\mathrm{ANS} = \frac{1}{\tilde{\lambda}_o} \int_{R_e/A}^{R_B/B} \tilde{n}\left(\frac{R_B}{\xi} - R_o\right)\left(-\frac{R_B}{\xi^2}\right)\frac{d\xi}{\sqrt{1-\xi^2}} \quad .$$

This integral is of the form $\int_a^b f(x)dx$ and must be modified as follows to obtain the form required for the Gauss-Chebyshev quadrature:

$$\int_a^b w(x)\,\frac{f(x)}{w(x)}\,dx = \int_a^b \frac{F(x)}{\sqrt{1-x^2}}\,dx \quad ,$$

where $w(x) = \frac{1}{\sqrt{1-x^2}}$ is the weight function. The Gauss-Chebyshev formula is[3]

$$\int_{-1}^{1} \frac{F(z)}{\sqrt{1-z^2}}\,dz \doteq \frac{\pi}{n} \sum_{j=1}^{n} F(z_j) \quad ,$$

where for the integral to be evaluated,

$$F(z_j) = \sqrt{1-z_j^2}\; f\left(\frac{a+b}{2} + z_j\,\frac{b-a}{2}\right) = \sqrt{1-z_j^2}\; f(\zeta_j)$$

with

$$f(\zeta_j) = \tilde{n}\left(\frac{R_B}{\zeta_j} - R_o\right)\left(-\frac{R_B}{\zeta_j}\right)\frac{1}{\sqrt{1-\zeta_j{}^2}} \quad .$$

The z_j are the zeros of the Chebyshev polynominals of degree n.

Values for the density (normalized to the plasma central density) $\tilde{n}(r)$ at the radii corresponding to the z_j are computed using a linear interpolation of the normalized density profile array.

Called from: W3XYZ

Commons required: CN, D, E, HZ, INTRP

Variables required:

A — lower limit of the integral

B — upper limit of the integral

CRB — current value of R_B (there are nested integrals), the distance from the major axis of the torus to the point of tangency of the pencil beam

NP — the number of points used by the quadrature formula

S — array of zeros z_j of the Chebyshev polynominals of degree NP

RO — torus major radius, cm

ZB — pencil beam vertical displacement z_B from plane of symmetry of the torus, cm

AB — torus minor radius, cm

DXX — spacing interval for independent variable array corresponding to the normalized density profile

PIDEN — plasma density profile array, normalized to the central density (see description of subroutine SUBH)

XXX — independent variable array corresponding to the normalized density profile

S1 — Christoffel numbers β_j for the Gauss-Chebyshev formula, the coefficient π/n for the sum on the right-hand side.

ØFP — reciprocal of the absorption mean free path of the neutral beam at r = 0, cm^{-1}

Variables changed:

ANS — the value of the integral

Significant internal variables:

X — the variable which takes on the successive values of ζ_j used to compute the value of the function $f(\zeta_j)$

SUBRØUTINE GAUSST

Subroutine GAUSST does the integration over θ for $H(r,\theta)$ to get the fast ion production profile $<H(r,\theta)>_\theta$ according to

$$H(r) = <H(r,\theta)>_\theta = \frac{1}{\pi} \int_0^\pi H[r(\theta)]d\theta \quad ,$$

using an eight point Gaussian quadrature. Figure 11 shows the geometry and conventions used as the basis for the computational procedure.

Called from: SUBH

Function called: YTERP

Commons required: D, TA

Variables required:

NGLP1 — number of computational grid elements plus one used for the $H[\rho(\theta)]$ profile

XSS — displacement between the magnetic axis and the stagnation point (see Fig. 10), cm

RHØMAX — maximum value of RHØ permitted, as determined by the position of the torus wall or diverter relative to the stagnation point

DELN — AM/NCGLM, radial increment with respect to torus minor radius AM, cm

DELH — RHØMAX/NCGLM, radial increment with respect to maximum value of RHØ, cm

RV — array of radial values at which the fast ion production profile, $H(r,\theta)$ before angle averaging, is determined, i.e., $0 \leqslant RV(I) < RHØMAX$

HRHØ — array of values of $H(r,\theta)$

Variables changed:

RHØ — distance from the stagnation point to points in the plasma defined by RHO $\leqslant$ AM - XSS, where AM is the torus minor radius

HTT — value of the integral $<H(r,\theta)>_\theta$ at each of the points in the interval [0,AM] with grid spacing DELN.

FUNCTIØN XTERP (XVAL, X, F, NØPTS, TØL, NCALL)

Function XTERP returns a value interpolated from the F-array corresponding to a value XVAL of the independent variable when XVAL lies in the range of the X-array values. The routine uses a Newton backward difference-forward difference method to get the interpolated value. The number of iterations is fixed at four since the F-array values do not necessarily lie along a smooth curve. High order polynomial interpolations to the F-array values generally cause numerical problems.

Called from: BEAM (alternate ENTRY INJECT), SUBH

Variables required:

XVAL — value of independent variable for which an interpolated value of the function (dependent variable) is desired

X — table of values of independent variable

F — table of values of dependent variable

NØPTS — number of points for which function values exist in the F-array

TØL — tolerance factor for test of error size

NCALL — call number of the subroutine, used only for printing out diagnostics

Variable changed:

XTERP — interpolated value of the dependent variable

Limitations:

The error test value is given by | TØL*F(1) |

FUNCTIØN YTERP(RRR, XXX, PIDEN, DXX, NN)

Function YTERP returns the value of a tabulated function at some specified point using a linear interpolation procedure. The routine assumes the values in the independent variable array are equally spaced. Let $f_i = f(x_i)$ and $\Delta = x_{i+1} - x_i$; then the interpolation formula is

$$f(x) = f_i + [f_{i+1} - f_i](x-x_i)/\Delta \quad .$$

If a value of x outside the range of the tables of values for f_i and x_i is supplied, the value of the function returned is an extrapolation based on the two end values of the tables appropriate for the cases when $x < x_1$ and when $x > x_{NN}$. The tables are assumed to be ordered according to increasing values of x_i.

Called from: BEAM (alternate ENTRY INJECT)

Variables required:

RRR — value of independent variable for which an interpolated value of the function (dependent variable) is desired

XXX — table of equally spaced values of independent variable, x_i

PIDEN — table of values of dependent variable, f_i

DXX — independent variable spacing, $\Delta = x_{i+1} - x_i$

NN — the number of pairs of values (f_i, x_i) in the dependent and independent variable tables

Variable changed:

YTERP = f(x)

SUBRØUTINE FPMØME

Subroutine FPMØME is the main calling routine for evaluating the moments of the Fokker-Planck equation, which is used to describe the slowing down of the injected neutral beam in the toroidal plasma. The moments are evaluated by a series of 30 function subprograms. Figure 12 shows the function routines and order of calls used to do the moments calculation. The expressions for the moments were derived by J. D. Callen.[4] The variables described below are used throughout the function subprograms, so are defined here only to avoid unnecessary repetition.

The velocity of the neutral beam at the point of entrance into the plasma is given by

$$\text{VO}\emptyset = v_o\sqrt{\frac{2\text{EO}\emptyset}{m_f}} \quad \text{, cm/sec}$$

where

EOØ = E_o = the energy of the injected neutral beam component, eV

and m_f = the mass of a neutral beam particle, amu.

The energy E_c, above which the injected neutral beam slowing is due to the plasma electrons and below which slowing is due to plasma ions, is given by

$$\text{EC}\emptyset = E_c = 14.8\ T_e\ (m_f/m_H)^{1/3}\ (m_f/m_i)^{2/3}\ [Z]^{2/3}\text{, eV}$$

where

m_i = mass of the plasma ions, amu

m_H = mass of the proton, amu

T_e = electron temperature, eV and

$$[Z] = (1/n_e)\left\{\sum_{\text{species, }k} (m_H/m_k) n_k Z_k{}^2\right\}\Big/\left\{\sum_{\text{species, }k} n_k Z_k\right\} \quad .$$

The velocity v_c corresponding to E_c is given by

$$\text{VC}\emptyset = v_c = \sqrt{\frac{2E_c}{m_f}}\text{ , cm/sec} \quad .$$

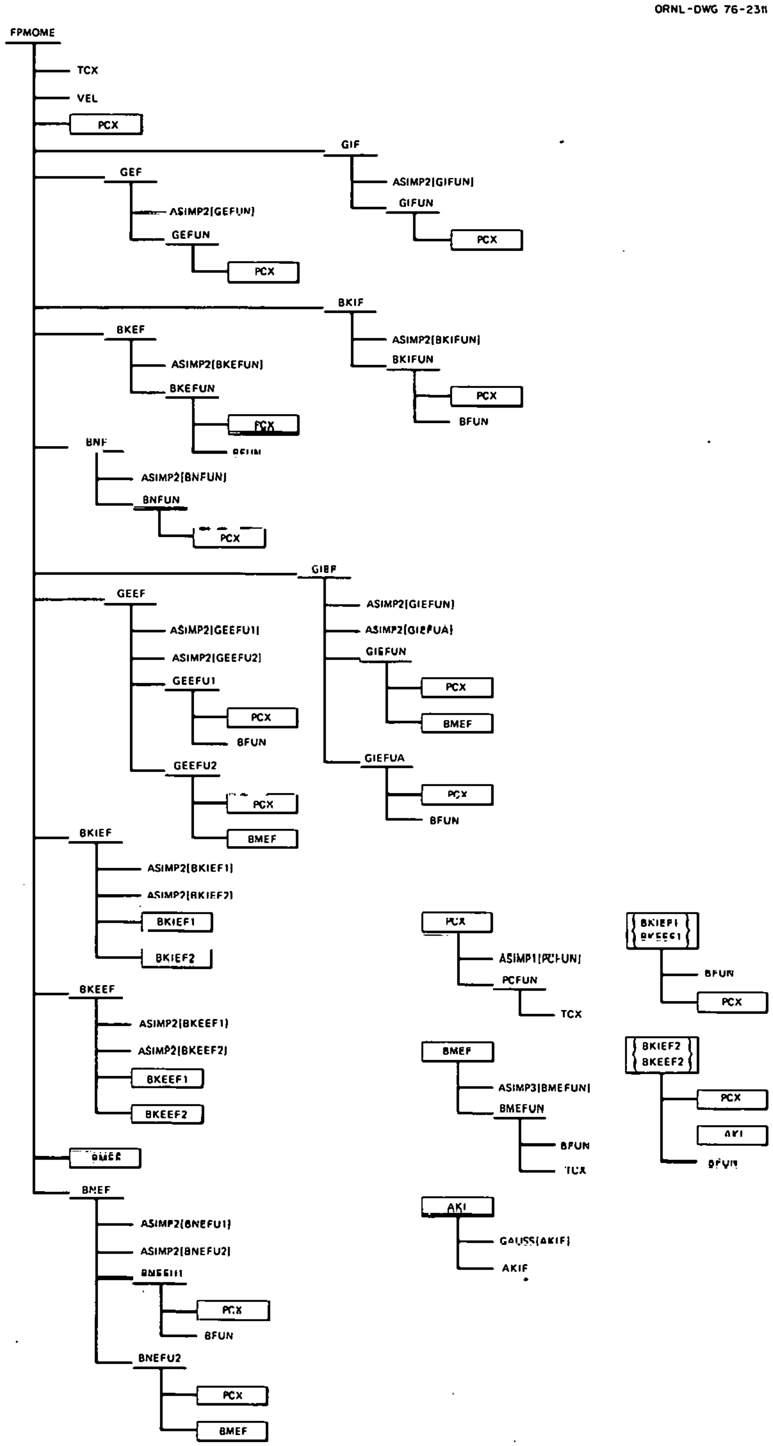


Fig. 12. Diagram of function routines and their order of call for the calculation of the moments of the Fokker-Planck equation.

The neutral beam slowing down time τ_s is given by

$$TSØ = \tau_s = 0.12(T_e/10^3 eV)^{3/2}\ (m_f/m_H)/[(n_e/10^{13})Z_f^{\,2}] \quad , \text{sec}$$

where n_e is the plasma electron density, in units of $10^{-13}cm^{-3}$.

The time interval τ_f, used as one of the criteria to determine when the Fokker-Planck code[4] solution has reached a steady state condition, is given by

$$TFØ = \tau_F = \frac{\tau_s}{3}\,\ell n[1+(v_o/v_c)^3] \quad .$$

At time t following entrance into the plasma, a fast ion generated from the injected neutral beam has slowed to a velocity v(t) and corresponding energy E(t) given by

$$ETØ = E(t) = \frac{1}{2}\,m_f v^2(t),\ eV \quad .$$

The remaining variables are described in detail under the headings of the various function subprograms called by subroutine FPMØME.

Called from: BEAM (alternate ENTRY INJECT)

Functions called: BKEEF, BKEF, BKIEF, BKIF, BMEF, BNEF, BNF, GEEF, GEF, GIEF, GIF, PCX, TCX, VEL

Commons required: INPUT, ØUTPUT (see Appendix II)

Variables changed: EOØ, ECØ, ETØ, TØ, TSØ, TFØ, VOØ, VCØ, VTØ

TCXVØ — $\tau_{cx}(v)$, the mean lifetime of a neutral beam particle before undergoing a charge-exchange event with a plasma ion

GE(GI) — fraction of injected neutral beam power transferred to the plasma electrons (ions)

BKE(BKI) — fraction of fast ion momentum transferred to the plasma electrons (ions)

BN — number of stored fast ions resulting from the injected neutral beam

PCXVT — probability that a charge-exchange event will not occur during the time a neutral beam particle slows from the injection velocity v_o to the velocity v(t)

GEE(GIE) — correction to the injected neutral beam power transferred to the plasma electrons (ions) due to the electric field

BKEE (BKIE) — correction to momentum transferred from injected neutral beam to the plasma electrons (ions) due to the electric field

BME — an integral arising from one of the moments of the Fokker-Planck equation and related to the electric field correction

BNE — correction to the number of stored fast ions due to the electric field

FUNCTIØN VEL (TIME)

Function VEL computes the velocity of the injected neutral beam at a point along the beam path reached by the beam particles at a time TIME following initial entrance into the plasma and taking into account the slowing down process. The expression used is

$$v(t) = \begin{cases} [v_o{}^3 \exp\{-3t/\tau_s\} - v_c{}^3(1 - \exp\{-3t/\tau_s\})]^{1/3} & , t < \tau_F \\ \sqrt{\dfrac{2E_{ion}}{M_{ion}}} & , t > \tau_F, \text{ or } t > 10^6 \text{ if } \tau_F > 10^6 \end{cases}$$

where

$$\tau_F = \frac{\tau_s}{3} \ell n\ [1 + v_o{}^3/v_c{}^3] \quad ,$$

v_o ≡ velocity of neutral beam at point of entrance into the plasma, and

v_c ≡ a critical velocity, above which the neutral beam slowing is due to the plasma electrons and below which slowing is due to plasma ions.

Called from:	FPMØME
Commons required:	INPUT, ØUTPUT (see Appendix II)
Variable required:	TIME
Variables changed:	VEL = v(t)

FUNCTIØN TCX (V)

Function TCX returns the value of the mean lifetime (in seconds) of a neutral beam particle before undergoing a charge-exchange collision with a plasma ion. The expression for this lifetime is

$$\mathrm{TCX} = \tau_{cx}(v) = \frac{6.6(10)^{-3}\left[\sqrt{\frac{25(10)^3}{E}}\,(1.1)(10)^{-15}E^{3.3} + \sqrt{\frac{25(10)^3}{E}}\right]}{(n_o/10^8)[1 - 0.155 \log E]^2}$$

where

$$E = \frac{1}{2} m_H v^2 \quad .$$

Called from: FPMØME, BMEFUN

Common required: INPUT (see Appendix II)

Variable required: V = v

Variable changed: TCX

Significant internal variable:

DNO = $n_o/10^8$ where n_o is the plasma neutral density, cm^{-3}

FUNCTIØN PCX (V, ICTCX)

Function PCX gives the probability of a charge-exchange event not occurring between an injected neutral beam particle and the plasma as the neutral beam particle slows down. This probability is given by

$$\mathrm{PCX} \equiv p_{cx}v(t) = \exp\left\{ - \tau_s \int_{v(t)}^{v_o} \frac{v'^2 dv'}{v'^3+v_c^3} \frac{1}{\tau_{cx}(v')} \right\} .$$

where

$v(t)$ — velocity to which neutral beam has slowed after time t following entrance into the plasma,

τ_s — injected neutral beam slowing down time,

$\tau_{cx}(v')$ — mean lifetime of a neutral beam particle before undergoing charge exchange with a plasma ion,

v_o — neutral beam injection velocity,

v_c — critical velocity, above which the neutral beam slowing is due to the plasma electrons and below which slowing is due to plasma ions.

Called from: FPMØME, GIFUN, GEFUN, BKIFUN, BKEFUN, BNFUN, GIEFUN, GIEFUA, GEEFU1, GEEFU2, BKIEF1, BKIEF2, BKEEF1, BKEEF2, BNEFU1, BNEFU2

Subroutines called: ASIMP1, PCFUN (through an EXTERNAL reference)

Commons required: INPUT, ØUTPUT (see Appendix II)

Variables required: V = v(t)

ICTCX — switch used to choose expression to be evaluated for value of PCX. If τ_{cx} is constant (ICTCX ≠ 0), then the approximate relation used is

$$p_{cx}(v) \simeq [(v_o^3 + v_c^3)/(v^3 + v_c^3)]^{-\tau_s/(3\tau_{cx})}$$

Variable changed: PCX

FUNCTIØN PCFUN (Y)

Function PCFUN returns the value of the integrand

$$\mathrm{PCFUN} = -\frac{\tau_s}{\tau_{cx}(v')}\,\frac{(v'/v_o)^2}{(v'/v_o)^3 + (v_c/v_o)^3}$$

used by the adaptive Simpson's rule quadrature subroutine called by function PCX. The variables are defined in the description of function PCX.

Called from: ASIMP1, referred to through EXTERNAL statement in function PCX

Commons required: INPUT, ØUTPUT (see Appendix II)

Variable required: $Y = v'/v_o$

Variable changed: PCFUN

m

FUNCTIØN GIF (VT)

Function GIF calculates the amount of injected neutral beam power transferred to the plasma ions according to the equation

$$GIF = G_i = \frac{2}{v_o^2} \int_{v(t)}^{v_o} vdv \frac{v_c^3}{v^3 + v_c^3} p_{cx}(v)$$

where

v_c — a critical velocity, above which the neutral beam slowing is due to the plasma electrons and below which slowing is due to plasma ions,

v_o — velocity of the neutral beam at point of entrance into the plasma,

$v(t)$ — velocity of the injected neutral beam at a point along the beam path reached at time t following initial entrance into the plasma,

$p_{cx}(v)$ — probability of avoiding a charge-exchange event by the neutral beam as it slows down,

v — velocity of the injected neutral beam.

