FAFNER
A Fully 3-D Neutral Beam Injection Code Using Monte Carlo Methods

Graeme G. Lister

IPP 4/222 January 1985
FAFNER
A Fully 3-D Neutral Beam Injection Code Using
Monte Carlo Methods

Graeme G. Lister

IPP 4/222 January 1985

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

2. Structure of the Code

2.1 General
2.2 Initialisation
2.3 Beamline Computations
2.4 Fast Ion Deposition Profile (H(R) Calculation)
2.5 Ion Slowing Down and Power Deposition in Plasma
2.6 Output
2.7 Diagnostics and Restart Facility
2.8 Numerical Functions

3. Neutral Beam Configuration

3.1 General
3.2 Geometry
3.3 Ion Source Details
3.4 Neutral Beam Power Fractions, Beam Currents
3.5 Beam Distribution, Focussing, Divergence
3.6 Scrapers and Ducts

4. Description of the Plasma

4.1 General
4.2 Flux Surfaces
4.3 Plasma Variables
4.4 Treatment of Impurities
4.5 Neutral Density in the Plasma
4.6 Magnetic Field Configuration

5. Description of the Physical Models

5.1 Fast Ion Deposition (H(R) Calculation)

5.1.1 Neutral Beam Ionisation
5.1.2 Neutral Beam Collision Cross Sections
5.1.3 Finite Larmor Radius Effects
5.1.4 Sputtering from Walls

5.2 Guiding Centre Theory and Power Deposition

5.2.1 Guiding Centre Theory
5.2.2 The Fokker-Planck Equation
5.2.3 Charge Exchange and Ionisation

6. Description of Numerical Schemes

6.1 Treatment of the Beamline
6.2 Fast Ion Deposition Profile (H(R) Computation)

6.2.1 Fast Ion Deposition
6.2.2 Finite Larmor Radius Effects
6.2.3 Sputtering Coefficients
6.2.4 Ion Source for Guiding Centre Code

6.3 Ion Slowing Down and Power Deposition in Plasma

6.3.1 Guiding Centre Motion
Abstract

A computer code is described which models the injection of fast neutral particles into 3-dimensional toroidal plasmas and follows the paths of the resulting fast ions until they are either lost to the system or fully thermalised. A comprehensive model for the neutral beam injection system is included. The code is written especially for use on the CRAY-1 computer: in particular, the modular nature of the program should enable the most time consuming sections of the program to be vectorised for each particular experiment to be modelled. The effects of plasma contamination by possible injection of impurities, such as oxygen, with the beams are also included. The code may also be readily adapted to plasmas for which a 1 or 2-dimensional description is adequate. It has also been constructed with a view to ready coupling with a transport or equilibrium code.
1. Introduction

This report describes the FAFNER Code, which is a computer program designed to model neutral beam injection into 3-D plasmas. In order to demonstrate the running of the program in a full 3-D geometry, parameters chosen for modelling the test runs for the code are those appropriate to the proposed WVII-AS Stellarator Experiment. However, the program may be readily adapted to treat a variety of 1, 2, or 3 dimensional plasma and magnetic field configurations, such as the 2 dimensional "D" configuration in JET.

The program has been designed with a view to running on the CRAY-I Computer. We have used the OLYMPUS system (Ref.1) and hence expect it to be readily transferrable to other computer systems.

In Section 2, an outline of the structure of the code is given, together with an accompanying Flow Chart. The program is highly modular, and the functions of each of the sections is discussed separately.

In Section 3, a model for the beamline system used to inject neutral particles into the plasma is discussed. This section of the code provides a source function of co-ordinates of neutral particles in velocity space, which is used firstly to determine the initial ionisation of the neutrals in the plasma, and later the distribution of power to the plasma as a result of its interaction with the fast ions. Details of power losses due to collisions of the neutrals emitted from each source with scrapers and ducts along the beamline are also computed.

A description of various models for the plasma is given in Section 4. Plasma temperatures and densities are defined to be constant on flux surfaces and magnetic field configurations must be supplied by the user. The present version uses a model magnetic field derived from a set of coefficients provided by Domaschk (Ref.2).

The physical models used in describing the interaction processes of the injected neutral particles and the resulting fast ions with the plasma is discussed in Section 5. Most of the theory has been well documented elsewhere, (e.g. Refs.3-7), and only the equations relevant to the numerical treatment are discussed here. The numerical schemes used are discussed in detail in Section 6.

The Output Package is described in Section 7, and in Section 8 various error diagnostics are discussed, together with a facility to restart the code from an intermediate point in the calculation. This is particularly useful in the event that CPU Time for the job is exceeded, in which case all relevant information necessary to restart the job is stored on file. Standard numerical functions required for the Monte Carlo computations are described in Section 9.

A set of Tables is provided in Section 10, containing a list of all relevant COMMON variables for use in the code. Default values for all variables available to the user as input data are also provided. Tabular Data containing information regarding sputtering
coefficients, neutralisation efficiencies and ionisation cross sections are supplied via BLOCK DATA. A discussion of the use of this routine is supplied in Appendix 1, and an example of control cards and data needed to run the code is provided on the Cray-1 Computer at IPP in Garching is provided in Appendix 2.