Called from:	FPMØME
Subroutines called:	ASIMP2, GIFUN (through an EXTERNAL reference)
Commons required:	INPUT, ØUTPUT (see Appendix II)
Variable required:	VT = v(t)
Variable changed:	GIF

FUNCTIØN GIFUN (Y)

Function GIFUN calculates the value of the integrand for the integral calculated by function GIF. The expression for the integrand is referred to by the two-point Simpson's rule quadrature subroutine called by function GIF. The integrand expression whose value is computed is

$$\mathrm{GIFUN} = \frac{2\ p_{cx}(v)\,(v/v_o)\,(v_c/v_o)^3}{[(v/v_o)^3 + (v_c/v_o)^3]} \quad .$$

Called from: ASIMP2, referred to through EXTERNAL statement in GIF

Function called: PCX

Commons required: INPUT, ØUTPUT (see Appendix II)

Variable required: $Y = v/v_o$

Variable changed: GIFUN

FUNCTIØN GEF (VT)

Function GEF calculates the amount of neutral beam power transferred to the plasma electrons according to the equation

$$\mathrm{GEF} = G_e = \frac{2}{v_o^2} \int_{v(t)}^{v_o} v\,dv \, \frac{v^3}{v^3 + v_c^3} \; p_{cx}(v)$$

where

v_c — a critical velocity, above which the neutral beam slowing is due to the plasma electrons and below which slowing is due to plasma ions,

v_o — velocity of the neutral beam at point of entrance into the plasma,

$v(t)$ — velocity of the injected neutral beam at time t following initial entrance into the plasma,

$p_{cx}(v)$ — probability that a charge-exchange event does not occur as the neutral beam slows down in the plasma,

v — velocity of the neutral beam.

Called from:	FPMØME
Subroutines called:	ASIMP2, GEFUN (through an EXTERNAL reference)
Commons required:	INPUT, ØUTPUT (see Appendix II)
Variable required:	VT = v(t)
Variable changed:	GEF

FUNCTIØN GEFUN (Y)

Function GEFUN calculates the value of the integrand for the integral calculated by function GEF. The expression for the integrand is referred to by the two-point Simpson's rule quadrature subroutine called by the function GEF. The integrand expression whose value is computed is

$$\text{GEFUN} = \frac{2\ p_{cx}(v)(v/v_o)^4}{[(v/v_o)^3 + (v_c/v_o)^3]}\ .$$

Called from: ASIMP2, referred to through EXTERNAL statement in GEF

Subroutine called: PCX

Commons required: INPUT, ØUTPUT (see Appendix II)

Variables required: $Y = v/v_o$

Variable changed: GEFUN

FUNCTIØN BKIF (VT)

Function BKIF computes the amount of momentum transferred from the injected neutral beam to the plasma ions according to the equation

$$\mathrm{BKIF} = K_i = \frac{1}{v_o} \int_{v(t)}^{v_o} dv \, \frac{v_c^3}{v^3 + v_c^3} \left[1 + \frac{m_i \langle Z \rangle}{m_f [Z]} \right] p_{cx}(v) b(v)$$

where

$$b(v) = \left[\left(\frac{v_o^3 + v_c^3}{v^3 + v_c^3} \right) \frac{v^3}{v_o^3} \right]^{\frac{m_i \langle Z \rangle}{3 m_f [Z]}} ,$$

$$\langle Z \rangle = [\sum_k n_k Z_k^2] \Big/ [\sum_k n_k Z_k] ,$$

and

$$[Z] = [\sum_k \frac{m_{proton}}{m_k} n_k Z_k^2] \Big/ [\sum_k n_k Z_k] .$$

The remaining variables have been defined in the descriptions of FPMØME and of the first few functions called by FPMØME.

Called from: FPMØME

Subroutines called: ASIMP2, BKIFUN (through an EXTERNAL reference)

Commons required: INPUT, ØUTPUT (see Appendix II)

Variable required: VT = v(t)

Variable changed: BKIF

FUNCTIØN BKIFUN (Y)

Function BKIFUN returns the value of the integrand for the integral calculated by function BKIF. The expression for the integrand is referred to by the two-point Simpson's rule quadrature subroutine called by function BKIF. The integrand expression whose value is computed is

$$\text{BKIFUN} = \frac{(v_c/v_o)^3}{(v/v_o)^3 + (v_c/v_o)^3}\; P_{cx}(v) \left[1 + \frac{m_i\langle Z\rangle}{m_f[Z]}\right] \left[\left(\frac{v_o^{\,3} + v_c^{\,3}}{v^3 + v_e^{\,3}}\right)\frac{v^3}{v_o^{\,3}}\right]^{\frac{m_i\langle Z\rangle}{3m_f[Z]}} .$$

Called from:	ASIMP2, referred to through EXTERNAL statement in BKIF
Functions called:	PCX, BFUN
Commons required:	INPUT, ØUTPUT (see Appendix II)
Variable required:	$Y = v/v_o$
Variable changed:	BKIFUN

FUNCTIØN BFUN (V)

Function BFUN returns the value of the expression

$$\text{BFUN} = b(v) = \left[\frac{v_o{}^3 + v_c{}^3}{v^3 + v_c{}^3}\left(\frac{v}{v_o}\right)^3\right]^{\frac{m_i\langle Z\rangle}{3m_f[Z]}} .$$

Called from: BKIF, BKEF, GIEFUA, GEEFU1, BKIEF1, BKIEF2, BKEEF1, BKEEF2, BNEFU1

Commons required: INPUT, ØUTPUT (see Appendix II)

Variable required:

V — the velocity v of the neutral beam at a specified time following injection into the plasma

Variable changed: BFUN

FUNCTIØN BKEF (VT)

Function BKEF computes the amount of momentum transferred from the neutral beam to the plasma electrons according to the equation

$$\mathrm{BKEF} = K_e = \frac{1}{v_o} \int_{v(t)}^{v_o} dv \, \frac{v^3}{v^3 + v_c^3} \, p_{cx}(v) b(v)$$

where

$$b(v) = \left[\frac{v_o^3 + v_c^3}{v^3 + v_c^3} \left(\frac{v}{v_o} \right)^3 \right]^{\frac{m_i \langle Z \rangle}{3 m_f [Z]}} ,$$

$$\langle Z \rangle = \left[\sum_k n_k Z_k^2 \right] \Big/ \left[\sum_k n_k Z_k \right] ,$$

and

$$[Z] = \left[\sum_k \frac{m_{proton}}{m_k} n_k Z_k \right] \Big/ \left[\sum_k n_k Z_k \right] .$$

The remaining variables have been defined in the descriptions of FPMØME and of the first few functions called by FPMØME.

Called from:	FPMØME
Subroutines called:	ASIMP2, BKEFUN (through an EXTERNAL reference)
Commons required:	INPUT, ØUTPUT (see Appendix II)
Variable required:	VT = v(t)
Variable changed:	BKEF

FUNCTIØN BKEFUN (Y)

Function BKEFUN returns the value of the integrand for the integral calculated by function BKEF. The expression for the integrand is referred to by the two-point Simpson's rule quadrature subroutine called by function BKEF. The integrand expression whose value is computed is

$$\mathrm{BEKFUN} = \frac{(v/v_o)^3}{(v/v_o)^3 + (v_c/v_o)^3} \; p_{cx}(v) \left[\frac{v_o^{\,3} + v_c^{\,3}}{v^3 + v_c^{\,3}} \left(\frac{v}{v_o} \right)^3 \right]^{\frac{m_i<Z>}{3m_f[Z]}} .$$

Called from: ASIMP2, referred to through EXTERNAL statement in BKEF

Functions called: PCX, BFUN

Commons required: INPUT, ØUTPUT (see Appendix II)

Variable required: $Y = v/v_o$

Variable changed: BKEFUN

FUNCTIØN BNF (VT, ICT)

Function BNF computes the number of stored fast ions according to

$$\mathrm{BNF} = N = \int_{v(t)}^{v_o} \frac{v^2 dv}{v^3 + v_c{}^3} \; p_{cx}(v)$$

where all quantities have been defined in the descriptions of the first few functions called by FPMØME.

Called from: FPMØME

Subroutines called: ASIMP2, BNFUN (through an EXTERNAL reference)

Commons required: INPUT, ØUTPUT (see Appendix II)

Variables required: VT = v(t)

ICT — switch used to choose expression to be evaluated for value of BNF. If τ_{cx} is constant (ICT ≠ 0), then the approximate relation used is

$$N \approx \frac{\tau_{cx}}{\tau_s} \left[1 - e^{-t/\tau_{cx}}\right] \text{ with } t = \begin{cases} t & , t < \tau_F \\ \tau_F & , t > \tau_F \end{cases}$$

(This feature has not yet been implemented.)

Variable changed: BNF

FUNCTIØN BNFUN (Y)

Function BNFUN returns the value of the integrand for the integral calculated by function BNF. The expression for the integrand is referenced by the two-point Simpson's rule quadrature subroutine called by function BNF. The integrand expression whose value is computed is

$$\text{BNFUN} = \frac{p_{cx}(v)\,(v/v_o)^2}{(v/v_o)^3 + (v_c/v_o)^3} \ .$$

Called from:	ASIMP2, referred to through EXTERNAL statement in BNF
Function called:	PCX
Commons required:	INPUT, ØUTPUT (see Appendix II)
Variable required:	$Y = v/v_o$
Variable changed:	BNFUN

FUNCTIØN GIEF (V)

Function GIEF calculates the correction to the amount of injected neutral beam power transferred to the plasma ions due to the electric field effects according to the equations

$$GIEF = G_i^E = \frac{2}{v_o^2} \int_{v(t)}^{v_o} vdv \, \frac{v_c^3}{v^3 + v_c^3} \, p_{cx}(v) \left\{ \frac{\xi_o v_o v^2}{v^3 + v_c^3} \, b(v) - M^E(v) \right\}$$

where

$$M^E[v(t)] = BMEF = \xi_o v_o \int_{v(t)}^{v_o} \frac{v'^4 dv'}{v'^3 + v_c^3)^2} \, \frac{\tau_s}{\tau_{cx}(v')} \, b(v') \quad ,$$

$$b(v) = BFUN = \left[\frac{v_o^3 + v_c^3}{v^3 + v_c^3} \left(\frac{v}{v_o} \right)^3 \right]^{\frac{m_i <Z>}{3m_f [Z]}} \quad ,$$

$\xi_o = \cos\,\theta_o$ = the cosine of the neutral beam injection angle.

The remaining variables have been defined in the descriptions of FPMØME and the first few functions called by FPMØME.

Called from: FPMØME

Subroutines called: ASIMP2, GIEFUN (through an EXTERNAL reference)
GIEFUA (through an EXTERNAL reference)

Commons required: INPUT, ØUTPUT (see Appendix II)

Variable required: V = v(t)

Variable changed: GIEF

FUNCTIØN GIEFUN (Y)

Function GIEFUN returns the value of the integrand for the second of two integrals contributing to the expression evaluated by GIEF. The expression for the integrand

$$\text{GIEFUN} = 2\,\frac{v}{v_o}\,p_{cx}(v)\;M^E(v)\,\frac{(v_c/v_o)^3}{(v/v_o)^3 + (v_c/v_o)^3}$$

is referred to by the two-point Simpson's rule quadrature subroutine called by function GIEF.

Called from:	ASIMP2, referred to through EXTERNAL statement in GIEF
Functions called:	PCX, BMEF
Commons required:	INPUT, ØUTPUT (see Appendix II)
Variable required:	$Y = v/v_o$
Variable changed:	GIEFUN

FUNCTIØN BMEF (X)

Function BMEF returns the value of the expression

$$\text{BMEF} = M^{E}[v(t)] = \xi_o v_o \int_{v(t)}^{v_o} \frac{v'^4 dv'}{(v'^3 + v_c{}^3)^2} \frac{\tau_s}{\tau_{cx}(v')} b(v') \quad .$$

Called from: GIEFUN, GEEFU2, BNEFU2

Subroutines called: ASIMP3, BMEFUN (through an EXTERNAL reference)

Commons required: INPUT, ØUTPUT (see Appendix II)

Variable required: X = v(t)

Variable changed: BMEF

FUNCTIØN BMEFUN (Y)

Function BMEFUN returns the value of the integrand for the integral calculated by function BMEF. The expression for the integrand

$$\text{BMEFUN} = \frac{(v/v_o)^4\ \xi_o \tau_s b(v)}{[(v/v_o)^3 + (v_c/v_o)^3]^2 \tau_{cx}(v)}$$

is referred to by the three-point Simpson's rule quadrature subroutine called by function BMEF.

Called from:	ASIMP3, referred to through EXTERNAL statement in BMEF
Functions called:	BFUN, TCX
Commons required:	INPUT, ØUTPUT (see Appendix II)
Variable required:	$Y = v/v_o$
Variable changed:	BMEFUN

FUNCTIØN GIEFUA (Y)

Function GIEFUA returns the value of the integrand for the first of two integrals contributing to the expression evaluated by GIEF. The expression for the integrand

$$\text{GIEFUA} = \frac{2\xi_o (v/v_o)^3 (v_c/v_o)^3 p_{cx}(v) b(v)}{[(v/v_o)^3 + (v_c/v_o)^3]^2} ,$$

where $\xi_o = \cos\theta_o$, the cosine of the neutral beam injection angle, is referred to by the two-point Simpson's rule quadrature subroutine called by function GIEF.

Called from:	ASIMP2, referred to through EXTERNAL statement in GIEF
Functions called:	PCX, BFUN
Commons required:	INPUT, ØUTPUT (see Appendix II)
Variable required:	$Y = v/v_o$
Variable changed:	GIEFUA

FUNCTIØN GEEF (V)

Function GEEF calculates the correction to the amount of injected neutral beam power transferred to the plasma electrons due to the electric field effects according to the equations

$$\mathrm{GEEF} = \frac{2}{v_o^2} \int_{v(t)}^{v_o} vdv \, \frac{v^3}{v^3+v_c^3} \, p_{cx}(v) \left\{ \frac{\xi_o v_o v^2}{v^3+v_c^3} \, b(v) - M^E(v) \right\}$$

where

$$M^E[v(t)] = \mathrm{BMEF} = \xi_o v_o \int_{v(t)}^{v_o} \frac{v'^4 dv'}{(v'^3+v_c^3)^2} \, \frac{\tau_s}{\tau_{cx}(v')} \, b(v') \quad ,$$

$$b(v) = \mathrm{BFUN} = \left[\frac{v_o^3+v_c^3}{v^3+v_c^3} \left(\frac{v}{v_o} \right)^3 \right]^{\frac{m_i<Z>}{3m_f[Z]}} \quad ,$$

and $\xi_o = \cos \theta_o$, the cosine of the neutral beam injection angle. The remaining variables have been defined in the description of FPMØME and of the first few functions called by FPMØME.

Called from: FPMØME

Subroutines called: ASIMP2, GEEFU1 (through an EXTERNAL reference),
GEEFU2 (through an EXTERNAL reference)

Commons required: INPUT, ØUTPUT (see Appendix II)

Variable required: V = v(t)

Variable changed: GEEF

FUNCTIØN GEEFU1 (Y)

Function GEEFU1 returns the value of the integrand for the first of two integrals contributing to the expression evaluated by GEEF. The expression for the integrand

$$\mathrm{GEEFU1} = 2\xi_o p_{cx}(v) b(v) \left[\frac{(v/v_o)^3}{(v/v_o)^3 + (v_c/v_o)^3} \right]^2$$

where $\xi_o = \cos\theta_o$, the cosine of the neutral beam injection angle, is referred to by the two-point Simpson's rule quadrature subroutine called by function GEEF.

Called from: ASIMP2, referred to through EXTERNAL statement in GEEF

Functions called: PCX, BFUN

Commons required: INPUT, ØUTPUT (see Appendix II)

Variable required: $Y = v/v_o$

Variable changed: GEEFU1

FUNCTIØN GEEFU2 (Y)

Function GEEFU2 returns the value of the integrand for the second of two integrals contributing to the expression evaluated by GEEF. The expression for·the integrand

$$\text{GEEFU2} = 2p_{cx}(v)M^{E}(v)\,\frac{(v/v_o)^4}{(v/v_o)^3 + (v_c/v_o)^3}$$

is referred to by the two-point Simpson's rule quadrature subroutine called by function GEEF.

Called from:	ASIMP2, referred to through EXTERNAL statement in GEEF
Functions called:	PCX, BMEF
Commons required:	INPUT, ØUTPUT (see Appendix II)
Variable required:	$Y = v/v_o$
Variable changed:	GEEFU2

FUNCTIØN BKIEF (V)

Function BKIEF computes the correction to the amount of momentum transferred from the injected neutral beam to the plasma ions due to electric field affects according to the equations

$$\mathrm{BKIEF} = K_i^E = \frac{1}{3\xi_o} \int_{v(t)}^{v_o} dv \, \frac{v_c^3}{v^3 + v_c^3} \left[1 + \frac{m_i \langle Z \rangle}{m_f [Z]} \right] p_{cx}(v) \left\{ g(v) + K_{ie}^E \right\}$$

where

$$g(v) = \frac{v^2}{v^3 + v_c^3} \left[1 + (3\xi_o^2 - 1) b^3(v) \right] \quad ,$$

$$K_{ie}^E(v) = \mathrm{AKI(A)*BFUN(A)} = \int_{v}^{v_o} \frac{v' dv'}{v'^3 + v_c^3} \, \frac{b(v)}{b(v')} \left[3 - \left(1 + \frac{\tau_s}{\tau_{cx}(v')} \, \frac{v_{,}^3}{v'^3 + v_c^3} + \frac{m_i \langle Z \rangle}{m_f [Z]} \, \frac{v_c^3}{v'^3 + v_c^3} \right) \left(1 + (3\xi_o^2 - 1) b^3 \, (v') \right) \right] \quad ,$$

and.

$$b(v) = \mathrm{BFUN(A)} = \left[\frac{v_o^3 + v_c^3}{v^3 + v_c^3} \left(\frac{v^3}{v_o^3} \right) \right]^{\frac{m_i \langle Z \rangle}{3 m_f [Z]}}$$

and $\xi_o = \cos \theta_o$, the cosine of the neutral beam injection angle. The remaining variables have been defined in the descriptions of FPMØME and of the first few functions called by FPMØME.

Called from:	FPMØME
Subroutines called:	ASIMP2, BKIEF1 (through an EXTERNAL reference), BKIEF2 (through an EXTERNAL reference)
Commons required:	INPUT, ØUTPUT (see Appendix II)
Variable required:	V = v(t)
Variable changed:	BKIEF

FUNCTIØN BKIEF1 (Y)

Function BKIEF1 returns the value of the integrand for the first of two integrals contributing to the expression evaluated by BKIEF. The expression for the integrand

$$\mathrm{BKIEF1} = \frac{p_{cx}(v)}{3\xi_o}\left[\frac{\left(\frac{v_c}{v_o}\right)^3}{\left(\frac{v}{v_o}\right)^3 + \left(\frac{v_c}{v_o}\right)^3}\right]\left[1 + \frac{m_i\langle Z\rangle}{m_f[Z]}\right]\left[\frac{\left(\frac{v}{v_o}\right)^2}{\left(\frac{v}{v_o}\right)^3 + \left(\frac{v_c}{v_o}\right)^3}\right]$$

$$\times\ [1 + (3\xi_o{}^2 - 1)b^3(v)]$$

where $\xi_o = \cos\theta_o$, the cosine of the neutral beam injection angle, is referred to by the two-point Simpson's rule quadrature subroutine called by function BKIEF.

Called from: ASIMP2, referred to through EXTERNAL statement in BKIEF

Functions called: BFUN, PCX

Commons required: INPUT, ØUTPUT (see Appendix II)

Variable required: $Y = v/v_o$

Variable changed: BKIEF1

FUNCTIØN BKIEF2 (Y)

Function BKIEF2 returns the value of the integrand for the second of two integrals contributing to the expression evaluated by BKIEF. The expression for the integrand

$$\text{BKIEF2} = \frac{1}{3\xi_o}\left[1 + \frac{m_i\langle Z\rangle}{m_f[Z]}\right]\left[\frac{(v_c/v_o)^3}{(v/v_o)^3 + (v_c/v_o)^3}\right] p_{cx}(v)\, K_{ie}^{\;E} \quad ,$$

where

$$K_{ie}^{\;E} = \text{AKI(A)*BFUN(A)} = b(v)\int_v^{v_o} \frac{v'dv'}{v'^3+v_c^{\;3}}\,\frac{1}{b(v')}$$
$$\times \left[3 - \left[1 + (3\xi_o^{\;2} - 1)b^3(v')\right] \text{*BAA}\right]$$

with

$$\text{BAA} = 1 + \frac{\tau_s}{\tau_{cx}(v')}\,\frac{v'^3}{v'^3+v_c^{\;3}} + \frac{m_i\langle Z\rangle}{m_f[Z]}\,\frac{v_c^{\;3}}{v'^3+v_c^{\;3}} \quad ,$$

$A = v$,

and $\xi_o = \cos\theta_o$, the cosine of the neutral beam injection angle, is referred to by the two-point Simpson's rule quadrature subroutine called by function BKIEF.

Called from: ASIMP2, referred to through EXTERNAL statement in BKIEF

Functions called: PCX, AKI, BFUN

Commons required: INPUT, ØUTPUT (see Appendix II)

Variable required: $Y = v/v_o$

Variable changed: BKIEF2

FUNCTIØN AKI (V)

Function AKI returns the value of the integral

$$\text{AKI} = K_{ie}^{E}(v) = \int_{v}^{v_o} \frac{v'dv'}{v'^3+v_c^3}\,\frac{1}{b(v')}\left\{3 - [1 + (3\xi_o^2-1)b^3(v)]*\text{BAA}\right\}$$

where

$$\text{BAA} = 1 + \frac{\tau_s}{\tau_{cx}(v')}\,\frac{v'^3}{v'^3+v_c^3} + \frac{m_i\langle Z\rangle}{m_f[Z]}\,\frac{v_c^3}{v'^3+v_c^3} \quad .$$

Called from: BKIEF2, BKEEF2

Subroutines called: GAUSS, AKIF (through an EXTERNAL reference)

Commons required: GAØRD, INPUT, ØUTPUT (see Appendix II)

Variable required: V = v(t)

Variable changed: AKI

FUNCTIØN AKIF (Y)

Function AKIF returns the value of the integrand

$$\mathrm{AKIF} = \frac{1}{b(v)}\,\frac{(v/v_o)}{(v/v_o)^3 + (v_c/v_o)^3}\left\{3 - [1 + (3\xi_o{}^2-1)b^3(v)]*\mathrm{BAA}\right\} ,$$

where

$$\mathrm{BAA} = 1 + \frac{\tau_s}{\tau_{cx}(v)}\,\frac{(v/v_o)^3}{(v/v_o)^3+(v_c/v_o)^3} + \frac{m_i\langle Z\rangle}{m_f[Z]}\,\frac{(v_c/v_o)^3}{(v/v_c)^3+(v_c/v_o)^3} ,$$

used by the Gaussian quadrature subroutine called by function AKI.

Called from: GAUSS, referred to through EXTERNAL statement in AKI

Commons required: INPUT, ØUTPUT (see Appendix II)

Variable required: $Y = v/v_o$

Variable changed: AKIF

FUNCTIØN BKEEF (V)

Function BKEEF computes the correction to the amount of momentum transferred from the injected neutral beam to the plasma electrons due to electric field effects according to the equations

$$\mathrm{BKEEF} = K_e^{\,E} = \frac{1}{3\xi_o} \int_{v(t)}^{v_o} dv \, \frac{v^3}{v^3+v_c^{\,3}} \, p_{cx}(v) \left\{ g(v) + K_{ie}^{\,E} \right\}$$

where

$$g(v) = \frac{v^2}{v^3+v_c^{\,3}} \, [1 + 3\xi_o^{\,2} - 1)b^3(v)] \quad ,$$

$$K_{ie}^{\,E}\,(v) = \mathrm{AKI(A)*BFUN(A)} = \int_{v}^{v_o} \frac{v'dv'}{v'^3+v_c^{\,3}} \, \frac{b(v)}{b(v')} \left[3 - \left(1 + \frac{\tau_s}{\tau_{cx}(v')} \, \frac{v'^3}{v'^3+v_c^{\,3}} + \frac{m_i\langle Z\rangle}{m_f[Z]} \, \frac{v_c^{\,3}}{v'^3+v_c^{\,3}}\right) \left(1 + (3\xi_o^{\,2}-1)b^3(v')\right) \right] \quad ,$$

$$b(v) = \mathrm{BFUN(A)} = \left[\frac{v_o^{\,3} + v_c^{\,3}}{v^3 + v_c^{\,3}} \left(\frac{v}{v_o}\right)^3 \right]^{\frac{m_i\langle Z\rangle}{3m_f[Z]}} \quad ,$$

$$A = v,$$

and $\xi_o = \cos\theta_o$, the cosine of the neutral beam injection angle. The remaining variables have been defined in the descriptions of FPMØME and of the first few functions called by FPMØME.

Called from:	FPMØME
Subroutines called:	ASIMP2, BKEEF1 (through an EXTERNAL reference), BKEEF 2 (through an EXTERNAL reference)
Commons required:	INPUT, ØUTPUT (see Appendix II)
Variable required:	V = v(t)
Variable changed:	BKEEF

FUNCTIØN BKEEF1 (Y)

Function BKEEF1 returns the value of the integrand for the first of two integrals contributing to the expression evaluated by BKEEF. The integrand

$$\text{BKEEF1} = \frac{p_{cx}(v)}{3\xi_o}\left[\frac{\left(\frac{v}{v_o}\right)^3}{\left(\frac{v}{v_o}\right)^3+\left(\frac{v_c}{v_o}\right)^3}\right]\left[\frac{\left(\frac{v}{v_o}\right)^2}{\left(\frac{v}{v_o}\right)^3+\left(\frac{v_c}{v_o}\right)^3}\right][1 + (3\xi_o{}^2-1)b^3(v)] \quad ,$$

where $\xi_o = \cos\theta_o$, the cosine of the neutral beam injection angle, is referred to by the two-point Simpson's rule quadrature subroutine called by function BKEEF.

Called from:	ASIMP2, referred to through EXTERNAL statement in BKEEF
Functions called:	BFUN, PCX
Commons required:	INPUT, ØUTPUT (see Appendix II)
Variable required:	$Y = v/v_o$
Variable changed:	BKEEF1

FUNCTIØN BKEEF2 (Y)

Function BKEEF2 returns the value of the integrand for the second of two integrals contributing to the expression evaluated by BKEEF. The integrand

$$\mathrm{BKEEF2} = \frac{1}{3\xi_o}\left[\frac{(v/v_o)^3}{(v/v_o)^3 + (v_c/v_o)^3}\right] p_{cx}(v)\, K_{ie}^{E} \quad ,$$

where

$$K_{ie}^{E} = \mathrm{AKI(A)*BFUN(A)} = b(v)\int_{v}^{v_o} \frac{v'dv'}{v'^3+v_c^3}\,\frac{1}{b(v')}\left[3 - [1+(3\xi_o^{2}-1) \times b^3(v')]\mathrm{*BAA}\right] ,$$

with

$$\mathrm{BAA} = 1 + \frac{\tau_s}{\tau_{cx}(v')}\,\frac{v'^3}{v'^3+v_c^3} + \frac{m_i\langle Z\rangle}{m_f[Z]}\,\frac{v_c^{3}}{v'^3+v_c^3} \quad ,$$

$$A = v,$$

and $\xi_o = \cos\theta_o$, the cosine of the neutral beam injection angle, is referred to by the two-point Simpson's rule quadrature subroutine called by function BKEEF.

Called from:	ASIMP2, referred to through EXTERNAL statement in BKEEF
Functions called:	PCX, BFUN, AKI
Commons required:	INPUT, ØUTPUT (see Appendix II)
Variable required:	$Y = v/v_o$
Variable changed:	BKEEF2

FUNCTIØN BNEF (V)

Function BNEF computes the correction to the number of stored fast ions due to electric field effects according to

$$\mathrm{BNEF} = N^E = \int_{v(t)}^{v_o} \frac{v^2 dv}{v^3+v_c^3} p_{cx}(v) \left\{ \frac{\xi_o v_o v^2}{v^3+v_c^3} b(v) - M^E(v) \right\} ,$$

where

$$p_{cx}(v) = \mathrm{PCX} = \exp\left\{ -\tau_s \int_v^{v_o} \frac{v'^2 dv'}{v'^3+v_c^3} \frac{1}{\tau_{cx}(v')} \right\} ,$$

$$b(v) = \mathrm{BFUN}(v) = \left[\frac{v_o^3+v_c^3}{v^3+v_c^3} \left(\frac{v}{v_o} \right)^3 \right]^{\frac{m_i \langle Z \rangle}{3 m_f [Z]}} ,$$

and

$$M^E(v) = \mathrm{BMEF}(v) = \xi_o v_o \int_{v(t)}^{v_o} \frac{v'^4 dv'}{(v'^3+v_c^3)^2} \frac{\tau_s}{\tau_{cx}(v')} b(v') .$$

The variables have been defined in the descriptions of subroutine FPMØME and of the first few functions called by FPMØME.

Called from:	FPMØME
Subroutines called:	ASIMP2, BNEFU1 (through an EXTERNAL reference), BNEFU2 (through an EXTERNAL reference)
Commons required:	INPUT, ØUTPUT (see Appendix II)
Variable required:	V = v(t)
Variable changed:	BNEF

FUNCTIØN BNEFU1 (Y)

Function BNEFU1 returns the value of the integrand for the first of two integrals contributing to the expression evaluated by BNEF. The expression for the integrand

$$\text{BNEFU1} = 2\xi_o p_{cx}(v) b(v) \frac{(v/v_o)^2}{(v/v_o)^3 + (v_c/v_o)^3} ,$$

where $\xi_o = \cos \theta_o$, the cosine of the neutral beam injection angle, is referred to by the two-point Simpson's rule quadrature subroutine called by function BNEF.

Called from: ASIMP2, referred to through EXTERNAL statement in BNEF

Functions called: PCX, BFUN

Commons required: INPUT, ØUTPUT (see Appendix II)

Variable required: $Y = v/v_o$

Variable changed: BNEFU1

FUNCTIØN BNEFU2 (Y)

Function BNEFU2 returns the value of the integrand for the second of two integrals contributing to the expression evaluated by BNEF. The expression for the integrand

$$\mathrm{BNEFU2} = p_{cx}(v)\ M^E(v)\ \frac{(v/v_o)}{(v/v_o)^3 + (v_c/v_o)^3}$$

is referred to by the two-point Simpson's rule quadrature subroutine called by function BNEF.

Called from:	ASIMP2, referred to through EXTERNAL statement in BNEF
Functions called:	PCX, BMEF
Commons required:	INPUT, ØUTPUT (see Appendix II)
Variable required:	$Y = v/v_o$
Variable changed:	BNEFU2

REFERENCES

1. R. H. Fowler, J. D. Callen, J. A. Rome, and J. Smith, *FIFPC: A Fast Ion Fokker-Planck Code*, ORNL/TM-5487 (July 1976).

2. J. A. Rome, J. D. Callen, and J. F. Clarke, Nucl. Fusion 14: 141 (1974); R. J. Colchin and J. Wooten, ORMAK Technical Memo 163 (May 6, 1974).

3. B. Carnahan, H. A. Luther, and J. O. Wilkes, *Applied Numerical Methods*, John Wiley and Sons, Inc., New York, 1969, pp. 100-105.

4. J. D. Callen (ORNL), private communication.

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

7. DESCRIPTION OF SURFACE MODULE

7.1 PURPOSE AND STRUCTURE OF MODULE

The surface module calculates the plasma/wall interaction. It describes the neutral gas and impurity influx resulting from the interaction of the efflux of neutral atoms and charged particles with the limiter and walls.

We describe two different versions of this module in this section. The first version is essentially a dummy routine which returns only the value of the edge neutral density to the main program. This quantity is pre-programmed as a function of time, and uses the input data variables FCDEN, FCF, and TPØ. The second version is more complete. The processes of charge-exchange desorption and sputtering, photon stimulated desorption, electron impact desorption, and thermodesorption of the limiter are computed. However, these processes are not treated in as detailed a fashion as possible, but are included here chiefly to illustrate the function of the module.

1. Simple Model. The module consists of the single statement

```
FCDEX = FCDEN + (FCF-FCDEN)(1-EXP(-TIM/TPØ))
```

where FCDEN, FCF, and TPØ are described in Sect. 1.4. FCDEX is the value at the elapsed simulation time TIM of the edge neutral density $N_0(a)$.

Called from: MAIN

Commons required: NEUTRL, TIME

Variables required:

From labeled common (see Appendix II for definitions)

/NEUTRL/ — FCDEX, FCDEN, FCF

/TIME/ — TIM, TPØ

Variables changed:

FCDEX — edge neutral density at time TIM

2. Detailed Model. The plasma/wall interaction is computed in the more detailed model. The routine includes DATA statements describing the kinds of wall/limiter provided, so each different material will require a different module. The physical processes calculated by the module are:

a) Charge-exchange sputtering. In some versions of the code, the neutral gas module is used to compute the spectrum of emergent charge-exchange (hot) neutrals, $f_h(v,\theta,a)$. This spectrum is then integrated with experimental values of the energy and angle dependent sputtering yield to calculate the impurity flux which results:

$$\Gamma_z\left(\frac{\text{number}}{\text{cm}^2\ \text{sec}}\right) = 2\pi\int_0^\infty dv.v^2\int_0^\pi d\theta\ \sin 0.v\ \cos\theta\ f_h(v,\theta,a)Y(v,\theta)\quad .$$

The geometry v,θ is described in the introduction to Sect. 4. For this version, however, we will use results from more detailed calculations to note that the number of impurities produced per eV of charge-exchange energy is nearly constant, so we will provide this quantity in DATA statements. Thus, the impurity flux is

$$\Gamma_z^{\ cx} = 6.25(10)^{21}\overline{Y}\ |\,\text{PSCX}\,|\ /\text{AREA}$$

where PSCX is the total charge-exchange loss (in kW) from PIIL (calculated in subroutine ØRMPWR) and PBCX (calculated in subroutine BEAM).

$\overline{Y}$ — number of impurities per eV, given in a DATA statement

AREA — torus surface area

b) Charge-exchange desorption. For this quantity we add PHICX to the data, representing the number of impurities desorbed/charge-exchange neutral. The routine first finds an approximation to the mean energy of charge-exchange loss by finding the radius with maximum charge-exchange emission. Then the flux of charge-exchange particles is:

$$\Phi_{cx} = 6.25(10)^{21}[\,|\,\text{PHIL}\,|\ /\text{TCX} + \text{PBCX}/\text{EBCX}]$$

where

PIIL is calculated in subroutine ØRMPWR,

TCX is the mean neutral energy, and

PBCX, EBCX are the charge-exchange loss from the beam (kW), and mean energy of lost particle (eV), respectively.

The resulting impurity flux is:

$$\Gamma_{CX}^{CXD} = 6.25(10)^{21}\ \overline{Y}_D\ (|\text{PIIL}|/\text{TCX} + \text{PBCX/EBCX})/\text{AREA}.$$

c) Photon-stimulated desorption. For this process we use the mechanism whereby photoelectrons, produced by photons in the 10-100 eV energy range, bombard the walls. We consider two sources of photon emission: photons from O^{VI} resonance line radiation (1032 Å) and from hydrogen Ly alpha emission. Thus we have the sources PLR, computed in subroutine ØRMPWR from the oxygen emission, and PHY, computed in subroutine SURFAC. The photon fluxes are

$$\Phi_{O^{VI}} = 6.25(10)^{21}\ \text{PLR/EØ6}$$

where

EØ6 = 12 eV .