2. Structure of the Code

2.1 General

A flow chart illustrating the structure of the FAFNER Code is given in Fig 1. The code is highly modular and comprises the following sections:

- Initialisation
- Beamline computations
- Fast ion deposition profile (H(R) Computation)
- Ion slowing down and power deposition in plasma
- Output
- Diagnostics and Restart Facility
- Numerical Functions

The OLYMPUS subroutine (1.8) START is used to control the beamline computations, while subroutine (2.1) STEPON forms the nucleus for the H(R) and Guiding Centre Codes, as illustrated in Fig 1. The functions and operation of each section of the code are discussed below. The Job Control Language and Data File required to run a job are described in Appendix 2.

2.2 Initialisation

Initialisation of variables and arrays in the code follows standard OLYMPUS nomenclature (Ref.1). Control of the program is through subroutine (0.8) MAIN, contained in the OLYMPUS Library. All COMMON variables are set to zero in subroutine (1.2) CLEAR. Default values for all independent variables required to run FAFNER are set in subroutine (1.3) PRESET, and these values may be overwritten using the NAMELIST facility in subroutine (1.4) DATA. Constants and arrays which are dependent on the input parameters are then computed in subroutine (1.5) AUXVAL.
All plasma and magnetic field variables to be used in the code are computed in (1.6) INITIAL. Subroutine INITIAL calls two subroutines:

(1.9) FLUX Set flux coordinates (section 4.2)
(1.10) FIELD0 Initialise Magnetic Field (section 4.6)

These subroutines are experiment dependent and will generally require modification by the user for use with other plasma devices. Subroutine INITIAL, however, is designed to model all plasmas in which the parameters may be described in terms of a single flux function (section 4.2).

Finally, tables of coefficients used in the code for neutralisation efficiencies in the beamline, sputtering coefficients for fast ions striking the torus and electron impact ionisation cross sections are tabulated in BLOCK DATA, a full description of which is given in Appendix 1. Connection to the main code is via the COMMON BLOCK COMDAT.

2.3 Beamline Computations

This section of the code describes the beamlines used to inject neutral atoms into the plasma and, in addition to providing a source of neutrals for use later in the code to compute power deposition profiles in the plasma, permits a detailed simulation of the power loadings on scrapers along the beamline due to collisions with escaping neutrals. Allowance is made for focussing and divergence of the individual beamlets, as well as efficiencies of the ion source and neutraliser. This package may also be run independently of the rest of the code.

This section of the code comprises three subroutines:

(1.8) START Control subroutine for beamline code
(1.11) SORSPT Selection of atom coordinates from sources
(1.12) BMLINE Check atoms avoid scrapers and ducts

The initial positions and directions of fast atoms emerging from the neutral beam sources are chosen in subroutine SORSPT using Monte Carlo techniques appropriate to the input data provided. Subroutine BMLINE then computes which of these particles reach the plasma and which are lost due to collisions with scrapers and ducts along the beamline. The relevant bookkeeping to compute scraper loadings is done in START and a set of coordinates and velocities of particles available to heat the plasma is stored for later use in the code (ref. section 6.1).
2.4 Fast Ion Deposition Profile (H(R) Calculation)

In this section of the code, the deposition profile of fast ions in the plasma (the so-called "H(R) profile") is computed. This computation is performed if the logical variable NLHSBR (Table 5) is set

\[ NLHSBR = \text{TRUE}. \]

The initial conditions for the calculation are supplied from the velocity-space coordinates of the atom source computed in the beamline code. The following subroutines and functions are required for this calculation:

- (2.2) SOURCE: Control subroutine for H(R) computation
- (2.3) CXSNEU: Compute mean free paths of atoms in plasma
- (2.4) BOUND: Coordinates of atoms at outer limiter
- (2.5) PSEUDO: Find if and where atom ionised in plasma
- (2.6) LOCATE: Find nearest flux surface to atom
- (2.7) VESSEL: Check if atom strikes confining torus wall

The initial coordinates of the atoms emerging from each neutral beam source are transposed such that each atom lies on the cylindrical surface defined by the outer point of the limiter, in subroutine BOUND. A Monte Carlo algorithm is then used to compute the position in the plasma in which the atom is ionised, either by ion or electron impact ionisation or by charge exchange with plasma ions. The computation of mean free paths for these processes is made in subroutine CXSNEU, for each of the relevant energy species injected.

During this calculation, it is necessary to compute if and where the atom is located in the plasma (FUNCTION LOCATE), and whether the atom remains confined in the toroidal vessel, or whether it strikes the torus wall and is lost (SUBROUTINE VESSEL).

The necessary book-keeping to compute H(R) profiles in the plasma as a function of the flux coordinate, "shine through" (the fraction of power entering the torus which is lost to the walls) and the resulting sputtering from metal surfaces, is summarised in subroutine SOURCE.

In studying the deposition of heavy atom impurities injected in the plasma with the neutral beams, Finite Larmor Radius effects are important for ions in low charge states. Thus an exact orbit following routine

(2.8) SUBROUTINE FLRORB

has been implemented, such that the paths of heavy ions (e.g., oxygen) may be followed through the plasma until they have reached a charge state for which FLR effects may be neglected and Guiding Centre Theory is appropriate.
The physical processes treated in this section are discussed in section 5.1, and the relevant numerical methods are discussed in Section 6.2. A set of coordinates describing the "birth" positions and velocities of a sample of fast ions created in SOURCE is then stored for later use in following the guiding centre motion of these ions and their interaction with the background plasma.