The photon flux for Lyα must be computed. The rate coefficient is computed in FUNCTIØN SLYAD. The photon flux emission is integrated over the plasma volume to obtain

$$\Phi^{\phi} = 2\pi\int_0^a dr.r\ N_e\ N_o\ S_x[T_e(r)] \quad .$$

The impurity stimulation rate is characterized by the basic quantum efficiency QEFF defined in a DATA statement. Thus the impurity flux is

$$\Gamma_z^{\phi} = \text{QEFF*(PHIØ + PHILA)}, \qquad (\#/\text{cm}^2/\text{sec}) \quad .$$

d) Charged-particle desorption. The plasma ions and electrons which leave also produce desorption characterized by the efficiencies $\text{QPR}\binom{W}{L}$ and $\text{QELE}\binom{W}{L}$ [number of impurities produced per proton and per electron, respectively (for wall and limiter)]. The emergent fluxes of protons and electrons are assumed to be equal, and are computed from the quantities PCVE, PCVI in subroutine ØRMPWR. The relative fraction of this flux going to the wall (limiter) is described by the constant in DATA PCTWAL (1-PCTWAL). Thus, the particle desorption contribution to the impurity flux is:

$$\Gamma_z^P = (\text{PCVE}*((\text{QELEW}*\text{PCTWAL} + \text{QELEL} *(1-\text{PCTWAL}))/\text{TE (N)} + \text{PCVI} *(\text{QPRW}*\text{PCTWAL} + \text{QPRL} *(1-\text{PCTWAL}))/\text{TI(N)})\ 6.25\ (10)^{21}.$$

e) Total. The total impurity production is thus

$$\Gamma_z = \Gamma_z^{CX} = \Gamma_z^{CXD} + \Gamma_z^{\phi} + \Gamma_z^{P} \quad .$$

This flux is treated in two ways, depending whether or not a detailed impurity model is being used [IMP(1) = 1 or 2]. With IMP(1) = 1, the ZEFF value is incremented uniformly by addition of the new impurity flux Γ_z. The rule is

$$\dot{Z}_{eff} = \dot{N}_z Z^2/N_e$$

where $N_{e,z}$ are the total numbers of electrons and impurities, Z (ZIMP) is the average Z of the impurity species and $N_z = \Gamma_z \times \text{AREA}$. With the detailed model the impurities are added to the species being monitored. Thus one can discuss heavy metals (from the wall or limiter), light impurities (from desorption), or hydrogen. For this discussion we will treat only the case of light impurities produced by the sum of all the processes discussed above. In fact, we will assume the impurity to be CARBØN, and the impurity is to be added to the CARBØN array in the last spatial zone. Thus, the increment of carbon atoms produced during time Δt is

$$\text{TDELC} = \Delta t*\Gamma_z*\text{AREA}$$

and the density in the outer spatial zone is

$$\text{DELC} = \text{TDELC}/2\pi R_o 2\pi a(\text{HR}) \quad .$$

7.2 BASIC INPUT/OUTPUT

The routine requires some of the major energy and particle fluxes from the plasma which are calculated in subroutine ØRMPWR. It calculates either an increment in ZEFF [if IMP(1) = 1], or an increment in the CARBØN(N) density [if IMP(1) = 2].

7.3 SUMMARY OF FUNCTION OF EACH SUBROUTINE

SUBRØUTINE SURFAC

Subroutine SURFAC is currently the only subroutine in this module.

Called from: MAIN

Commons required: IMPURT, INDEX, NEUTRL, TIME, SURF, PØWER, ØUTPT, RATIØS, GEØM, IØNS

Variables required:

From labeled common blocks (described in Appendix II):

/IMPURT/ — CARBØN, ZEFF, ZIMP, DENP

/INDEX/ — N

/TIME/ — TIME, TPØ

/NEUTRL/ — FCF, FCDEX, FCDEN

/SURF/ — CARN

/POWER/ — PED, PID, PIIL

/ØUTPT/ — PLR

/RATIØS/ — PCVE, PCVI

/CEØM/ — AREA

/IØNS/ — DEN

In a DATA statement:

YBAR — number of impurities per eV

YBARC — number of impurities of carbon per eV

QEFF — basic quantum efficiency

QELE$\begin{pmatrix}W\\L\end{pmatrix}$ — electron desorption efficiencies for wall (W) and limiter (L)

QPR$\begin{pmatrix}W\\L\end{pmatrix}$ — ion desorption efficiencies for wall (W) and limiter (L)

YBARD — number of impurities from charge-exchange desorption per eV

Variables changed:

ZEFF — effective charge Z_{eff} of plasma ions

CARN — edge density of carbon atoms, cm^{-3}

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

8. DESCRIPTION OF DIAGNOSTIC MODULE

8.1 PURPOSE AND STRUCTURE OF MODULE

Development of this module has just begun. It will contain all the diagnostic procedures, checks on numerical accuracy, error messages, and diagnostic printout of variables and arrays to help locate where problems are occurring during a simulation run. Three levels of diagnostics will be considered in order of increasing severity of problems encountered.

Level one diagnostics. This level includes checks on numerical accuracy, some consistency checks and associated error messages. Errors occurring at this level will not stop a simulation run unless the user specifies that this be done by his choice of input data for setting various switch variables. The following numerical accuracy checks are automatically made:

a) The computed current

$$I = 2\pi \int dr \; r \; j(r)$$

is compared to the program current.

b) The quantity $\eta(j - j_{BEAM})$ is compared to the electric field E.

c) The quantity

$$-(NV)/(D\frac{\partial N_e}{\partial r})$$

should be unity.

Level two diagnostics. This level will contain diagnostic procedures and associated error messages that can be used to attempt to recover if the simulation is going badly. Problems encountered at this level will be more serious. An option will exist for stopping the program by the choice of input data for various switch variables.

Level three diagnostics. This will be the level of diagnostics for the most serious problems. This level will be used when extensive diagnostic information is required in order to locate the source of a problem. It must be requested deliberately by the user since it will usually generate a large amount of output.

8.2 SUMMARY OF FUNCTION OF EACH SUBROUTINE

The only subroutine at present is DIAGNØ. It provides a few level one and one level three diagnostic aids.

SUBRØUTINE DIAGNØ

Subroutine DIAGNØ contains the three level one diagnostics mentioned above. A level three diagnostic was added when some problems occurred in subroutine IBTM in the plasma module, and checks for vanishing matrix elements in the coefficient matrix A of the set of finite-differenced coupled partial differential equations describing the plasma evolution. If a vanishing matrix element is found, it is given the value 1.0 and the location of that element is printed out using alternate ENTRY DPRT. This allows the program to continue even though the simulation is in error, so that further degeneration of the computation can occur and perhaps yield more helpful diagnostic information. Alternate ENTRY DAPRT examines the coefficient array for elements $>10^{40}$ and prints any that are found.

Called from: MAIN, IBTM [alternate ENTRY DAPRT]

Commons required: BEAMC, CDCLBX, FIELDS, GEØM, INDEX, IØNS, LBMIBT, SUMS, TEMP, THRMDF

Variables required:

For ENTRY DPRT (KDUM, IDUM, LØØP, VALUE)

KDUM — a variable used to isolate which group of statements was being executed when trouble occurred. The value of this variable is printed during program execution and is also a useful aid when examining the contents of a core dump when execution fails.

IDUM — the product of the DØ loop indices I1*I2 in subroutine IBTM. The value of IDUM is printed during program execution if the value of certain elements of the coefficient matrix A are $<10^{-40}$.

LØØP — index value of the A-array element causing difficulty.

VALUE — value of the quantity 1./A(LØØP).

From labeled common blocks (See Appendix II for definitions)

/BEAMC/ — DJT(101)

/CDCLBX/ — ETA(101)

/FIELDS/ — ZJ(101)

/GEØM/ — AM

/INDEX/ — N

/IØNS/ — DEN(101)

/LBMIBT/ — A(8008)

/SUMS/ — SUMC

/TEMP/ — VEL(101)

/THRMDF/ — TDE(101)

Variables changed:

RCUR — ratio of computed current to program current (should be unity).

SIGV — ratio of plasma ion flux computed from drift velocity to flux computed with thermal diffusion coefficient (should be unity).

APPENDIX I

Appendix I contains a listing of the input data set used by the transport code. This listing contains two parameters, SCAIJ and SCAEP1, which are not described in the User's Manual. They appear with the parameters listed under the heading NEUTRAL GAS DATA and are included in the NAMELIST list | NEGADA | in subroutine GETSET. They are not used in the version of the transport code described in this manual. Except for these two parameters, the rest of the listing should agree with the description in the User's Manual text.

```
    THIS IS THE OFFICIAL INPUT DATA SET FOR THE VERSION OF THE OAK
RIDGE TOKAMAK TRANSPORT CODE AS DESCRIBED IN THE USER'S MANUAL.
I N P U T   D A T A   A N D   S W I T C H   S E T T I N G S
    THE FOLLOWING INPUT DATA SET IS COMPLETELY INTERNALLY DOCUMENTED SO
THAT THE CASUAL USER SHOULD BE ABLE TO RUN THE PLASMA SIMULATION CODE
WITHOUT HAVING TO CONSULT A SERIES OF MEMOS.  CHANGES TO THE INPUT DATA
ARE MADE BY EDITING THIS FILE, I.E. THE NAMELIST LISTS.
    THE VARIABLES APPROPRIATE FOR EACH NAMELIST LIST GROUPING ARE
DESCRIBED IMMEDIATELY PRECEEDING THE NAMELIST DATA SET.  DEFAULT VALUES
FOR THE INPUT DATA ARE GIVEN IN PARENTHESES FOLLOWING THE
DEFINITIONS OF THE VARIABLES.  THE DEFAULT VALUES GIVEN ARE THOSE
APPROPRIATE FOR THE ORMAK DEVICE.

M A C H I N E   C O N F I G U R A T I O N
    AM      MINOR RADIUS OF TORUS, CM       (23.0CM)
    BO      MAJOR RADIUS OF TORUS, CM       (79.8CM)
    BT      TOROIDAL MAGNETIC FIELD, GAUSS     (18KG)
    R9      INITIAL PLASMA RADIUS,CM      (23.0CM)
    N       NUMBER OF SPATIAL GRID (RADIAL) POINTS       (51)
 &MACHIN
 AM=23.0,RO=79.5,BT=18.E3,R9=23.0,N=51
 &END
C
T I M I N G   I N T E R V A L S
    TMAX    MAXIMUM DURATION OF CURRENT PULSE, MSEC      (65.0MSEC)
    TC      RISE TIME OF THE CURRENT, MSEC      (10.0MSEC)
    TPO     FALLOFF TIME OF EXTERNAL NEUTRAL DENSITY, MSEC        (4.0MSEC)
    TSTAR    SIMULATION PROBLEM STARTING TIME, MSEC      (0.0MSEC)
    NT      TOTAL NUMBER OF TIME INTERVALS FOR CURRENT PULSE AND PLASMA
            SIMULATION      (1000)
 &TIMINT
 TMAX=65.0,TC=10.0,TPO=4.0,TSTAR=0.0,NT=1000
 &END
C
P L A S M A   D A T A
    TEO     BOUNDARY TEMPERATURE FOR ELECTRONS (IONS), EV      ( 10.EV)
    TIO     BOUNDARY AND INITIAL TEMPERATURE OF IONS, EV      (10.EV)
    TEB     INITIAL CENTRAL VALUE OF ELECTRON TEMPERATURE, EV       (20.0EV)
    TIB     INITIAL CENTRAL VALUE OF ION TEMPERATURE, EV       (20.EV)
    VOLT    TOROIDAL VOLTAGE, V       (3.0V)
    DENO    INITIAL VALUE FOR DENSITY AT PLASMA CENTER, CM**-3       (6.E12)
    DENB    BOUNDARY ELECTRON DENSITY, CM**-3       (5.E12CM**-3)
    ZPC     INITIAL VALUE OF THE TOTAL CURRENT, AMPS       (15.0KA)
    ZPI     ADDITIONAL AMOUNT OF TOTAL CURRENT AFTER TIME TC, AMPS       (85.0KA)
    AMU     MASS OF PLASMA IONS (SEE TABLE FOLLOWING), AMU        (1.0AMU)
 *** N O T E          AS OF 25 FEB 75, CONSTANT AM DOES NOT APPEAR
 ***                  IN ALL EXPRESSIONS WHERE IT IS REQUIRED.  UNTIL
 ***                  THIS DEFFICIENCY IS REMEDIED, AMU=1.0  A L W A Y S
=====================================================================
      FILLING GAS                                    AMU
---------------------------------------------------------------------
100% HYDROGEN                                        1.0
75% HYDROGEN, 25% DEUTERIUM                          1.25
50% HYDROGEN, 50% DEUTERIUM                          1.50
      ETC.

    XI(1)   EXPONENT FOR INITIAL PROFILE OF CURRENT DENSITY (2.0)
    XI(2)                                   ELECTRON TEMP. (2.0)
    XI(3)                                   ION TEMP. (2.0)
    XI(4)                                   ELECTRON DENSITY (2.0)
    INIT(1) STARTING MODE FOR INITIAL PLASMA VARIABLE PROFILES  (0)
                =0, ANALYTIC FUNCTION GENERATED PROFILES
                =1, PROFILES READ FROM DATA CARDS

    INIT(2) NOT CURRENTLY USED
    INIT(3) SWITCH FOR EXAMINING MHD STABILITY CHARACTERISTICS OF THE  (1)
            PLASMA
              =0, DETERMINES LOCATION OF SINGULAR SURFACE FOR M=3,5
              =1, CALCULATES RADIAL EIGENFUNCTIONS AND TEARING MODE GROWTH
                  RATE ACCORDING TO FORTH, RUTHERFORD, AND SELBERG, FOR
                  M=3,5.  CAN BE USED TO EVALUATE TRANSPORT COEFFICIENTS IN
                  SUBROUTINE CDC.
```

```
   D11(1)  ELECTRON THERMAL TRANSPORT COEFFICIENT, FOR SCALING DIFFUSION
           COEFFICIENT   = D11(1) * PSEUDO-CLASSICAL      (0.350)
   D11(2)  ION THERMAL TRANSPORT COEFFICIENT, FOR SCALING DIFFUSION
           COEFFICIENT   = D11(2) * NEO-CLASSICAL      (0.300)
   D11(3)  PARTICLE DIFFUSION COEFFICIENT    = 1.00 * PSEUDO-CLASSICAL
   D11(4)  RESISTIVITY    = D11(4) * NEO-CLASSICAL      (1.00)
   D11(5)  THRESHOLD (NUE*) OF TRAPPED ELECTRON SCALING       (0.0)
   D11(6)  THRESHOLD (NUI*) OF TRAPPED ION SCALING      (0.0)
 &PLADAT
 TEO=10.0,TIO=10.0,TEB=20.0,TIB=20.0,VOLT=3.0,DENO=6.E12,DENB=5.E12,
 ZPC=1.5E4,ZPI=8.5E4,AMU=1.0,XI=2.0,2.0,2.0,2.0,0.0,0.0,0.0,0.0,0.0,0.0,
 INIT=0,0,1,0,0,0,0,0,0,0,
 D11=0.350, 0.300, 1.0, 1.0, 0.0, 0.0
 &END
C
N E U T R A L   G A S   D A T A
   FCDEX(TIM)=(FCDEN-FCP)EXP(-TIM/TPO)+FCP
   FCDEX   FRANCK-CONDON BOUNDARY DENSITY OF NEUTRALS AT TIME=TIM
   FCDEN   FRANCK-CONDON BOUNDARY DENSITY FCDEX(0), CM**-3      (18.0E09CM**-3)
   FCP     FINAL BOUNDARY DENSITY OF NEUTRALS FCDEX(INF), CM**-3 (9.0E09CM**-3)
   NOX     NUMBER OF POINT3 IN [0,1] AT WHICH TO APPROXIMATE
               HOT NEUTRAL GAS DENSITY      (21)
   NOZ     NUMBER OF UNIFORMLY DISTRIBUTED POINTS IN [-1,1] USED TO
               APPROXIMATE HOT NEUTRAL GAS DENSITY      (101)
   NSIMP   NUMBER OF SIMPSON-NODES USED TO APPROXIMATE AN INTEGRAL NEAR
               A LOGARITHMIC SINGULARITY IN G(Q)      (9)
   NUNIF   SWITCH FOR THE NODAL POINTS IN [0,1]      (0)
               = 0, POINTS ARE DEFINED IN NEUTIO SUBROUTINE TO BE
                     (I-1)/(NOX-1)
               > 0, POINTS SHOULD BE PLACED AT THE BEGINNING OF THE
                     ARRAY W1 IN COMMON /NUTRLP/
   QF, QC  DETERMINE AN INTEGRATION INTERVAL ON EACH SIDE OF THE LOGA-
               RITHMIC SINGULARITY OF G(Q) AT EACH NODAL POINT IN [0,1].
               REQUIRE 0 < OR = QF < QC.  WHENEVER QF > QC, NEUTIO
               SETS QF=0.1*QC.      (0.001,0.11)
   EPSNI   CONVERGENCE CRITERION FOR HOT NEUTRAL GAS DENSITY SOLUTION
               TO THE DISCRETE INTEGRAL EQUATION      (0.001)
   INTEPS  =1, 2, 3 SELECTS RESPECTIVELY LINEAR, QUADRATIC, OR CUBIC
              INTERPOLATION FOR APPROXIMATING NI, TI, AND EATN    (2)
   NSM0    =NUMBER OF DIVISION POINTS IN THE 2 PARTITION INTERVALS
               THAT CONTAIN THE POINT OF THE LOGARITHMIC SINGULARITY (3)
   IDEBUG  =0 FOR NO INFORMATION                                          (1)
           =1, PRINTS A=P1, B=P2, C=AFC, LIPSHITZ CONSTANT, ESTIMATED
               NUMBER OF ITERATIONS
           =2, PRINTS SAME AS =1 PLUS ARRAYS NI, TI (OR AI), AND EATN
   EO      ENERGY OF NEUTRALS OFF THE WALL, EV.     (10.)
   EPSN0   IF THE MAGNITUDE OF THE RELATIVE ERROR BETWEEN CURRENT    (.5)
           A*NI/AI AND ONE USED AT PREVIOUS TIME TO GET LAST BASIC
           NEUTRAL PROFILE IS .GT. EPSN0, THEN A NEW SLAB CALCULATION
           WILL BE MADE
   IPN0    APPLICABLE ONLY IF NEW SLAB CALCUATION IS TO BE MADE     (1)
             =0 FOR NO PRINT
             =1, PRINTS RANGE OF RELATIVE ERROR CURVE (CF. EPSN0)
             =2, PRINTS SAME AS =1 PLUS OLD AND NEW NI/AI AND NEUTRALS
   NWDW1   NUMBER OF WORDS IN ARRAY W1 OF COMMON /NUTRLP/


   NWDWSC  NUMBER OF WORDS IN ARRAYS IN COMMON /SPACE/
   NUTE(1) SWITCH FOR MODEL OF NEUTRAL GAS CALCULATION      (2)
               =1  FOR ANALYTIC MODEL
               =2  FOR SLAB MODEL (JOHN HOGAN), NO REFLECTION
                    NWDW1 .GE. (5N+4NOX+3NOZ)
                    NWDWSC .GE. NOX(NOZ+10 NSM0+5)+NSIMP(8 NOX+1) +4(NOZ+NSM0)
                                  -2
               =3  FOR SLAB MODEL WITH M. ROBINSON'S WALL REFLECTION
                   (DIETER SIGMAR, NOT AVAILABLE YET)
                    NWDW1 .GE. (5N+5NOX+3NOZ)
                    NWDWSC .GE. (15NOX+8NOZ+NOX*NOZ)
               =4  FOR XCDRN VERSION OF ANISN PROGRAM (NOT INCLUDED YET)
```

```
    SCAIJ   SCALING FACTOR FOR THE I (AND J) FUNCTION APPEARING IN THE  (1.)
            REFLECTION PART OF THE KERNEL.  APPLIES IF NUTE(1)=3.
    SCAEP1  SCALING FACTOR FOR THE FUNCTION EPSILON SUB 1 THAT APPEARS  (1.)
            IN THE REFLECTION PART OF THE KERNEL, APPLICABLE ONLY
            IF NUTE(1)=3.
    NUTE(2) EXPONENT IN POWER LAW FOR NEUTRAL GAS ANALYTIC MODEL      (4)
N.B.***IF NO SLAB CALCULATION, THEN LABELLED COMMONS /NUTRLP/, /SLAB/,
      /INTRPS/, /PRNT/ ARE NOT USED AND THE ARRAY W1 IN /NUTRLP/ SHOULD
      HAVE EXACTLY 1 ELEMENT.
***       MXIT, IPRIT, EGGT1, AND BNT SHOULD BE SPECIFIED WHENEVER JOHN
      HOGAN'S SLAB MODEL IS REQUESTED.  IPRIT, NC, EGGT1, AND BNT
      APPLY ONLY IF GAMMA (=LIPSCHITZ CONSTANT) .GE. 1 OR IF THE
      ESTIMATED NUMBER OF FUNCTIONAL ITERATIONS EXCEEDS MXIT.
      NEUTC SETS NC TO 1 (0) IF FAILURE (SUCCESS).  IT=NOIT IS THE
      NUMBER OF FUNCTIONAL ITERATIONS PERFORMED IN NEUTC.
      PRINT ROW OF INFORMATION ON ITERATIONS=1+(J-1)*IPRIT, J=1,...
      UNTIL CONVERGENCE OR FAILURE IS DECLARED.  N1 DENOTES THE
      HOT NEUTRAL DENSITY FUNCTION.  CONVERGENCE IS DECLARED WHEN-
      EVER SUM FOR I=1,...,NOX OF ABS(N1CUR(I)-N1LST(I)) .LE. (SUM
      OF N1CUR(I))*EGGT1.  IF MAX(N1CUR(I)) .GE. BNT, THEN FAILURE
      IS DECLARED.
 &NEGADA
 FCDEN=18.E9,FCF=9.E9,NOX=21,NOZ=101,NSIMP=9,NUNIF=0,QF=0.001,
 QC=0.11,EPSNI=0.001,NUTE=1,4,0,0,0,0,0,0,0,0,INTRPS=2,NSMO=3,IDEBUG=1,
 NWDW1=642,NWDWSC=8008,EO=10.,EPSNO=.5,MXIT=200,IPRIT=5,EGGT1=.001,
 BNT=10.,SCAIJ=1.,SCAEP1=1.,
 &END
C
I M P U R I T Y   I O N   D A T A
    PCIMP   PERCENTAGE OF ELECTRON DENSITY DUE TO A HIGH-Z IMPURITY.  (0.1)
            THIS PARAMETER HAS TWO DIFFERENT MEANINGS ACCORDING TO THE
            IMPURITY DIFFUSION MODEL OPTION DETERMINED BY THE VALUE
            OF IMP(2).  FOR IMP(2)=1 OR 3, VALUE IS DUE TO SOME EVERAGE
            EFFECTIVE HIGH-Z IMPURITY; FOR IMP(2)=2 OR 4, IT IS DUE TO
            IRON IMPURITIES.
    ZEFF    EFFECTIVE CHARGE OF IONS      (4.0)
    ZIMP    MAXIMUM CHARGE OF THE HIGH-Z IMPURITY ION    (25.0)
    XCAR    PERCENTAGE OF ELECTRON DENSITY DUE TO CARBON IMPURITIES
            (APPROX. 2%)
    XOX     PERCENTAGE OF ELECTRON DENSITY DUE TO OXYGEN IMPURITIES
            (APPROX. 2%)
    IMP(1)  SWITCH FOR INITIAL IMPURITY STRIPPING CALCULATION    (1)
               =1  CORONA EQUILIBRIUM TABLE FOR CARBON, OXYGEN
               =2  DYNAMIC RATE CALCULATION FOR CARBON, OXYGEN (NOT
                   AN OPTION IN CURRENT VERSION OF THE PROGRAM)
    IMP(2)  SWITCH FOR DIFFUSION OF IMPURITIES       (1)
               =1  A CORONA EQUILIBRIUM TABLE TOGETHER WITH RADIAL
                   DISTRIBUTION FORMULAE ARE USED FOR CARBON AND OXYGEN.
                   HIGH-Z IMPURITIES ARE TREATED USING A CRUDE ESTIMATE
               =2  CORONA EQUILIBRIUM TABLES USED FOR CARBON, OXYGEN, AND A
                   HIGH-Z IMPURITY (FE) ATOM CHARGE STATES.  RADIAL DISTRI-
                   BUTIONS ARE ASSIGNED USING FORMULAE.
               =3  A CORONA EQUILIBRIUM TABLE TOGETHER WITH RADIAL DISTRI-
                   BUTIONS GENERATED BY PFIRSCH-SCHLUTER DIFFUSION IS USED
                   FOR CARBON AND OXYGEN.  HIGH-Z IMPURITIES ARE TREATED USING
                   A CRUDE ESTIMATE.
               =4  CORONA EQUILIBRIUM TABLES ARE USED FOR CARBON, OXYGEN,
                   AND A HIGH-Z IMPURITY (FE) ATOM CHARGE STATES.  RADIAL
                   DISTRIBUTIONS ARE GENERATED BY A PFIRSCH-SCHLUTER DIFFUSION
                   MODEL.
    IMP(3)  SWITCH FOR IMPURITY MODEL CALCULATION            (1)
              =1  USE FIXED VALUES
              =2  USE MODEL CALCULATION DETERMINED BY CHOICE OF SWITCH VALUE
                  FOR IMP(2).
    IMP(4)  SWITCH FOR DETAILED MODEL FOR IRON          (1)
              =0  INCLUDE IRON CORONAL EQUILIBRIA
              =1  OMIT DETAILED DESCRIPTION
 &IMPIDA
 PCIMP=0.10,ZEFF=4.0,ZIMP=25.0,XCAR=2.0,XOX=2.0,IMP=1,1,1,
 &END
```

```
C
N E U T R A L    B E A M    I N J E C T I O N    D A T A
    CURBM    3 X 2 ARRAY OF NEUTRAL BEAM CURRENTS (SEE TABLE BELOW), AMPS
    ENGY(1)  HIGHEST NEUTRAL BEAM ENERGY, EV       (3.0E4 EV)

  TABLE OF NEUTRAL BEAM CURRENT ARRAY, CURBM(COLUMN,ROW), DEFAULT VALUES

                            ENGY(1)=EO     ENGY(2)=EO/2     ENGY(3)=EO/3
  CO-INJECTION                 1.2              1.2              1.2
  COUNTER-INJECTION            1.2              1.2              1.2

    DTCH     TIME INCREMENT FOR CALCULATION OF H(R), MSEC       (10.0MSEC)
    DTCGL    TIME INCREMENT FOR CALCULATION OF GE, GI, AND KE, MSEC      (2.0MSEC)
    RC       MAJOR RADIUS OF BEAMLINE POINT OF TANGENCY, CM       (75.0CM)
    RB       NEUTRAL BEAM RADIUS, CM        (15.0CM)
    BHLF                      HALF-WIDTH (ASSUMING GAUSSIAN PROFILE),CM      (7.5CM)
    TNJCT    STARTING TIME FOR INJECTION, MSEC          (20MSEC)
    DTNJCT   NEUTRAL BEAM INJECTION CURRENT RISE TIME, MSEC      (6.0MSEC)
    N1       NUMBER OF POINTS IN CURRENT RISE INTERVAL       (500)
    N3       NUMBER OF POINTS IN INJECTED NEUTRAL BEAM RISE TIME INTERVAL    (200)
    FQE      FRACTION OF INJECTED NEUTRAL BEAM POWER TRANSFERRED    (0.95)
             TO ELECTRONS (USED FOR QUICK AND DIRTY INJECTION CALCULATIONS)
    INJ(1)   NEUTRAL BEAM INJECTION CALCULATION SWITCH       (1)
                = 0, NO INJECTION
                = 1, MOMENTS CALCULATION
                = 2, DEBUG CALCULATION
    INJ(2)   SWITCH FOR NUMBER AND TYPE OF NEUTRAL BEAM INJECTORS USED       (3)
                = 1, CO-INJECTION ONLY
                = 2, COUNTER-INJECTION ONLY
                = 3, TWO INJECTORS ONLY, ONE CO- AND ONE COUNTER-
                = 4, FOUR INJECTORS, TWO CO- AND TWO COUNTER-

                FOR THE FOLLOWING SWITCHES, 1 = ON AND 0 = OFF
    INJ(3)   SWITCH FOR CURRENT PERTURBATION FROM FAST IONS      (1)
    INJ(4)                 ELECTRIC FIELD EFFECTS ON FAST IONS         (0)
    INJ(5)                 LOSS CONE CORRECTION TO FAST IONS THERMALIZATION (0)
    INJ(6)   SWITCH FOR TYPE OF INJECTED GAS        (0)
               =0  HYDROGEN
               =1  DEUTERIUM
 &NBMIDA
 ENGY(1)=30000.0,DTCH=10.0,DTCGL=2.0,RC=75.0,RB=15.0,
 BHLF=7.5,TNJCT=20.0,DTNJCT=6.0,N1=500,N3=200,INJ=1,3,1,0,0,0,
 CURBM(1,1)=1.2,CURBM(1,2)=1.2,CURBM(2,1)=1.2,
 CURBM(2,2)=1.2,CURBM(3,1)=1.2,CURBM(3,2)=1.2,


 FQE=0.95
 &END
C
I / O    DATA    A N D    C O N T R O L    S W I T C H E S
    NR       NUMBER OF PRINT OUT SETS DURING SIMULATION OF PLASMA DISCHARGE
                (5)
 &IODATA
 NR=5
 &END
```