2.5 Ion Slowing Down and Power Deposition in Plasma

In this section of the code, the guiding centre motion of a set of fast ions, with initial co-ordinates in phase space provided from the program described in the previous section, are followed and the power deposition profiles resulting from their interaction with the plasma is computed. The fast ions may represent either light ions used to heat the plasma or heavy impurity ions. The full Fokker-Planck equation is solved over the slowing down time of the ions. For light ions the effect of charge exchange with background neutral atoms and possible re-ionisation are included, while for heavy ions, the successive ionisation through all possible ionisation states is computed. The computation is performed if the logical variable (Table 5)

\[
\text{NLGUID} = \text{TRUE.}
\]

The following subroutines and functions are required for this calculation:

- (2.9) HEAT: Control subroutine for Guiding Centre Code
- (2.10) COEF: Coefficients for Fokker-Planck Equation
- (2.11) SIGCHX: Fast ion charge exchange cross sections
- (2.12) NEUCHX: Re-ionisation of charge exchange neutrals
- (2.13) DENSNO: Function for background neutral density
- (2.14) FIELDS: Compute local magnetic fields
- (2.15) SIGION: Rate coefficients for ionisation

In addition, function (2.6) LOCATE and Subroutine (2.7) VESSEL, described in the previous section, are also required for this computation.

The physics processes treated in this section are discussed in detail in section 5.2, and the relevant numerical methods are discussed in Section 6.3. The book-keeping for power losses due to unconfined ions, and the power deposition to ions and electrons in the plasma is performed in subroutine HEAT, while details of losses due to charge exchange of the fast ions with background neutrals is computed in NEUCHX. Sputtering of impurities due to collisions with ions which leave the plasma and strike the torus walls is accounted for in Subroutine VESSEL.
2.6 Output

The Output package is controlled using the standard OLYMPUS Routines

(3.1) OUTPUT(K) Output Control Routine
(3.2) MPRINT(K) Control of Line Printer Output

where K=1 designates initial output, and K=2 final output. Subroutine MPRINT then calls the following routines:

K=1: Initial Output

(3.3) BEAMP Output for Beamline Code
(3.4) PLASMP Print details of Plasma
(3.5) FIELDP Print magnetic field details
(3.6) NUMERP Print details of numerical scheme

K=2: Final Output

(3.7) HSUBRP Output for H(R) calculation
(3.8) GUIDCP Output for Guiding Centre code

Details of the contents of these routines are given in section 7.

2.7 Diagnostics and Restart Facility

This section comprises two useful subroutines

(4.3) CRASH Save data if time limit exceeded
(5.4) ERROR Stop job for prescribed conditions

The usefulness of subroutine CRASH is self explanatory, and its exact construction and use is discussed in section 8. This facility is made available to the user by setting the logical variable (Table 10)

NLRPRV = .TRUE.

and ensures that if during the running of a job, CPU Job Time Limit is exceeded, all relevant information required to restart the job is written onto a file. Subroutine CRASH is called from the CRAY System reprieve routine SETRPV, and in using this facility the user must reckon with a 20 sec overhead on the job running time.

Subroutine ERROR is used to stop the code in the case where:

1. Losses in the beamline are unacceptable
2. "Shine through" of neutrals in the plasma is unacceptable
3. Collision cross sections for element are not available

If either of these conditions occur, the code is aborted and the relevant error message is printed (ref. section 8).
2.8 Numerical Functions

The following numerical routines are used for the Monte-Carlo calculations:

(2.1) RNFL(K) Random number generator
(2.2) GAUSS(PS) Select from Gaussian distribution

The routine RNFL(K) calls the Cray 1 random number generator RANF, while function GAUSS(PS) is a function used to select from a Gaussian distribution of half width PS.

3. Neutral Beam Configuration

3.1 General

We assume that the neutral beam system consists of N_b beamlines, each containing n_b neutral beam sources, with a total number n_s of sources available to heat the plasma. Provision is made for a maximum of 16 sources. The species of neutral injected is set by the integer N_s (see Table 8 for list of elements) which corresponds to an atomic weight A_b and atomic number Z_b, assumed the same for each beam.

Unless otherwise stated, units are in cgs throughout the program. Default parameters correspond to hydrogen injection in WVII-AS. A list of input parameter is given in Table 1 and a list of the default values is given in Table 2.

3.2 Geometry

The neutral beam configuration adopted in FAFNER is illustrated in Fig 2. We have adopted two separate coordinate systems:

(i) The plasma and torus geometry is described using a cylindrical coordinate system (R, \phi, z), with respect to the centre O of the toroidal vessel containing the plasma, where the toroidal angle \phi is measured with respect to the line OC (usually one of the toroidal field coils). The reference point on the torus for beamline n is the centre of the injection duct D (where the subscript n is assumed), located at \( (R_0, \phi_0) \).

(ii) The geometry of the individual neutral beam sources in each beamline (Fig 2a) is described using cartesian coordinates (x, y, z), with origin S at a distance R_0 from O along the projection of the line OD (where the beamline subscript n is again assumed). The x-axis is defined in the mid-plane of the torus directed outwards at an angle \( \Theta_x \) to OS. \( \Theta_x \) is deemed positive if the beamline direction is such that a particle injected along the beamline axis would move initially in the direction of positive \( \phi \).
Each neutral beam source is described by an index \( j \), \((1 \leq j \leq n)\), numbering consecutively the individual components of beamlines \( l \) to \( N \). The centre of a typical neutral beam source is situated at \( Q_j \) with co-ordinates \((x_j, y_j, z_j)\). The central beamlet is directed at an angle \( \Phi_j \) to the horizontal (Fig 2b), in the \( x-z \) plane and such that its projection in the horizontal plane intersects the source axis \( SQ_j \) at an angle \( \alpha_j \) (Fig 2a) in the \( x-y \) plane. An additional logical variable, NLBTYP, is included to allow for the coupling of differently constructed beamlines to the torus. If NLBTYP(\( n \)) is set FALSE, it is assumed that beamline \( n \) is internally identical to beamline \( n-1 \), although the injection angle \( \Phi_j \) and the intersection point \( S \) may differ.

We may also define a radius of tangency \( R \), which is defined from the horizontal projection of \( SQ_j \) with respect to the centre \( O \) (Fig 1a), and is calculated in the code from the input data. Further, each beam may be switched on or off using the logical variable NLBEAM.

Neutral particles are assumed to emerge from a planar surface which is either rectangular (height \( h_\theta \), width \( w_\theta \)) or circular (diameter \( h_\theta \)). Rectangular beams may also have a "tilt" \( \Phi_j \), measured clockwise with respect to the axis \( SQ \) (Fig 2c). Beam shape is set by the parameter NSHAPE (Table 1). It is sometimes instructive to replace the real beam model with a simple pencil beam and this may also be simulated using the NSHAPE parameter.

### 3.3 Ion Source Details

The code provides an option to supply details of the ion source. In the case where one of the isotopes of hydrogen is to be injected three species of ion \( H^+, H_2^+, \) and \( H_3^+ \) (where \( H \) is used to represent hydrogen, deuterium or tritium) will be emitted from the source with relative power fractions \( p_{1}^+, p_{2}^+, p_{3}^+ \). A fraction \( \eta \) (\( k=1,3 \)) of these ions will be dissociated and neutralised into \( H \) atoms with energies \( E, E/2 \) and \( E/3 \) respectively, where \( E \) (eV) is the energy of ions leaving the ion source. A table of ideal neutralisation efficiencies is supplied in BLOCK DATA, and these are used to interpolate for values of \( \eta \) unless alternative values are supplied in the input data. The power fractions of the species in the neutral beam \( p_{1}^+, p_{2}^+, p_{3}^+ \) are then computed. In addition if the current in the ion source \( I_\theta \) is supplied the current and power leaving the neutral beam injector may also be computed. Assuming the ion source to be 100% efficient, the total power delivered to the neutraliser \( P_\psi \) is given (in watt) by

\[
P_\psi = I_\psi V_\psi
\]

where \( V_\psi \) volt is the voltage across the ion source. Including the neutralisation efficiencies, \( \eta \), we find the maximum power available from the neutral beams to be \( P_{N(max)} \)

\[
P_{N(max)} = \eta \left[ P_{1}^+ + P_{2}^+ + P_{3}^+ \right]
\]

where the normalisation \( p_{1}^+ + p_{2}^+ + p_{3}^+ = 1 \) has been assumed.
However, we can isolate 4 possible loss mechanisms which may reduce the power available for plasma heating:

(i) Inefficiency of the ion source
(ii) Non-ideal neutralisation efficiency
(iii) Re-ionisation between neutraliser and plasma
(iv) Collisions of neutral particles with scrapers or ducts.

Point (iv) is treated in detail in section 3.6, while mechanisms (i) to (iii) are collectively simulated by an efficiency factor, $\epsilon_N$. Thus the power available at each neutral source is:

$$P_N = \epsilon_N \sum_{k} n_k^+ \epsilon_k^+$$

and the power entering the torus $P_T$ is:

$$P_T = (1 - f_L) P_N$$

where $f_L$ represents losses described in section 3.6.

3.4 Neutral Beam Power Fractions, Beam Currents

In the event that no ion source details are supplied, it is necessary to give the neutral power fractions $p_{b_1}, p_{b_2}, p_{b_3}$ as input data. Also, if the current in the ion source is not given, then the neutral beam power $P_N$ must be supplied, from which $P_T$, the power entering the torus, is computed by including the effects of the scrapers and ducts (Eq. 3.3) The neutral beam current entering the torus is thus

$$I_T = (p_{b_1} + 2p_{b_2} + 3p_{b_3}) P_T / \gamma_s$$

where

$$p_{b_k} = \frac{\epsilon_k^+ \gamma_k^+}{\sum_k \epsilon_k^+ \gamma_k^+} p_k^+$$

(3.4)

It may also be noted that the particle species fractions $p_{b_k}^n$, are given by

$$p_{b_k}^n = kp_{b_k} / \sum_k kp_{b_k}$$

(3.5)

and this parameter is used internally in the code for statistical purposes.
3.5 Beam Distribution, Focusing, Divergence

Neutral particles are selected from the beam face assuming a uniform distribution. The centre of each beamlet is directed such that its horizontal and vertical focal lengths are $F_x$ and $F_y$, respectively, measured along the line of the central beamlet. The neutral particles in each beamlet will diverge about their focal line, and this is simulated by assuming a Gaussian distribution, with e-folding angles $\delta_x$ and $\delta_y$ representing horizontal and vertical divergences respectively. Values of $\delta_x$ and $\delta_y$ supplied as input data assume a distribution $\exp(-\Theta^2/2)$, where $\Theta$ is the divergence and $\Theta$ the mean divergence (in degrees). These $\delta$ values are converted to $\delta'$ in the code, such that $\delta'$ represents a Gaussian divergence ($\delta' = \delta/\sqrt{2}$) in radians.