APPENDIX II

A. Names and Primary Functions of Labeled Common Blocks

B. Identification of Variables and Arrays in Labeled Common Blocks

A. Names and Primary Functions of Labeled Common Blocks

/BDYCØN/ - parameters associated with boundary conditions in initial values for the plasma.

/BEAMC/ - parameters associated with injected neutral beams, neutral beam current.

/BEAMP/ - quantities associated with neutral beam injection and slowing down in the plasma.

/CDCLBM/ - profiles of diffusion coefficients.

/CDCLBX/ - arrays generated by subroutine CDC for primary use in subroutine LBM.

/CN/ - items used for Gauss-Chebyshev integration.

/CURENT/ - parameters associated with the plasma current.

/D/ - neutral beam injection geometry for pencil beam calculations.

/E/ - injected neutral beam profile geometry data.

/ELCTRN/ - parameters and profiles associated with the plasma electrons.

/ERGBAL/ - profiles due to various processes contributing to the plasma energy balance.

/FIELDS/ - parameters and profiles associated with the electric and magnetic fields present in the plasma.

/FPPP/ - various mean free paths associated with neutral beam injection.

/FUDGE/ - neutral beam injection loss cone correction arrays.

/GAØRD/ - arguments used in list of calling parameters for Gauss-Legendre integration in neutral beam injection module.

/GEØM/ - parameters associated with the tokamak geometry.

/HIST/ - time histories of contributions to plasma power balance.

/HIST1/ - time integrals of components of plasma power for some selected processes and energy fluxes through the plasma surface for electrons and ions arising from thermal conduction and convection.

/HZ/ - intermediate values used in pencil beam geometry calculations.

/IBTMN/ - the solution vector for the set of differenced equations describing the model of the basic plasma.

/IMPURT/ - parameters and profiles associated with the impurities module.

/INDEX/ - switch arrays, number of spatial, time, and printout intervals in a simulation run.

/INPUT/ - input variables for Fokker-Planck moments calculations.

/INTRP/ - set of variables and arrays used for linear interpolation in the neutral beam injection module.

/INTRPS/ - parameters and switch governing type of interpolation used and switch for level of diagnostic printout for the neutral gas detailed slab model calculation.

/IØNLØS/ - profiles associated with ionization loss mechanisms.

/IØNS/ - parameters and profiles of quantities associated with dominant ions in the plasma (protons).

/JTH/ - total neutral gas attenuation coefficient profile array, EATN(101).

/LBMIBT/ - coefficient matrix storage array which is loaded in LBM and used by IBTM to generate the solution vectors for the set of finite differenced, coupled partial differential equations describing the basic plasma.

/MEASUR/ - quantities calculated specifically for making comparisons with measurements using microwave interferometer and any other device for which a special calculation must be made.

/NEUTPT/ - pointers used to partition the array WSC.

/NEUTRL/ - parameters and profiles associated with the neutral gas model module.

/NEWS/ - summary of selected information about the evolution of the plasma throughout the simulation.

/NUTRLP/ - parameters, profiles, and switches used in the neutral gas slab model calculations.

/ØUTPT/ - variables and profiles used primarily for purposes of providing output.

/ØUTPUT/ - output variables for Fokker-Planck moments calculations.

/PINT/ - profiles of power contributions due to the dominant modes involved in the power balance. These profiles are integrals from the plasma center out to the successive radial points in the spatial mesh.

/PØWER/ - profiles due to various processes contributing to the plasma power balance.

/PRNT/ - print control switches associated with the neutral gas module.

/QT/ - source term arrays for basic two-fluid model of the plasma.

/RADLØS/ - profiles of power loss through electromagnetic radiation.

/RATIØS/ - ratios of the various plasma power contributions to the ohmic heating power or total (ohmic + neutral beam injection) input power to the plasma and other assorted quantities.

/RATIØ1/ - ratios of the dominant plasma power contributions to the total (ohmic + neutral beam injection) input power to the plasma.

/SLAB/ - some variables used in the slab model treatment of neutral gas effects.

/SUMS/ - values of integrals obtained in the process of determining the plasma power balance.

/SURF/ - densities of various chemical elements at the plasma boundary.

/TA/ - injected neutral beam profiles and integration mesh size values.

/TEMP/ - ion, electron, and neutral gas temperature profiles, plasma ion velocity.

/THRMDF/ - profiles of thermal diffusion coefficients.

/TIME/ - time information for the simulation run exclusive of timing information required by the neutral beam injection module and the impurities module.

/TMINDX/ - timing information required by neutral beam injection module.

/TSTØR/ - ion and electron temperature profiles stored from the previous time step.

/XPØRT/ - parameters and profiles related to diffusion and thermal transport properties of the plasma.

/XTRA/ - some miscellaneous profiles needed in several subroutines.

/ZDFPS/ - diffusion coefficients for impurity diffusion calculations for the Pfirsch-Schlüter regime.

/ZRAD/ - high- and low-Z profiles of radiation loss due to impurities.

B. Identification of Variables and Arrays in Labeled Common Blocks

<u>CØMMØN /BDYCØN/</u>

DENB - electron density at the plasma boundary, cm^{-3}.

DENØ - initial value of electron density at the plasma center, cm^{-3}.

DNØ - at one time, density of particles added by the neutral beam, cm^{-3}. No longer used.

TEB - initial value of electron temperature at plasma center, eV.

TIB - initial value of ion temperature at plasma center, eV.

TEØ - temperature for electrons at the plasma boundary, eV.

TIØ - boundary and initial temperature of ions.

XI(10) - array of exponents for initial plasma profiles.
XI(1), of current density
XI(2), of electron temperature
XI(3), of ion temperature
XI(4), of electron density

CØMMØN /BEAMC/

CURBM(3,2) - array of values of injected neutral beam currents for the three energy components for co- and counter-injection, A.

ENGY(3) - array of energy values for the three beam components, eV. The value for the highest energy component is an input value. The half and third energies are calculated by alternate ENTRY BMSTRT.

DTCH - time increment for calculation of H(r) profile, msec.

DTCGL - time increment for calculation of GE, GI, and KE, msec.

RC [R7] - radial distance from torus major axis to point at which injected neutral beam axis is tangent to torus, cm.

RB [R8] - injected neutral beam cut-off radius r_B, cm.

BHLF [HALF] - neutral beam current half-width, assuming Gaussian profile, cm.

DJT(101) - current perturbation profile.

TNJCT - starting time for neutral beam injection, msec.

INJ(10) - neutral beam injection switch array.

INJ(1) [NJCT] - neutral beam injection calculation switch:
=0, no injection
=1, moments calculation
=2, debug (quick and dirty) calculation

INJ(2) [LJCT] - switch for number and type of neutral beam injectors used:
=1, co-injection only
=2, counter-injection only
=3, two injectors only, one co- and one counter-
=4, four injectors, two co- and two counter-

INJ(3) - switch for current perturbation from fast ions

INJ(4) - switch for electric field effects on fast ions

INJ(5) - switch for loss cone correction to fast ions thermalization

INJ(6) - switch for type of injected gas:
=0, hydrogen
=1, deuterium

CØMMØN /BEAMP/

Used primarily with the neutral beam injection module. The following four arrays contain profiles for the three energy components of the neutral beam and the two injection directions, co- and counter-.

HTMP(11,3,2) - H(r) profiles.

GETMP(11,3,2) - profiles of the fraction of beam power transferred to plasma electrons.

GITMP(11,3,2) - profiles of the fraction of beam power transferred to plasma ions.

DJTMP(11,3,2) - current perturbation profiles.

The following three arrays contain temporary profiles.

QETMP(11) - injected neutral beam heat input to electrons.

QITMP(11) - injected neutral beam heat input to ions.

DJTTMP(11) - current perturbation profile.

PATH(3) - no longer used. Originally had mean free path values for use in moments calculation.

TCGS(2) - pair of time values defining time interval over which H(r) profiles and results of moments calculation are interpolated. Array element TCGS(1) contains the larger time valuc.

SDT(3,4) - array of slowing-down times computed from the moments calculation for each beam energy component.

SDT(J,1) = TAVE, average lifetime for slowing down of fast ion

SDT(J,2) = TCXVØP, average charge-exchange lifetime

SDT(J,3) = TSØPP, initial slowing-down time of fast ions

SDT(J,4) = TFØPP, time for complete thermalization of fast ions

ECRS(3) = E_c, the critical energy, above which power is transferred from neutral beam to electrons and below which power is transferred predominantly to the ions.

CØMMØN /BEAMP/ (continued)

The following seven arrays contain values of quantities for the three energy components of the neutral beam and the two injection directions, co- and counter-.

FNBA(3,2) - fraction of injected neutral beam deposited in the plasma (neutral beam absorption factor).

QEB(3,2) - power deposited in the electrons.

QIB(3,2) - power deposited in the protons.

QCX(3,2) - power lost to charge exchange.

QTB(3,2) - power out of injector.

QTD(3,2) - power deposited in plasma.

QBL(3,2) - power deposited directly into the liner.

DIT(3,2) - injected current.

The following seven arrays contain values of quantities for the three energy components of the neutral beam for the sum of the two injection directions.

QTBT(3) - power out of the injector.

QTDT(3) - power deposited in the plasma.

QEET(3) - power deposited in the electrons.

QIIT(3) - power deposited in the protons.

QCXT(3) - power lost to charge exchange.

QBLT(3) - power deposited directly into the liner.