3.6 Scrapers and Ducts

In a typical beamline, a number of $n_a$ apertures (or scrapers) shape the beam before it enters the torus. Each neutral particle is required to pass through a series of apertures, whose geometry is described relative to the co-ordinate system of the beamline to which it belongs, in an identical way to that in which the sources are described in section 3.2. The centre of each aperture is located at the point $(x_A, y_A, z_A)$. The normal to the plane of each aperture is inclined at an angle $\phi_A$ to the x axis in the x-y plane and $\phi_A$ to the horizontal plane (Fig 3a). The shape of each aperture may be either circular, elliptical or rectangular, or a rectangle with semicircles at top and bottom (as in the W7AS configuration, (Fig 3b) set by the parameter NSCRAP. Each aperture has a width $W_A$ and in the case of non-circular apertures, a height $H_A$. In the case of non-circular apertures, a rotation of angle $\delta_A$ about the beamline axis is also allowed for. Non-standard scraper configurations may also be included by use of the OLYMPUS routine EXPERT. Finally, since some beamlines will contain scrapers which are used only for specific sources in each beamline, a logical variable NLSCRAP is introduced which is set TRUE for each source which is required to pass through a particular scraper.

Thus, before computing the deposition of fast neutrals in the plasma, those particles which would not reach the torus due to intermediate collisions with ducts or scrapers may be eliminated, and the fraction of such losses, $f_L$ calculated. The fraction of neutrals deposited on each scraper is first calculated, and then summed for each beamline and finally the sum over all beamlines is computed.
4. Description of the Plasma

4.1 General

The plasma is assumed to be composed of a total of \( n_p \) different species of ion, specified by a set of integers \( N_{p} \) (see Table 8 for a Table of elements), from which the corresponding atomic weights \( A_{p} \) and atomic numbers \( Z_{p} \) are specified. The list of ion species may include plasma impurities, although an alternative treatment is described in section 4.4. A summary of COMMON variables used to describe the plasma is given in Table 3, and a list of default parameters for input data may be found in Table 4.

The plasma is assumed to be contained in a toroidal vessel described by subroutine

\[
(2.7) \quad \text{SUBROUTINE VESSEL}
\]

As default, this is a toroidal cylinder of minor radius \( r_N \) and major radius \( R_N \).

In order to compute the sputtering of metal ions due to the collision of unconfined particles (ions and atoms) with the torus wall, a variable \( N_{v} \) is introduced, which defines the metal surfaces \( N_{v} = 1: Fe, 2: Mo, ref. Table 9 \). A discussion of sputtering computations is given in section 5.1.

4.2 Flux Surfaces

The plasma densities and temperatures are defined as functions of a flux coordinate \( s \), where \( s \) has dimension of length. A set of \( N_s \) flux surfaces, defined by \( s(j) \), is used in the code, where \( s(N_s) \) corresponds to the plasma boundary. The coordinate \( s \) is defined in subroutine

\[
(1.9) \quad \text{SUBROUTINE FLUX}
\]

and may be used to represent a 1, 2 or 3 dimensional plasma. In the present (WVII-AS) version of the code, tables of \( s \) values are read from a mesh in cylindrical coordinates \( (R, \phi, z) \), with dimensions \( (N_R, N_\phi, N_z) \). The volume enclosed by flux surface \( s(j) \), \( V_s \), is also read from a table in subroutine FLUX. The volume enclosed by \( s(N_s) \) is clearly the plasma volume \( V_p \).

For use in 1 dimensional plasmas, \( s \) clearly corresponds to the minor radius coordinate, while \( s(N_s) \) is the plasma radius. In the WVII-AS version, \( s \) is defined as the distance from the plasma axis in the \( \phi = 0 \) plane (with coordinates \( (R_0, \phi_0, n_0) \) at which the flux surface cuts the mid-plane \( z=0 \) of the plasma, on the outer side of the torus. Thus \( s(N_s) \) may be regarded as the effective plasma radius.
4.3 Plasma Variables

Each plasma variable is characterised by an integer \( j_V \), where \( j_V \) has the following meaning:

<table>
<thead>
<tr>
<th>( j_V )</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( s )</td>
<td>Flux coordinate</td>
</tr>
<tr>
<td>2</td>
<td>( n_e )</td>
<td>Electron density</td>
</tr>
<tr>
<td>3</td>
<td>( n_i )</td>
<td>Individual ion densities</td>
</tr>
<tr>
<td>4</td>
<td>( T_e )</td>
<td>Electron temperature</td>
</tr>
<tr>
<td>5</td>
<td>( T_i )</td>
<td>Ion temperature</td>
</tr>
<tr>
<td>6</td>
<td>Zeff</td>
<td>Effective Z of plasma</td>
</tr>
<tr>
<td>7</td>
<td>( \langle Z \rangle )</td>
<td>See section 4.4</td>
</tr>
<tr>
<td>8</td>
<td>( n_o )</td>
<td>Neutral particle density</td>
</tr>
<tr>
<td>9</td>
<td>( V_3 )</td>
<td>Volume enclosed by ( s )</td>
</tr>
</tbody>
</table>

If the logical variable NLEXPT(\( j_V \)) = .TRUE., then a table of experimental values for variable \( j_V \) is read as input data through NAMELIST NEWRUN in subroutine (1.4) DATA. If NLEXPT(\( j_V \)) = .FALSE., a prescribed profile is used. For the variables \( n_e, T_e, T_i \), we make use of an integer variable NPROF(\( j_V \)), such that for variable \( g \):

(i) if NPROF(\( j_V \)) = 1

\[
g = (g_o - g_a)(1 - (s/r_3)k_3) + g_a \tag{4.1}
\]

(ii) if NPROF(\( j_V \)) = 2

\[
g = g_o/(1 + (s/r_3)k_3) \tag{4.2}
\]

where \( g_o \) and \( g_a \) are the values of variable \( g \) on the plasma axis and the plasma edge respectively, \( r_3 \), \( k_3 \) and \( p_3 \) are a set of parameters set by the user.