DITT(3) - injected current.

FPRT(5,3) - fractions of total injected neutral beam power deposited, for each energy component, into the following:

FPRT(1,J) - plasma

FPRT(2,J) - electrons

FPRT(3,J) - ions

CØMMØN /BEAMP/ (continued)

FPRT(4,J) - lost to charge exchange

FPRT(5,J) - deposited directly into the liner

DJSUM(11,2) - current density profiles, the sums of the profiles of the three energy components, for the co- and counter-injected beams.

PNUM - factor for doubling the number of injectors when computing the various contributions to the power balance.

TCHS - time at which H(r) profile is calculated; used for print-out.

XPRT(3) - stagnation distances x_s used in the H(r) calculation for the three beam energy components.

MM - subscript variable used with loss cone matrices to distinguish high- and low-Z impurity cases.
= 1 for low-Z
= 2 for high-Z, D1(1) > 10.0

C&#216;MM&#216;N /CDCLBM/

EE(101) = (B/χ_e) $(\partial\chi_e/\partial B)$, where χ_e is the electron thermal conductivity.

EI(101) = (B/χ_i) $(\partial\chi_i/\partial B)$, where χ_i is the ion thermal conductivity.

DIV(101) - no longer used.

The following seven arrays are reciprocals of the values of the profiles listed.

RB(101) = B^{-1}, B is the poloidal magnetic field.

RJ(101) = j^{-1}, j is the current density.

RTE(101) = T_e^{-1}, T_e is the plasma electron temperature.

RTI(101) = T_i^{-1}, T_i is the plasma ion temperature.

RTE12(101) = $T_e^{-\frac{1}{2}}$.

RTE32(101) = $T_e^{-3/2}$.

RTE52(101) = $T_e^{-5/2}$.

TDP(101) = D, the particle diffusion coefficient.

WE(101) = (T_e/χ_e) $(\partial\chi_e/\partial T_e)$.

WI(101) = (T_i/χ_i) $(\partial\chi_i/\partial T_i)$.

CØMMØN /CDCLBX

WPT(101) = (T_e/D) $(\partial D/\partial T_e)$, where D is the particle diffusion coefficient.

WPB(101) = (B/D) $(\partial D/\partial B)$, where B is the poloidal magnetic field.

WPN(101) = (N/D) $(\partial D/\partial N)$, where N is the particle density.

ETA(101) = η, the plasma resistivity.

WGE(101) = (N_e/χ_e) $(\partial \chi_e/\partial N_e)$, where N_e is the electron density and χ_e the electron thermal conductivity.

WGI(101) = (N_p/χ_i) $(\partial \chi_i/\partial N_p)$, where N_p is the proton density and χ_i the ion thermal conductivity.

CØMMØN /CN/

S(15) - array of zeros of the Chebyshev polynomials of degree NP.

S1 - Christoffel numbers β_j for the Gauss-Chebyshev formula, the coefficient π/n for the sum on the right-hand side (see description of subroutine RPT3XY in the neutral beam injection module).

NP - the number of points used by the quadrature formula.

CØMMØN /CURENT/

ZFI - additional amount of total current (A) which is added with a characteristic rise time of TC msec.

ZPC - initial amount of total current (A). This component remains constant throughout the simulated plasma discharge.

C∅MM∅N /CXL∅S/

QCX(101) - profile of energy loss to ions due to charge exchange.

PCX(101) - profile of power loss to ions due to charge exchange.

CØMMØN /D/

A - minor radius of torus, cm.

XS - displacement of stagnation point from the torus minor axis, cm.

RB - injected neutral beam radius r_B, cm.

RHØ - radial distance to a point in the plasma measured from the stagnation point x_s, cm.

RO - major radius of the torus R_0, cm.

RC - radial distance from torus major axis to point at which injected neutral beam axis is tangent to the torus, cm.

ØFP - reciprocal of the charge-exchange mean free path, cm^{-1}.

AI - does not appear to be used anywhere.

NZ - switch to determine which integral in the expression for $H(\rho,\Theta)$ is to be evaluated.

C∅MM∅N /E/

CRB - current value of R_B, required by a nested integral, cm.

A5 (A6) - not currently used.

XUP1 - a variable containing one of the limits for the integrals in the pencil beam calculations for neutral beam injection. It has one of the values $R_o + x_s \pm \sqrt{\rho^2 - z_B^2}$ according to whether NZ = 1, 2.

BEAM - square of the half width of the assumed Gaussian profile shape of the injected neutral beam.

BSHAPE - profile factor for injected neutral beam.

DXX - spacing interval for the XXX array.

CØMMØN /ELCTRN/

FQE - fraction of injected neutral beam power transferred to electrons. Used only for debug (quick and dirty) calculation in subroutine BEAM.

CØMMØN /ERGBAL/

EB(101) - power density profile due to electron convection.

EEØ(101) - power density profile of electrons due to neutral beam injection.

EIØ(101) - power density profile of ions due to electron-ion transfer.

ETE(101) - power density profile due to electron thermal conduction.

ETI(101) - power density profile due to ion thermal conduction.

CØMMØN /FIELDS/

B(101)	- poloidal magnetic field profile.
E(101)	- toroidal electric field profile.
BZ(101)	- electron density profile from the previous time step, cm^{-3}.
EZ(101)	- does not appear to be used anywhere.
QS(101)	- q, safety factor profile.
ZJ(101)	- current density profile, A/cm^2.
ZJE(101)	- does not appear to be used anywhere.
BT	- toroidal magnetic field, G.
BTHE	- β_p, poloidal beta.
ELI	- toroidal plasma inductance, H.
VAX	- axial emf at the surface of the torus.
ZJØ	- initial value of current density at plasma center, A/cm^2.

C∅MM∅N /FPPP/

FPCX - mean free path for charge exchange for each neutral beam energy component, cm^{-1}.

FPI - mean free path for impact ionization for protons for each beam energy component, cm^{-1}.

FPE - mean free path for impact ionization for electrons for each beam energy component, cm^{-1}.

FP - this array is generated in subroutine SUBH, but appears to serve no purpose.

FPIMP - this array also appears to serve no useful purpose.

NCGLM - number of computational (spatial) grid elements.

C∅MM∅N /FUDGE/

Neutral beam injection loss cone correction arrays.

ST1(10,4,2,2) - for power transferred to plasma electrons, no correction.

ST2(10,4,2,2) - for power transferred to plasma ions, no correction.

ST3(10,4,2,2) - for KE, no corrections.

ST4(10,4,2,2) - for power transferred to plasma electrons, with corrections.

ST5(10,4,2,2) - for power transferred to plasma ions, with corrections.

ST6(10,4,2,2) - for KE, with corrections.

C∅MM∅N /GA∅RD/

NN - the number of equal subintervals that [XL,XU] is to be divided into. The lower limit of integration is XL, the upper limit XU.

MM - the number of points (order) at which the integrand is to be evaluated in each subinterval, i.e. total number of evaluations = NN * MM. Permissible values of MM are 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, 20, 24, 32, 40, 48, 64, 80, and 96. If MM < 2, 2 points are used. Within the range of 2-96, if MM is not one of the permissible values, the next higher permissible value is used.

CØMMØN /GEØM/

AM	- torus minor radius, cm.
AREA	- total surface area of torus, cm^2.
HR	- radial element of length, cm.
RØ	- torus major radius, cm.
R9	- plasma minor radius, cm.
VØL	- plasma (torus) volume, cm^3.
ZB	- no longer used.

CØMMØN /HIST/

B1(101) - time history of energy added to plasma through ohmic heating.

B2(101) - time history of energy loss from plasma by electron diffusion (particle plus heat).

B3(101) - time history of energy loss from plasma by ion diffusion (particle plus heat).

B4(101) - time history of energy loss from plasma by line radiation.

B5(101) - time history of energy loss from plasma by charge exchange.

B6(101) - time history of energy stored in the electrons.

B7(101) - time history of energy stored in the ions.

B8(101) - time history of energy added to electrons from neutral beam injection.

B9(101) - time history of energy added to ions from neutral beam injection.

C∅MM∅N /HIST1/

P∅X - total amount of energy added to plasma by ohmic heating.

PEX - energy loss from plasma due to electron thermal conduction and convection.

PIX - energy loss from plasma due to ion thermal conduction and convection.

PLRX - energy loss from plasma due to line radiation.

PCXX - energy loss from plasma due to charge exchange.

SUMEX - plasma energy stored in the electrons.

SUMIX - plasma energy stored in the ions.

ICM - index variable that is incremented for each new value of time for which a record of the plasma energy balance is desired, used to generate the time history arrays B1-B9.

PHE - energy added to plasma electrons from neutral beam injection.

PHI - energy added to plasma ions from neutral beam injection.

CØMMØN /HZ/

XZB,(ZB) - a particular value of z_B, used in a nested integration, cm.

CØMMØN /IBTMN/

X(360) - the solution vector of the plasma profiles, determined by the system of differenced partial differential equations appearing in the plasma module.

CØMMØN /IMPURT/

CARBØN(101) - total density profile of carbon impurity species.

ØXYGEN(101) - total density profile of oxygen impurity species.

CC(7,101) - CC(K,I) is the density of the (K-1)th charge state of carbon at the Ith grid point.

CØ(9,101) - CØ(K,I) is the density of the (K-1)th charge state of oxygen at the Ith grid point.

PCIMP - percentage of electron density due to a high-Z impurity. For IMP(2) = 1 or 3, PCIMP is due to some average effective high-Z impurity; for IMP(2) = 2 or 4, it is due to iron impurities.

XØX - percentage of electron density due to oxygen impurities.

XCAR - percentage of electron density due to carbon impurities.

ZEFF - effective charge of ions.

ZIMP - maximum charge of the high-Z impurity ion.

RM1 - no longer used.

RM2 - no longer used.

DENP(101) - plasma proton (ion) density profile, cm^{-3}.

ZBR(101) - profile of the quantity [Z] defined to be

$$[Z] = \frac{\sum_{\text{species},k} \frac{m}{m_k}\text{proton} \quad n_k Z_k^2}{\sum_{\text{species},\ k} \quad n_k Z_k}.$$

EIMP(101) - power loss profile due to presence of carbon and oxygen impurities.

CØMMØN /INDEX/

TIC - basic time interval used for generating history arrays in CØMMØN /HIST/.

N - number of spatial grid points used for radial direction throughout the simulation code.

NT - total number of time intervals for current pulse and plasma simulation.

NR - number of times sets of simulated quantities obtained during execution of the program are printed.

NM1 = N - 1, the number of spatial grid intervals in the radial direction.

NUTE(10) - switches used to select a simulation path for the neutral gas effects modeling.

NUTE(1), switch for type of computation to be used for the neutral gas simulation:

= 1, analytic model
= 2, slab model (Hogan's model, without reflection)
= 3, for slab model with wall reflections (not yet included)
= 4, Monte Carlo transport (NUTRLSN, not yet included)

NUTE(2), exponent in power law for analytic model

IMP(10) - switches used to select a simulation path for modeling impurity atom effects.

IMP(1), switch for initial impurity stripping calculation:

= 1, corona equilibrium table for carbon, oxygen
= 2, dynamic rate calculation for carbon, oxygen (not an option in current version of the program)

IMP(2), switch for diffusion of impurities:

= 1, a corona equilibrium table and radial distribution formulae are used for carbon and oxygen. High-Z impurities are treated using a crude estimate
= 2, corona equilibrium tables are used for carbon, oxygen, and high-Z impurity (Fe) atom charge states. Radial distributions are assigned using formulae

CØMMØN /INDEX/ (continued)

= 3, a corona equilibrium table together with radial distributions generated by Pfirsch-Schlüter diffusion is used for carbon and oxygen. High-Z impurities are treated using a crude estimate

= 4, corona equilibrium tables are used for carbon, oxygen, and high-Z impurity (Fe) atom charge states. Radial distributions are generated by a Pfirsch-Schlüter diffusion model

IMP(3), switch for impurity model:

= 1, use fixed values
= 2, use model calculation determined by choice of switch value for IMP(2)

IMP(4), switch for detailed model for iron:

= 0, include iron coronal equilibria
= 1, omit detailed description

INIT(10) - switches for initiating various options available in the program.

INIT(1), switch for starting mode for initial plasma variable profiles:

= 0, for profiles generated by analytic functions
= 1, for profiles read from data cards

INIT(2), not currently used

INIT(3), switch for examining MHD stability characteristics of the plasma:

= 0, determines the location of the singular surface for m = 3,5
= 1, calculates the radial eigenfunctions and tearing mode growth rate according to Furth, Rutherford, and Selberg, for m = 3,5. Can be used to evaluate transport coefficients in subroutine CDC

CØMMØN /INPUT/

/INPUT/ is used for the moments calculation.

T - elapsed time measured from the time of neutral beam turn-on to the times when the quantities G_e, G_i, and ΔJ are computed.

EO - variable used for temporary storage of one of the three neutral beam energy components. Used in the moments calculation.

AMF - atomic mass of injected fast ions.

AMI - atomic mass of plasma ions.

The following six variables are used for temporary storage of quantities at each of the 11 points of the calculational grid elements of the moments calculation.

TEE - electron temperature, eV.

TII - ion temperature, eV.

DNE - electron density, normalized to 10^{13} cm^{-3}.

DNO - total neutral density, normalized to 10^8 cm^{-3}.

BRAZ = $\langle Z \rangle$ or Z_{eff}.

SQBRZ = $[\bar{Z}]$.

ZF - fast ion charge.

ETO = cos Θ, the cosine of the neutral beam injection angle.

The following three variables contain values of the upper bounds on error tolerance required by the Simpson's rule integration subroutines used in the moments calculation.

EPA3 - upper bound for ASIMP1.

EPA4 - upper bound for ASIMP2.

EPA5 - upper bound for ASIMP3.

C∅MM∅N /INPUT/ (continued)

The following three variables contain values of the convergence test selector required by the Simpson's rule integration subroutines.

M3 - convergence test selector for ASIMP1.

M4 - convergence test selector for ASIMP2.

M5 - convergence test selector for ASIMP3.

ICTX - switch variable, used to choose between analytic expression or numerical evaluation of an integral; $\neq 0$, if τ_{CX} is constant.

INO - switch variable, used to skip electric field perturbation corrections (= 1).

C∅MM∅N /INTRP/

XXX(101) - independent variable array corresponding to the normalized density profile.

PIDEN(101) - plasma density profile array, normalized to the central density (see description of subroutine SUBH).

T∅L - tolerance factor for test of error size for interpolation procedure.

RTMP(15) - array of normalized (to the maximum radial value associated with the computational grid elements of the H(r) calculation in the neutral beam injection module) radial coordinate values.

CØMMØN /INTRPS/

INTRPS - switch for selecting type of interpolation for approximating NI, TI, and EATN.

= 1, for linear
= 2, for quadratic
= 3, for cubic

NSMO - number of division points in the two partition intervals which contain the point of the logarithmic singularity.

NSMO21 = 2 * NSMO-1.

IDEBUG - diagnostic print switch.

= 0, for no information
= 1, prints A = P1, B = P2, C = AFC, Lipschitz constant, estimate of the number of iterations
= 2, prints everything listed above plus the profiles NI, TI (or AI), and EATN

CØMMØN /IØNLØS/

QEE(101) - electron power loss due to ionization of neutral hydrogen by electron impact.

QII(101) - proton power loss due to ionization of neutral hydrogen by electron impact.

FF(101) - rate for ionization of neutral hydrogen by electron impact.

SIØ(101) - ionization cross section profile.

CØMMØN /IØNS/

DEN(101) - plasma electron density profile, cm^{-3}.

DSØ(101) - profile is no longer used.

TES(101) - electron density profile from the previous simulation time step.

AMU - mass of plasma ions, amu.

C∅MM∅N /JTH/

EATN(101) - total neutral gas attenuation coefficient profile.

C∅MM∅N /LBMIBT/

A(8008) - matrix of coefficients of set of coupled, differenced partial differential equations describing the plasma evolution in the plasma module.

CØMMØN /MEASUR/

FRING - number of microwave fringe shifts with respect to the vacuum due to the presence of plasma electrons.

RVD - ratio of plasma drift velocity to sound velocity. This variable is not currently used.

XFRI - value of the integral of the electron density profile.

CØMMØN /NEUTPT/

INIDSQ - reference location of the scratch array used in the neutral gas slab model calculations.

IØDSQT = INIDSQ + NØZ, where NØZ is the number of z nodes (101) distributed uniformly in the interval (-1,1).

IQA = IØDSQT + NØX, where NØX is the number of x nodes (21) in the interval (0,1) at which to approximate the neutral density.

IN1LST = IQA + NØZ * NØX.

IF = IN1LST + NØX.

IN1T = IF + NØZ.

INOT = IN1T + NØZ.

INL = INOT + NØZ.

INH = INL + NØX.

ISTPLO = INH + NØX.

ISTPRO = ISTPLO + NØX.

INISTG = ISTPRO + NØX.

INOZA = INISTG + NCØN, where NCØN = (2 * NSMO-1) * NØX and NSMO is the number of integration nodes in the interval (P1,X) or (X,P2). [See Sect. 4.1 for detailed explanation of the meaning of these integration intervals.]

IT1D2 = INOZA + NCØN.

IQLNQ = IT1D2 + NCØN.

IZO = IQLNQ + NCØN.

INOPN1 = IZO + NCØN.

IINTGM = INOPN1 + 2 * NSMO-1.

IZL = IINTGM + 2 * NSMO-2.

IZH = IZL + NØX.

IHLF = IZH + NØX.

IGLN = IHLF + NØX.

C∅MM∅N /NEUTPT/ (continued)

INOL = IGLN + NSPN∅X, where NSPN∅X = NSIMP * N∅X.

IHRF = INOL + NSPN∅X.

IGRN = IHRF + N∅X.

INOR = IGRN + NSPN∅X.

IFZ = INOR + NSPN∅X.

IRATZL = IFZ + NSIMP, where NSIMP is the number of integration nodes in the interval (R1, P1) or in (P2, R2). [See Sect. 4.1 for a detailed explanation of the meaning of these integration intervals.]

IRATZH = IRATZL + NSPN∅X.

IIZXL = IRATZH + NSPN∅X.

IIZXH = IIZXL + NSPN∅X.

CØMMØN /NEUTRL/

QE1(101) - injected neutral beam heat input to electrons.

QI1(101) - injected neutral beam heat input to ions.

SIGV(101) - ratio of particle fluxes in radial direction computed two different ways to test consistency of the simulation. If computations are consistent, the profile values should be unity.

ZNØ(101) - cold neutral density profile. This array is not currently used.

ZN1(101) - total neutral density profile, cm^{-3}.

ATT - input parameter used in analytic model for neutral gas density (currently calculated explicitly).

DNI - no longer used.

FCDEN - initial Franck-Condon boundary density, cm^{-3}.