The ion densities \( n_i \) of each species \( k \) are defined as a fraction \( f_{ik} \) of the electron density of the plasma, where \( f_{ik} \) is supplied by the user. The exact treatment of ion densities to allow for a self consistent treatment of Zeff, is discussed in the next section.
4.4 Treatment of Impurities

If the set of ion densities $n_i$ includes impurity ions, the quantities $Z_{eff}$ and $Z$ may be calculated directly, using

$$Z_{eff} = \sum_{i} n_i Z_i / n_i = \sum_{i} f_i Z_i$$  \hspace{1cm} (4.3)

$$<Z> = \sum_{i} n_i Z_i / A_n = \sum_{i} f_i Z_i / A_i$$  \hspace{1cm} (4.4)

However, if $n_i$ is used to represent only the "pure" plasma, an effective impurity ion density, corresponding to ions of atomic weight $A_i$ and atomic number $Z_i$, is introduced and the self consistent pure plasma densities $n_e$ and impurity ion density are then:

$$n_i = \frac{Z_i - Z_{eff}}{Z_i \sum_{i} f_i Z_i - \sum_{i} f_i Z_i} f_i n_e$$  \hspace{1cm} (4.5)

$$n_e = \left( n_e - \sum_{i} n_i Z_i \right) / Z_{eff}$$  \hspace{1cm} (4.6)

Unless $Z_{eff}$ is read from tables of experimental data, it is assumed to be a constant in the plasma, $Z_{eff} = Z_{eff_0}$.

Finally, the total ion density of the plasma is $n_i = \sum_{i} n_i + n_{\text{imp}}$.

4.5 Neutral Density in the Plasma

The background neutral density in the plasma will contribute to the loss of heating power available to the plasma, since fast ions may be re-neutralised by charge exchange and hence be lost to the system. The neutral density, $n_n$, is modelled in the code as a function

\begin{equation}
(2.13) \text{FUNCTION DENSNO}(K)
\end{equation}

where $K$ is a dummy parameter. If NPROF(8)=3, a radial profile is used of the form

$$n_n(r) = n_n(r_\text{L}) \left( \frac{r_\text{L}}{r} \right)^{10}$$  \hspace{1cm} (4.7)

where $n_n(r_\text{L})$ is the density at the limiter radius, $r_\text{L}$, and $\alpha_n$ is a suitable scaling coefficient.
4.6 Magnetic Field Configuration

The stellarator magnetic field configuration used in the WVII-AS version of FAFNER is constructed from a set of coefficients computed by Domaschke (Ref. 2). Two subroutines, RFPAR and VGDFI, have been supplied by the WVII-A group to compute the magnetic fields. They are called in the following way:

(i) Subroutine RFPAR is called firstly by subroutine

\[(1.10) \text{ SUBROUTINE FIELD0} \]

in FAFNER in order to supply the initial magnetic field coefficients and later in subroutine

\[(3.5) \text{ SUBROUTINE FIELDP} \]

in order to print these coefficients in the OUTPUT package.

(ii) Subroutine VGDFI is called in the FAFNER subroutine

\[(2.14) \text{ SUBROUTINE FIELDS (PROCYL, PHI, PZ, PB, PBAIS, PGRDB2, PBXGB)} \]

where the arguments have the following meanings:

- PROCYL, PHI, PZ
- PB(3)
- PBAIS
- PGRDB2(3)
- PBXGB(3)

WVII-AS is an m=5 stellarator, and m is also a parameter used in the code.

In addition to the stellarator fields, a toroidal field is included, such that at major radius coordinate \( R \), it has the value \( B_T \), where

\[ B_T = B_0 R_0 / R. \]

where \( R_0 \) is the major radius of the plasma axis. It is also possible to include a constant vertical field \( B_v \).
5. Description of the Physical Models

5.1 Fast Ion Deposition (H(R) Calculation)

5.1.1 Neutral Beam Ionisation

The deposition of fast ions in the plasma due to interaction with the neutral beams is governed by the equation

\[ F(s) = \int f(x, v) \exp\left(-\int_0^s \frac{dl}{\lambda(x, v)}\right) dx dv \]  

(5.1)

where \( F(s) \) represents the flux of neutral particles crossing the flux surface \( s \), \( f(x, v) \) is the source function of neutral particles (see section 3) with space and velocity co-ordinates \( x, v \), respectively, and \( \lambda \) is the total mean free path to collision for all ionisation and charge exchange processes (see Section 5.1.2), which is a function of \( v \) and the plasma co-ordinates \( x_p \). The integral \( dl \) represents an integration along the path of each neutral particle from its point of origin to its intersection with the flux surface \( s \), at point \( x \).