FCDEX - Franck-Condon boundary density at time TIM, cm^{-3}.

FCF - final boundary density of neutrals, cm^{-3}.

FQI - fraction of injected neutral beam power transferred to ions (no longer used).

PINJ - no longer used.

PINJQ - no longer used.

C∅MM∅N /NEWS/

RNEWS(20,10) - array of values of plasma central electron and ion temperature, electron and proton densities, fast neutral density, effective charge, conduction energy and gross energy confinement times, and safety factor. The safety factor at the plasma edge is also included. The number 1.E 11 or the minimum values of ν_e* and ν_i*, whichever is smaller, are also stored. All these quantities, together with the value of the elapsed simulation time, are stored at each of 10 times during the simulation when plasma profiles and power balances are printed. This summary array is then printed at the end of the simulation.

NEWS - a counter variable for the number of times summary information is stored in the array RNEWS.

CØMMØN /NUTRLP/

NØZ - number of uniformly distributed points in the interval [-1,1] used to approximate hot neutral gas density.

NØX - number of points in the interval [0,1] at which to approximate hot neutral gas density in transport calculation.

NSIMP - number of Simpson nodes used to approximate an integral near a logarithmic singularity in G(q).

NUNIF - switch for the nodal points in the interval [0,1].

=0, points are defined in subroutine NEUTIO to be (I-1)/(NØX-1)
>0, points should be placed at the beginning of the array W1 in CØMMØN /NUTRLP/

QF,QC - determine an integration interval on each side of the logarithmic singularity of G(q) at each nodal point in the interval [0,1]. Require $0 \leq QF < QC$. Whenever $QF \geqslant QC$, NEUTIO sets QF = 0.1*QC.

DZ = 2/(NØZ-1).

DZD2 = 1/(NØZ-1).

DZD3 = 2/(3*(NØZ-1)).

EPSN1 - convergence criterion for hot neutral gas density solution to the discrete integral equation.

SIMPM1 = NSIMP-1.

BDSQTX or INO - not currently used.

TCINT - C * (integral from -1 to 1 of EATN), where

$$C = (1.13a\sqrt{T_i(0)})/(\sigma_{cx}N_i(0)\sqrt{E\emptyset})$$

or TIO = $T_i(0)$, the ion temperature at the plasma center.

NØXM1 = NØX-1.

NØXP1 = NØX + 1.

NØZP1 = NØZ + 1.

NSC - used as a switch in subroutine NEUTC to bypass some already tabulated values of the convergence function.

CØMMØN/NUTRLP/ (continued)

ININ = 0, pointer for the normalized ion density profile.

ITIN = ININ+N, pointer for the normalized ion temperature profile.

IN1 = ITIN+N, pointer for the hot neutral density profile.

IEATN = IN1+NØX, pointer for the estimate of the total neutral gas attenuation coefficient profile.

IINIOX = IINIX1+N, pointer for array of values of the integral

$$\int_0^{y_I} dZ \; EATN(Z)$$

for I = 1, ..., N and $y_I = (I-1)/(N-1)$.

IINIX1 = IEATN+N, pointer for array of values of the integral

$$\int_{y_I}^{1} dZ \; EATN(Z)$$

for I = 1, ..., N and $y_I = (I-1)/(N-1)$.

IX = IINIOX+N, pointer for array of nodes in [0,1] at which to approximate the neutral density.

IZ = IX+NØX, pointer for array of nodes uniformly distributed in [-1,1].

IJL = IZ+NØZ, pointer for the JL array.

IJH = IJL+NØX, pointer for the JH array.

IIJ = IJH+NØX, pointer for the IJ array used in the linear interpolation of the total neutral density at the nodes Z(J).

IRAT = IIJ+NØZ, pointer for the RAT array used in the linear interpolation of the total neutral density N_1 at the nodes Z(J).

NWDW1 - number of words in the array W1 of the common block /NUTRLP/.

MWDSC - number of words in array A in common block /LBMIBT/.

N - the number of nodal points in the plasma spatial net (see also common block /INDEX/).

ANM1 = FLØAT (N-1).

DSP = 1./ANM1.

A = 1.13 * AM * $\sigma_{cx}(o)$ $N_i(o)$.

B = A.

C = A * $\sqrt{T_i(o)/E\emptyset}$.

W1(1300) - a partitioned array containing the following elements:

W1(ININ+K) = NI(K), K = 1, ..., N: normalized ion density
W1(ITIN+K) = TI(K), K = 1, ..., N: normalized ion temperature
W1(IN1+I) = N1(I) = N1(X(I)), I = 1, ..., NØX
W1(IEATN+K) = EATN profile approximated at the K-th nodal point, K = 1, ..., N

C&ØMMØN /NUTRLP/ (continued)

W1(IINIX1+K) = integral from (K-1)/(N-1) to 1 of EATN,
K = 1, ..., N
W1(IINIOX+K) = integral from 0 to (K-1)/(N-1) of EATN,
K = 1, ..., N
W1(IX+I) = X(I), I = 1, ..., NØX
W1(IZ+J) = Z(J), J = 1, ..., NØZ
W1(IJL+I) = JL(I), I = 1, ..., NØX
W1(IJH+I) = JH(I), I = 1, ..., NØX
W1(IIJ+J) = IJ(J), J = 1, ..., NØZ
W1(IRAT+J) = RAT(J), J = 1, ..., NØZ

RAT(J) and IJ(J) are used in the linear interpolation of N1 at Z(J):

$N1\ (Z(J)) = N1(L) + RAT(J) * (N1(L+1) - N1(L))$
where $L = IJ(J)$
$X(L) < |Z(J)| \leqslant X(L+1)$
and $RAT(J) = (|Z(J)| - X(L))/(X(L+1) - X(L))$

CØMMØN/ØUTPT/

CTE(101) - stored energy, mJ/cm^3.

TAUE(101) - conduction energy confinement time profile, msec.

GAMIN - incoming neutral flux.

GAMØUT - outgoing flux of charged particles (= $N_p V$).

PLR - component of plasma power associated with line radiation.

CØMMØN/ØUTPUT/

EOØ - E_o, same as EO in common /INPUT/, a variable used for temporary storage of one of the three neutral beam energy components.

ECØ - E_c, the energy above which injected neutral beam slowing is due to the plasma electrons and below which slowing is due to plasma ions, eV.

ETØ - energy to which neutral beam particles have slowed after time T following neutral beam turn-on, eV.

TØØ - same as T in common /INPUT/, elapsed time measured from the time of neutral beam turn-on to the times when the quantities G_e, G_i, and ΔJ are computed, sec.

TSØ - τ_s, neutral beam initial slowing down time, sec.

TFØ - τ_f, time interval criterion used to determine when the Fokker-Planck code solution has reached a steady state; also, time for complete thermalization of the fast ions.

TCXVØ - $\tau_{cx}(v_o)$, mean lifetime of a neutral beam particle before undergoing charge exchange with a plasma ion.

VOØ = $\sqrt{2*EO\emptyset/AMF}$, entrance velocity of neutral beam particles into plasma for each of the three beam energy components.

VCØ - v_c, the critical velocity of the neutral beam particles with energy E_c.

VTØ - velocity to which neutral beam particles have slowed after time T following neutral beam turn-on.

GIT - fraction of injected beam power transferred to the plasma ions.

GEE - fraction of injected beam power transferred to the plasma electrons.

BKI - momentum transferred from the injected beam to the plasma ions.

BKE - momentum transferred from the injected beam to the plasma electrons.

BN - number of stored fast ions resulting from the injected neutral beam.

PCXVT - probability that a charge-exchange event will not occur during the time a neutral beam particle slows from the injection velocity VOØ to the velocity VTØ.

PCXO - variable is not currently being used.

C∅MM∅N/∅UTPUT/

The following seven variables are corrections due to the electric field:

GIE - to fraction of injected neutral beam power transferred to the plasma ions.

GEEE - to fraction of injected neutral beam power transferred to the plasma electrons.

BKIE - to momentum transferred from injected beam to the plasma ions.

BKEE - to momentum transferred from injected beam to the plasma electrons.

BME - to the value of an integral occurring in the moments treatment of the Fokker-Planck equation.

BNE - to the number of stored fast ions.

C∅MM∅N/PINT/

This common block contains the following time histories generated by subroutine ∅RMPWR.

C1(101) - ohmic heating power flow throughout the plasma.

C2(101) - electron thermal conduction flux.

C3(101) - electron convection flux.

C4(101) - electron radiation loss (line, recombination, and synchrotron) power flow throughout the plasma.

C5(101) - power flow due to electron-ion transfer for electrons.

C6(101) - ion thermal conduction flux.

C7(101) - ion convection flux.

C8(101) - power flow associated with energy loss due to charge exchange.

C9(101) - power flow associated with electron energy due to neutral beam injection.

C10(101) - power flow associated with ion energy due to neutral beam injection.

CØMMØN/PØWER/

PB(101) - power density profile due to ion convection.

PEØ(101) - power density profile for ionization loss due to electrons.

PIN(101) - power density profile due to ohmic heating.

PIØ(101) - power density profile for ionization loss due to charge exchange.

EDØT - time derivative of electron energy, power in plasma electrons.

PED - electron thermal conduction flux.

PEI - component of plasma power due to electron-ion transfer.

PEIL - component of plasma power arising from ionization loss due to electrons.

PID - ion thermal conduction flux.

PIE - component of plasma power due to ion-electron transfer.

PIIL - component of plasma power arising from ionization loss due to charge exchange.

PØH - component of plasma power due to ohmic heating.

COMMON/PRNT/

IPA - not currently used.

IPNO - applicable only if a new slab calculation is to be made

= 0, for no print
= 1, prints range of relative error curve (see EPSNO)
= 2, prints same as = 1 plus old and new NI/AI and neutral density

CØMMØN/QT/

QE(101)	- power gain to the plasma electrons (source term).
QI(101)	- power gain to the plasma ions (source term).

CØMMØN /RADLØS/

A1(101) - no longer used.

A2(101) - no longer used.

A3(101) - rate of change of plasma electron density profile, $cm^{-3} \cdot sec^{-1}$.

SHR(101) - power density profile for synchrotron-bremsstrahlung radiation, kW/cm^3.

BREM - not currently used.

SYN - not currently used.

CØMMØN /RATIØS

PCVE - power flux from electron convection.

PCVI - power flux from ion convection.

RCUR - ratio of the plasma current computed by integrating the current density to the program current computed by function PC.

RCVE - ratio of power transported by electron convection to the total power input.

RCVI - same as RCVE, except for ions.

TAX - gross electron confinement time, msec.

TAXZ - gross ion confinement time, msec.

TAXI - confinement time characteristic of plasma transport and radiation, msec.

CØMMØN /RATIØ1/

This common block contains ratios of power transported by various processes to the total power input (from ohmic heating plus neutral beam injection).

REI	- electron-ion transfer.
RTH	- total thermal conduction.
RPLR	- line radiation.
RDØT	- electron heating or cooling.
RTHE	- electron thermal conduction.
RTHI	- ion (proton) convection.

CØMMØN /SLAB/

EØ - energy of neutrals coming from the wall, eV.

EPSNO - an error upper bound for the magnitude of the relative error between the current value of (A*NI)/AI and the value used previously to get the last basic neutral profile. If this bound is exceeded, then a new slab calculation will be made.

NWDWSC - the number of words in the A array in common block /LBMIBT/.

CØMMØN /SUMS/

SUMBT - energy density associated with the poloidal magnetic field.

SUMC - plasma current obtained by integrating the current density profile.

SUMEI - energy density associated with electron-ion collisions.

SUMLR - energy density due to line radiation.

SUMØ - energy density associated with ohmic heating.

SUMP - energy density associated with electrons.

SUMQE - ionization loss due to electrons.

SUMQI - ionization loss due to charge exchange.

ZE - internal energy of plasma electrons.

ZSI - internal energy of plasma ions.

SNJE - energy in electrons due to neutral beam injection.

SNJI - energy in ions due to neutral beam injection.

C∅MM∅N /SURF/

CARN - density of carbon atoms at the edge of the plasma, cm^{-3}.

FENS - density of iron atoms at the edge of the plasma, cm^{-3}.

ISR∅N - logic control variable, not currently used.

CØMMØN /TA/

HRHØ(11) - array of values of $H(r,\theta)$, the fast ion production profile.

RV(11) - array of radial values at which the fast ion production profile, $H(r,\theta)$ before angle averaging, is determined, i.e., $0 \leqslant RV(I) < RHØMAX$.

HTT(11) - value of the integral $\langle H(r,\theta)\rangle_\theta$ at each of the points in the interval [0,AM] with grid spacing DELN.

RHØMAX - maximum value of RHØ permitted, as determined by the position of the torus wall or diverter relative to the stagnation point.

DELH = RHØMAX/NCGLM, radial increment with respect to maximum value of RHØ, cm.

DELN = AM/NCGLM, radial increment with respect to torus minor radius AM, cm.

NGLP1 - number of computational grid elements plus one, used for the $H(\rho(\theta))$ profile.

CØMMØN /TEMP/

TE(101) - plasma electron temperature profile, eV.

TI(101) - plasma ion temperature profile, eV.

TØ(101) - neutral gas temperature profile, eV.

VEL(101) - plasma ion drift velocity, cm/sec.

COMMON /THRMDF/

TDE(101) - the (pseudoclassical) electron thermal diffusion coefficient.

TDI(101) - the (neoclassical) ion thermal diffusion coefficient.

CØMMØN /TIME/

HT - current increment for simulation time step, msec.

TC - rise time of the plasma current, msec.

TSTAR - simulation starting time, msec.

TIM - current value of elapsed simulation time, msec.

TMAX - maximum duration of plasma current pulse, msec.

TNJ - not currently used.

TPØ - falloff time of external neutral density, msec.

TPR - time interval between successive detailed printouts of plasma variable profiles and power balances, msec.

C∅MM∅N /TMINDX/

DTNJCT - neutral beam injection current rise time, msec.

N1 - number of computational time steps in plasma current rise time interval.

N3 - number of computational time steps in injected neutral beam current rise time interval.

C∅MM∅N /TST∅R/

TEL(101) - profile of electron temperature stored from previous time interval.

TIL(101) - profile of ion temperature stored from previous time interval.

CØMMØN /XPØRT/

D1(101) = Z_{eff}, profile of plasma effective charge.

D6(101) = ν_i^*, ratio of the ion collision frequency to the ion bounce frequency.

D11(10) - array of factors for scaling plasma transport characteristics.

D11(1), electron thermal transport
D11(2), ion thermal transport
D11(3), particle diffusion
D11(4), resistivity
D11(5), threshold of trapped electron scaling (NUE*)
D11(6), threshold of trapped ion scaling (NUI*)

C∅MM∅N /XTRA/

ALF(101) = ν_e^*, ratio of the electron collision frequency to the electron bounce frequency.

F(101) - τ_p, particle confinement time profile, msec.

TE12(101) = $\sqrt{T_e}$, square root of values of the electron temperature profile.

CØMMØN /ZDFPS/

TDZZ(101) - impurity species diffusion coefficient.

ZGRAD(101) - a collection of ion temperature and density gradient terms.

CØMMØN /ZRAD/

QRD(101) - low-Z impurity radiation contribution to power loss, W/cm^3.

HIZ(101) - high-Z impurity radiation contribution to power loss, W/cm^3.

DHI(101) - not currently used.

ZHI(101) - not currently used.

APPENDIX III

Appendix III presents some examples to show how the matrix elements in subroutine LBM are obtained. The T_e and E equations (numbers 1 and 5) will be used in the examples.

Consider the electron thermal conduction term in the T_e equation evaluated at the time step $N + \frac{1}{2}$,

$$\frac{1}{rN_e}\frac{\partial}{\partial r}\left(rN_e\chi_e\frac{\partial T_e}{\partial r}\right)\Bigg]^{N+\frac{1}{2}} = \chi_e\frac{\partial^2 T_e}{\partial r^2}\Bigg]^{N+\frac{1}{2}} + \frac{1}{rN_e}\frac{\partial}{\partial r}(rN_e\chi_e)\frac{\partial T_e}{\partial r}\Bigg]^{N+\frac{1}{2}} .$$

Now let

$$\chi\frac{\partial^2 r}{\partial r^2} = \frac{\chi}{(\Delta r)^2}\Big[T_{I+1} - 2T_I + T_{I-1} = \frac{1}{(\Delta r)^2}\delta_{1I}$$

and

$$\frac{1}{rN_e}\frac{\partial}{\partial r}(rN_e\chi_e)\frac{\partial T_e}{\partial r} = \frac{1}{rN_e}\left[\frac{y_{I+1} - y_{I-1}}{2\Delta r}\right]\left[\frac{T_{I+1} - T_{I-1}}{2\Delta r}\right]$$

where $y = rN_e\chi_e$ and

$$\delta_{2I} = \frac{1}{4rN_e}\Big[y_{I+1} - y_{I-1}\Big]\Big[T_{I+1} - T_{I-1}\Big] .$$

The difference representation above can be put in the form of a linear combination of B^{N+1} and T^{N+1} if the expressions δ^{N+1}_{1T} and δ^{N+1}_{2T} are linearized in time:

$$\delta^{N+1}_{1I} = \delta^{N}_{1I} + \frac{\partial\delta^{N}_{1I}}{\partial t}\Delta t + \theta[(\Delta t)^2]$$

$$\Delta t\frac{\partial\delta^{N}_{1I}}{\partial t} = \Delta t\frac{\partial}{\partial t}\Big[\chi_I(T_{I+1}-2T_I+T_{I-1})\Big]$$

$$= \Delta t\left[\chi_I\frac{\partial}{\partial t}(T_{I+1}-2T_I+T_{I-1}) + (T_{I+1}-2T_I+T_{I-1})\left(\frac{\partial\chi_I}{\partial B_I}\frac{\partial B_T}{\partial t} + \frac{\partial B_I}{\partial T_I}\frac{\partial T_I}{\partial t}\right)\right] .$$

Now $\chi = A\ B^{\theta}\ T^{\omega}$ so that $\frac{\partial\chi}{\partial B} = \frac{\theta}{B}\chi$ and $\frac{\partial\chi}{\partial T} = \frac{\omega}{T}\chi$.

Finally, using $\frac{\partial h}{\partial t} = \frac{1}{\Delta t}\Big[h^{N+1} - h^{N}\Big]$ gives

$$\Delta t \frac{\partial \delta_{1I}^{N}}{\partial t} = \chi_I \left\{ T_{I+1}^{N+1} - 2T_{I}^{N+1} + T_{I-1}^{N+1} + \left(T_{I+1}^{N} - 2T_{I}^{N} + T_{I-1}^{N}\right) \left[\frac{\theta_I}{B_I^N} \left(B_I^{N+1} - B_I^N\right) + \frac{\omega_T}{T_I^N} \left(T_I^{N+1} - T_I^N\right) - 1 \right] \right\} .$$

Similarly,

$$\Delta t \frac{\partial \delta_{2I}^{N}}{\partial t} = \frac{1}{4 r_I N_I} \left\{ r_{I+1} N_{I+1} \chi_{I+1} \left[\frac{\theta_{I+1}}{B_{I+1}} \left(B_{I+1}^{N+1} - B_{I+1}^{N}\right) + \frac{\omega_{I+1}}{T_{I+1}} \left(T_{I+1}^{N+1} - T_{I+1}^{N}\right) \right] - r_{I-1} N_{I-1} \chi_{I-1} \left[\frac{\theta_{I-1}}{B_{I-1}} \left(B_{I-1}^{N+1} - B_{I-1}^{N}\right) + \frac{\omega_{I-1}}{T_{I-1}} \left(T_{I-1}^{N+1} - T_{I-1}^{N}\right) \right] \right\}$$
$$\times \left(T_{I+1} - T_{I-1}\right)$$
$$+ \frac{1}{4 r_I N_I} \left(r_{I+1} N_{I+1} \chi_{I+1} - r_{I-1} N_{I-1} \chi_{I-1}\right) \left(T_{I+1}^{N+1} - T_{I+1}^{N} - T_{I-1}^{N+1} + T_{I-1}^{N}\right) .$$

Let $\alpha = \frac{\chi}{2(\Delta r)^2}$, $\delta_T = T_{I+1}^N - 2T_I^N + T_{I-1}^N$, $\gamma_1 = \frac{r_{I+1} N_{I+1} \chi_{I+1}}{8 r_I N_I (\Delta r)^2}$,

$\gamma_{-1} = \frac{r_{I-1} N_{I-1} \chi_{I-1}}{8 r_I N_I (\Delta r)^2}$, and let I+1, I, I-1 be denoted by 1,0,-1,

then

$$\left. \frac{1}{rN_e} \frac{\partial}{\partial r} \left(r N_e \chi_e \frac{\partial T_e}{\partial r} \right) \right|^{N+\frac{1}{2}} = 2\alpha\delta_T + 2\gamma_1 (T_1 - T_{-1})^N - 2\gamma_{-1} (T_1 - T_{-1})^N$$
$$+ \alpha \left\{ T_1^{N+1} - 2T_0^{N+1} + T_{-1}^{N+1} + \delta_T \left[\frac{\theta_0}{B_0^N} (B_0^{N+1} - B_0^N) + \frac{\omega_0}{T_0^N} (T_0^{N+1} - T_0^N) - 1 \right] \right\}$$
$$+ \gamma_1 \left[\frac{\theta_1}{B_1^N} (B_1^{N+1} - B_1^N) + \frac{\omega_1}{T_1^N} (T_1^{N+1} - T_1^N) \right] (T_1 - T_{-1})$$
$$- \gamma_{-1} \left[\frac{\theta_{-1}}{B_{-1}^N} (B_{-1}^{N+1} - B_{-1}^N) + \frac{\omega_{-1}}{T_{-1}^N} (T_{-1}^{N+1} - T_1^N) \right] (T_1 - T_{-1})$$
$$+ (\gamma_1 - \gamma_{-1}) \left[T_1^{N+1} - T_{-1}^{N+1} + T_{-1}^N - T_1^N \right] .$$

The plasma variables at the new time (N+1) and their coefficients can be tabulated:

Component	Coefficient (quantities evaluated at time N)
T_1^{N+1}	$\alpha + \gamma_1(T_1-T_{-1})\frac{\omega_1}{T_1} + \gamma_1 - \gamma_{-1}$
T_0^{N+1}	$-2\alpha + \alpha\delta_T \frac{\omega_0}{T_0}$
T_{-1}^{N+1}	$\alpha-\gamma_{-1}(T_1-T_{-1}) \frac{\omega_{-1}}{T_{-1}} -\gamma_1 + \gamma_{-1}$
B_1^{N+1}	$\gamma_1 \frac{\theta_1}{B_1}(T_1-T_{-1})$
B_0^{N+1}	$\alpha\delta_T \frac{\theta_0}{B_0}$
B_{-1}^{N+1}	$-\gamma_{-1} \frac{\theta_{-1}}{B_{-1}} (T_1-T_{-1})$
1	$\alpha\gamma_T(1-\omega_0-\theta_0) + (T_1-T_{-1}) \left[\gamma_1(1-\omega_1-\theta_1) - \gamma_{-1}(1-\omega_{-1}-\theta_{-1})\right]$

The remainder of the terms in the T_e equation can be evaluated in a straightforward fashion.

For the E equation,

$$\eta = \eta(T_e,\ldots) \sim CT_e^{-3/2}$$

$$E^{N+1/2} \approx \left(E + \frac{\Delta t}{2}\frac{\partial E}{\partial t}\right) = \left(\eta + \frac{\Delta t}{2}\frac{\partial \eta}{\partial t}\right)\left(j + \frac{\Delta t}{2}\frac{\partial j}{\partial t} - j_B\right)$$

and

$$\frac{\partial \eta}{\partial t} = \frac{\partial \eta}{\partial T_e}\frac{\partial T_e}{\partial t} = -\frac{3}{2}\frac{\eta}{T_e}\frac{\partial T_e}{\partial t} \quad .$$

Let

$\frac{\partial E}{\partial t} = \frac{1}{2}(E^N-E)$ where in this case E^N is the value at the new time. Then to order $(\Delta t)^2$

$$E^N + E = 2\eta(j - j_B) + \left[(j - j_B)\frac{\partial\eta}{\partial T_e}(T_e^N - T_e) + \eta(j^N - j)\right]$$

and

$$E^N - \eta j^N + \frac{3}{2}\frac{\eta}{T_e}(j - j_B)T_e^N = \frac{3}{2}\eta\ (j - j_B) - \eta j_B \quad .$$

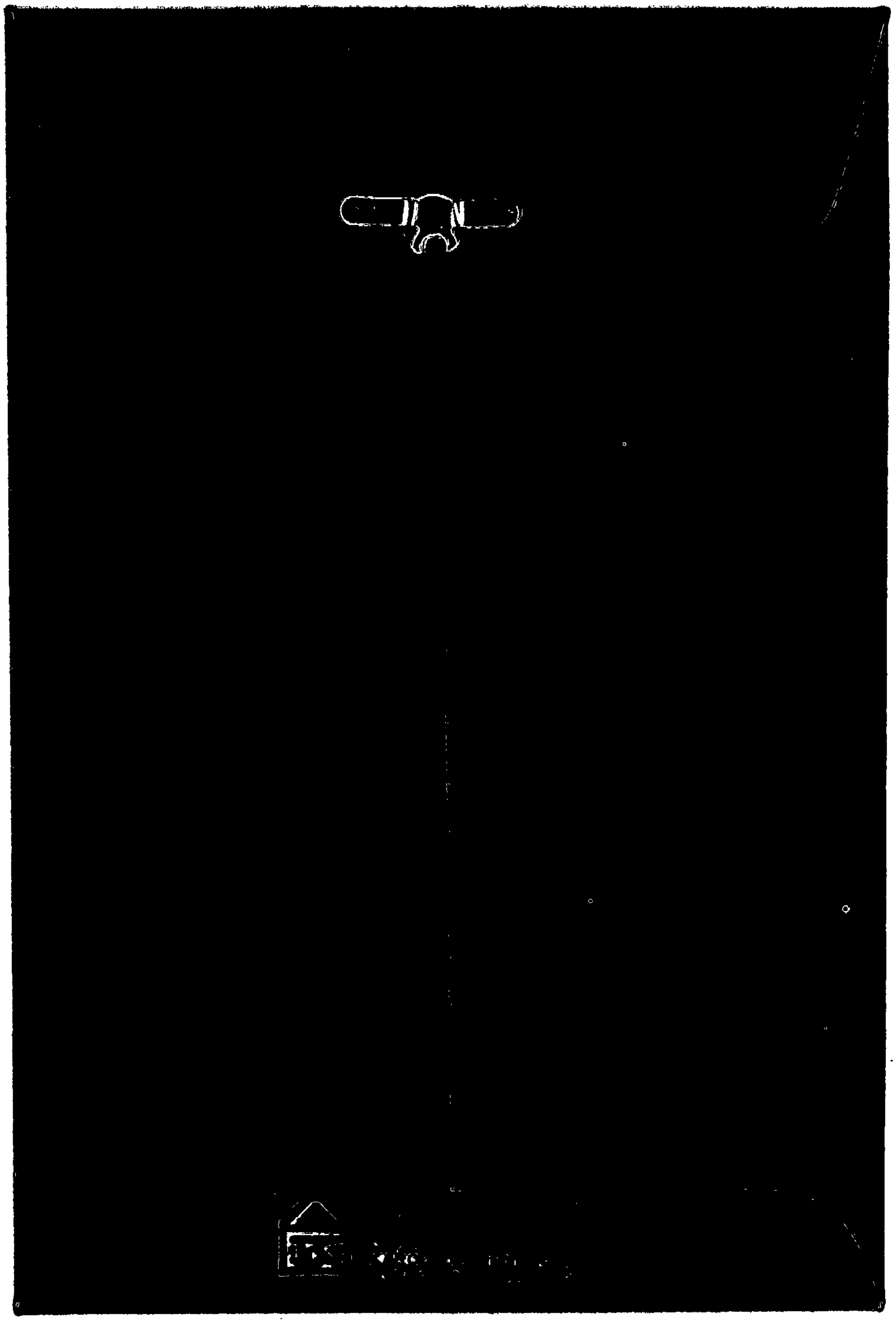
APPENDIX IV

ORNL/TM-5262

A USER'S MANUAL FOR THE OAK RIDGE TOKAMAK TRANSPORT CODE

J. K. Munro H. C. Howe
J. T. Hogan D. E. Arnurius

Appendix IV ▼ Microfiche



INTERNAL DISTRIBUTION

1. J. D. Callen
2. J. F. Clarke
3. R. A. Dory
4-27. J. T. Hogan
28. G. G. Kelley
29. D. G. McAlees
30. O. B. Morgan
31. M. W. Rosenthal
32-34. Laboratory Records
35. Laboratory Records — ORNL-RC
36. Y-12 Document Reference Section
37-38. Central Research Library
39. Fusion Energy Division Library
40. Fusion Energy Division Reports Office
41. ORNL Patent Office

EXTERNAL DISTRIBUTION

42. Plasma Physics Library, Plasma Physics Laboratory, Princeton Univ., Forrestal Campus, P.O. Box 451, Princeton, NJ 08540

43. Controlled Thermonuclear Research Library, Lawrence Livermore Laboratory, P.O. Box 808, Livermore, CA 94550

44. Q Division Library, c/o F. L. Ribe, Los Alamos Scientific Laboratory, P.O. Box 1663, Los Alamos, NM 87544

45. Controlled Thermonuclear Research Library, c/o Weston M. Stacey, Jr., Argonne National Laboratory, 9700 S. Cass Ave., Argonne, IL 60439

46. Magnetic Fusion Energy Computer Center, c/o Dr. John Killeen, Lawrence Livermore Laboratory, P.O. Box 808, Livermore, CA 94550

47. Librarian, Culham Laboratory, U.K. Atomic Energy Authority, Abingdon, Oxon, OX14, 3DB, United Kingdom

48. Ruth Lengye, Bibliothek, Max Planck Institut für Plasmaphysik, 8046 Garching bei München, Federal Republic of Germany

49. Library, Centre de Récherches en Physique des Plasmas, 21 Avenue des Bains, 1007, Lausanne, Switzerland

50. A. M. Dupas, Documentation S.I.G.N., Départment de la Physique du Plasma et de la Fusion Controlée, Association EURATOM-CEA sur la Fusion, Centre d'Etudes Nucléaires, BP 85 Centre Du TRI 38041 Grenoble Cedex (France)

51. Bibliothèque, Service du Confinement des Plasmas, C.E.A., B.P. No. 6, 92, Fontenay-aux-Roses (Seine) France

52. Library, International Centre for Theoretical Physics, Trieste, Italy

53. Library, Laboratorio Gas Ionizzati, Frascati, Italy

54. V. E. Ivanov, Physical-Technical Institute of the Ukranian Academy of Sciences, Sukhumi, U.S.S.R.

55. L. M. Kovrizhnikh, Lebedev Institute of Physics, Academy of Sciences of the U.S.S.R., Leninsky Prospect 53, Moscow, U.S.S.R.

56. Thermonuclear Laboratory, Kurchatov Institute of Atomic Energy, 46 Ulitsa Kurchatova, P.O. Box 3402, Moscow, U.S.S.R.

57. Library, Institute for Plasma Physics, Nagoya University, Nagoya, Japan 464

58. Library, FOM-Institut voor Plasma-Fysica, Rijnhuizen, Jutphaas, Netherlands

59. Plasma Physics Group, Department of Engineering Physics, Australian National University, P.O. Box 4, Canberra A.C.T. 2600 Australia

60. Thermonuclear Library, Japan Atomic Energy Research Institute, Tokai, Naka, Ibaraki, Japan

61. CTR Reading Room, c/o Prof. Dieter J. Sigmar, Room 37-391, MIT, Cambridge, MA 02139

62. CTR Reading Room, c/o Prof. D. W. Kerst, Dept. of Physics, Sterling Hall, Univ. of Wisconsin, Madison, WI 53706

63. CTR Reading Room, c/o Prof. I. B. Bernstein, Yale Univ., New Haven, CT 06510

64. Center for Plasma Physics and Thermonuclear Research, c/o D. W. Ross, Univ. of Texas, Physics Dept., Austin, TX 78712

65. CTR Reading Room, c/o Prof. B. D. Fried, Physics Dept., Univ. of California, Los Angeles, CA 90024

66. CTR Reading Room, c/o Prof. David C. Montgomery, Physics & Astronomy Dept., Univ. of Iowa, Iowa City, IA 52240

67. Magneto-Fluid-Dynamics Library, Courant Inst. of Math. Sci., New York Univ., 251 Mercer St., New York, NY 10012

68. CTR Reading Room, c/o Prof. Allan N. Kaufman, Physics Dept., Univ. of California, Berkeley, CA 94720

69. CTR Reading Room, c/o Prof. W. B. Thompson, Physics Dept., Univ. of California, San Diego, La Jolla, CA 92037

70. CTR Reading Room, c/o Prof. Alvin W. Trivelpiece, Dept. of Physics & Astronomy, Univ. of Maryland, College Park, MD 20742

71. CTR Reading Room, c/o Prof. T. Kammash, 103 Research Admin. Bldg., N. Campus, Univ. of Michigan, Ann Arbor, MI 48105

72. CTR Reading Room, c/o Dr. Ravi N. Sudan, Phillips Hall, Cornell Univ., Ithaca, NY 14850

73. Prof. Marshall N. Rosenbluth, Institute for Advanced Study, Princeton, NJ 08540

74. CTR Reading Room, c/o Prof. R. Gross, Plasma Research Lab., Columbia Univ., New York, NY 10027

75. CTR Reading Room, c/o Prof. Roy Gould, California Inst. of Tech., M.S. 116-81, Pasadena, CA 91125

76. Dr. Nicholas A. Krall, Science Applications, Inc., P.O. Box 2354, 1200 Prospect St., La Jolla, CA 92037

77. CTR Reading Room, c/o Dr. Jay P. Boris, Plasma Physics, Naval Research Laboratory, Washington, DC 20390

78. Professor A. Simon, Dept. of Mechanical & Aerospace Sciences, University of Rochester, Rochester, NY 14627

79. CTR Library, c/o Dr. Alan F. Haught, United Technologies Research Labs, East Hartford, CT 06108

80. Dr. Robert E. Price, Division of Magnetic Fusion Energy, G-234, Energy Research and Development Administration, Washington, DC 20545

81. Dr. Bennett Miller, Division of Magnetic Fusion Energy, G-234, Energy Research and Development Administration, Washington, DC 20545

82. Dr. Arthur Sleeper, Division of Magnetic Fusion Energy, G-234, Energy Research and Development Administration, Washington, DC 20545

83. Dr. Walter Sadowski, Division of Magnetic Fusion Energy, G-234, Energy Research and Development Administration, Washington, DC 20545

84. Dr. D. H. Priester, Division of Magnetic Fusion Energy, G-234, Energy Research and Development Administration, Washington, DC 20545

85. Mr. E. E. Kintner, Division of Magnetic Fusion Energy, G-234, Energy Research and Development Administration, Washington, DC 20545

86. Dr. L. D. Pearlstein, L-388, Lawrence Livermore Laboratory, P.O.
Box 808, Livermore, CA 94550

87. Dr. J. P. Friedberg, Los Alamos Scientific Laboratory, Los Alamos
NM 87544

88. Dr. David J. Rose, Dept. of Nuclear Engineering, MIT, Cambridge, MA
02139

89. Dr. Gareth E. Guest, General Atomic Co., P.O. Box 81608, San
Diego, CA 92138

90. Dr. Claude Mercier, Service du Theorie des Plasmas, Centre d'Etudes
Nucléaires, Fontenay-aux-Roses (Seine) France

91. Dr. J. B. Taylor, Culham Laboratory, UKAEA Abingdon, Oxon, OX14
3DB, United Kingdom

92. Dr. D. Pfirsch, Institute for Plasma Physics, 8046 Garching bei
München, Federal Republic of Germany

93. Dr. V. D. Shafranov, I. V. Kurchatov Inst. of Atomic Energy, 46
Ulitsa Kurchatova, P.O. Box 3402, Moscow, U.S.S.R.

94. Dr. Harold Grad, Courant Inst. of Math. Sci., New York Univ.,
251 Mercer St., New York, NY 10012

95. Dr. J. G. Cordey, Culham Laboratory, UKAEA, Abingdon, Oxon, OX14,
3DB, United Kingdom

96. Dr. David Baldwin, L-388, Lawrence Livermore Lab., P.O. Box
808, Livermore, CA 94550

97. Prof. Bruno Coppi, Dept. of Physics, MIT, Cambridge, MA
02139

98. Dr. Harold P. Furth, Princeton Plasma Physics Lab., Princeton
Univ., P.O. Box 451, Princeton, NJ 08540

99. Dr. Paul H. Rutherford, Princeton Plasma Physics Lab., Princeton
Univ., Princeton, NJ 08540

100. Dr. Herbert H. Woodson, Dept. of Electrical Engineering, Univ. of
Texas, Austin, TX 78712

101. Research & Technical Support Div., Oak Ridge Operations, Energy
Research and Development Administration, P. O. Box E, Oak Ridge,
TN 37830

102-128. Technical Information Center, Energy Research and Development
Administration, P.O. Box 62, Oak Ridge, TN 37830

☆ U.S. GOVERNMENT PRINTING OFFICE: 1977-748-189/178