Allowing for the discrete nature of the neutral energies and the flat surfaces of the neutral sources, the computation of \( F(s) \) requires an integration with a dimensionality somewhat greater than 7. A representation of the source function \( f(x, v) \) is computed in the beamline code (sections 3 and 6.1). The numerical methods required for computing the interaction of the neutral beam with the plasma are discussed in section 6.2.

5.1.2 Neutral Beam Collision Cross Sections

The mean free path for collision is defined by

\[ \lambda = \frac{v}{\nu_b} \]  

(5.2)

where

\[ \nu_b = \langle \delta v \rangle \rho_e + \sum_k (\delta_k^a + \delta_k^b) v n_k \]  

(5.3)

In equation (5.3), \( \langle \delta v \rangle \) is the rate coefficient for electron impact ionisation with the incoming neutral, \( v \) its velocity, \( \rho_e \) is the local electron density, \( n_k \) the density of ion species \( k \) and \( \delta_k^a \) and \( \delta_k^b \) are the cross sections for ion impact ionisation and charge exchange between plasma ions and injected neutrals.

In principal, the code is capable of treating any number of plasma species and any species of injected neutral. In the present version
of the code, the plasma is assumed to consist of a mixture of hydrogen isotopes, with additional impurities. The injected species may be either one of the hydrogen isotopes or oxygen.

Cross sections for ion impact ionisation in hydrogen plasma are from a form fit by Freeman and Jones (Ref. 8),

$$\sigma = \sum_{m=0}^{n} C_m (\frac{e}{E/A})^m$$

(5.4)

where $E/A$ is the neutral energy per nucleon and $C_m$ are the set of coefficients:

- $C_0 = -42.03$
- $C_1 = 3.557$
- $C_2 = -1.045$
- $C_3 = 0.3133$
- $C_4 = -7.045 \times 10^{-2}$
- $C_5 = 8.459 \times 10^{-3}$
- $C_6 = -3.495 \times 10^{-4}$

Cross sections for impact ionisation with impurity ions are computed from a formula by Dilson et al. (Ref. 9),

$$\sigma = 4.16 \times 10^{-16} Z_2^2 (1 - \exp(-C/E)) \sigma_0$$

(5.5)

where $E = E/(3.2 \times 10^4)/A/2$, $A$ and $Z$ being the mass number and charge of the plasma impurity ions. This equation is applicable for $E/A$ greater than 40 keV/nucleon. For injection energies less than 40 keV/nucleon, the classical limit, $\sigma = 4.6 \times 10^{-16} E/(keV)/Z_2^2$ has been used.

Rate coefficients for electron impact ionisation are read from tables. In the case of $H, D$ or $T$ injection, rate coefficients again come from Freeman and Jones (Ref. 8), while for oxygen injection data is taken from Ref. 10.

Charge exchange cross sections are only available in the code for $H, D$ or $T$ injection into a plasma composed of a mixture of hydrogen isotopes, and are taken from a fit by Riviere (Ref. 11),

$$\sigma = \frac{6.937 \times 10^{-15} (1 - 0.115 \log_{10}(E/(keV)/A)^{1.5})}{(1 + 1.112 \times 10^{-15} (E/(keV)/A)^{0.3})}$$

(5.6)

5.1.3 Finite Larmor Radius Effects

For the study of injected impurity ions such as oxygen, the guiding centre approximation (section 5.2) is inadequate for ions in low charge states, since in this case the ion Finite Larmor Radius (FLR) may be of the same order as the plasma dimensions. However, ionisation from lower to higher charge states is generally rapid, and a routine has been written which solves the real equation of motion for the ion:

$$\frac{dv}{dt} = qv \times B/m$$

(5.7)
where $q$ is the ion charge, $m$ its mass, $\mathbf{v}$ its velocity and $\mathbf{B}$ the local magnetic field. Successive ionisation by electron impact ionisation using Monte Carlo techniques is then used to bring the ion to a charge state for which FLR effects are no longer important and the computation may be continued using the Guiding Centre approximation.

5.1.4 Sputtering from Walls

Escaping neutrals which have failed to ionise in the plasma and strike the torus walls may make a considerable contribution to plasma impurities due to sputtering. The impurity current due to this effect is computed in the code, both for escaping neutrals and unconfined ions. (Section 5.2). The computation allows for sputtering from iron (Ref.12) or molybdenum (Ref.13), and coefficients for sputtering from other metals could readily be added.

5.2 Guiding Centre Theory and Power Deposition

5.2.1 Guiding Centre Theory

The equations describing the "guiding centre" motion of the fast ions are

$$\frac{d\mathbf{v}}{dt} = v_{\parallel} + v_{\perp}.$$  \hspace{1cm} (5.8)

$$\mathbf{\mu} = \frac{v_{\perp}}{B} = \text{constant}.$$  \hspace{1cm} (5.9)

where $v_{\parallel}$ and $v_{\perp}$ are the components of the fast ion velocity parallel and perpendicular to the magnetic field, and $v_{\perp}$ represents the sum of "Grad B" and "Curvature" drift velocities of the ions from the magnetic surfaces, given by

$$v_{\perp} = 1.0 \times 10^6 \left(1 + \frac{v_{\parallel}}{\sqrt{v_{\perp}}}\right)E_0 \mathbf{B} \times \nabla \mathbf{B} / Z_e B^3.$$  \hspace{1cm} (5.10)

where $E_0$ (eV) is the fast ion energy and $Z_e$ its charge state.

Equation (5.9) for the conservation of magnetic moment is more readily treatable numerically in the form

$$\frac{dv_{\parallel}}{dt} = -\frac{1}{2} \mathbf{\mu} \cdot (\nabla \times \mathbf{B}).$$  \hspace{1cm} (5.11)

where $(\nabla \times \mathbf{B})_{\parallel}$ is the component of $\nabla \times \mathbf{B}$ parallel to the magnetic field.
5.2.2 The Fokker-Planck Equation

The Fokker-Planck equation may be written in the form (Cf. Ref.7)

\[
\frac{\partial f}{\partial t} = \frac{1}{u^2} \frac{\partial}{\partial u} \left( u^3 \left( \frac{1}{\tau_e} + \frac{1}{\tau_i} \right) f \right) - \frac{1}{\epsilon_b} \frac{\partial}{\partial \epsilon} \left( \frac{\epsilon}{\tau_e} + \frac{\epsilon}{\tau_i} \right) \frac{\partial f}{\partial \epsilon} + \dot{\epsilon} \frac{\partial}{\partial \epsilon} \left( 1 - \dot{\epsilon}^2 \right) \frac{\partial f}{\partial \epsilon}
\]

(5.12)

where

\[
\tau_e = 3.16 \times 10^6 \frac{A_b \rho_e}{\epsilon_b Z_b^2} \ln \Lambda_e
\]

\[
\tau_i = \frac{3^{1/2}}{\Gamma}
\]

\[
\Gamma = 1.804 \times 10^{-7} n_{\text{eff}} A_b Z_b^2 \ln \Lambda_e
\]

\[
\dot{\epsilon}_e = 1.195 \times 10^{11} n_{\text{eff}} \ln \Lambda_e / A_b v^3
\]

\[
\dot{\epsilon}_i = \frac{\nu_{\text{HI}}}{v}
\]

The quantity \(\Lambda_e\) has been defined in Eq.4.4. \(A_b\) is the mass number of the ion, \(\ln \Lambda_e\) is from the Spitzer formula (Ref.14).

\[
\ln \Lambda_e = 23.4 - 1.15 \log_{10} n_e + 3.45 \log_{10} T_e - \ln Z_b \quad : \quad T_e < 50 \text{ eV}
\]

\[
= 25.3 - 1.15 \log_{10} n_e + 2.35 \log_{10} T_e - \ln Z_b \quad : \quad T_e > 50 \text{ eV}
\]

and

\[
\Lambda_e = 7.7 \times 10^8 \frac{\nu_\epsilon}{T_e} \frac{\epsilon}{\nu_\epsilon} (T_e + E_{\text{th}}) / n_e
\]

A discussion of the numerical model used to solve this equation is given in Section 5.3.

5.2.3 Charge Exchange and Ionisation

Fast ions may also undergo charge exchange and be either lost to the system or re-ionised elsewhere in the plasma. The cross sections for this process for the case of hydrogen isotope injection are those of equation (5.6). For the study of oxygen deposition, tables supplied from Ref.10 are used in a way identical to that described in Section 5.2.
6. Description of Numerical Schemes

6.1 Treatment of the Beamline

In order to compute the power loadings on the beamline scrapers, the space coordinates of a set of \( N_p \) atoms are selected randomly from the surface of each neutral source, according to the dimensions supplied by the user (see section 3.2). The direction of each atom is such that its divergence about its focal line is selected from a Gaussian distribution (section 3.5) using the Box-Muller technique (Ref. 15). The velocity-space coordinates so selected are in the reference frame of the source \((x_o, y_o, z_o)\), illustrated in Figs 2b and 2c. A transformation is made into the coordinate frame of the beamline \((x, y, z)\), as shown in Fig 2a. From the initial position and direction of an atom on the source, the point of intersection with the plane of the scraper, described by the equation,

\[
(x - x_A) \cos \alpha_A + (y - y_A) \sin \alpha_A + (z - z_A) \tan \beta_A = 0
\]  

(6.1)

may be computed, where the coordinates \((x_o, y_o, z_o)\) and the angles \(\alpha_A\) and \(\beta_A\) are described in section 3.6. It is then possible to test if the selected atom has struck the scraper by a simple geometric transformation. The contribution of each atom which strikes the scraper is then stored for later use in computing the power loadings.

Finally, a further geometrical transformation is made such that the coordinates in velocity space of the atoms are in the coordinate frame of the torus \((x_r, y_r, z_r)\), illustrated in Fig 2a. A set of \( N_o \) coordinates in velocity space,

\[
(x_i, y_i, z_i, \tilde{v}_{x_i}, \tilde{v}_{y_i}, \tilde{v}_{z_i}, i=1,N_o)
\]

corresponding to atoms which are available for plasma heating, is then stored (Table 7) for later use in the code (section 5.2). For convenience, the velocity coordinates described above are normalised such that

\[
\tilde{v}_{x_i}^2 + \tilde{v}_{y_i}^2 + \tilde{v}_{z_i}^2 = 1
\]  

(6.2)

Finally, the beamline code will terminate with an error message

BEAMLINE FOCUSSING FAULTY

if \( f_L > f_L(\text{max}) \), where \( f_L \) is the fractional power loss to the scrapers and \( f_L(\text{max}) \) is the maximum permitted loss, set by the user.
6.2 Fast Ion Deposition Profile (H(R) Computation)

6.2.1 Fast Ion Deposition

This section of the code is concerned with the solution of equation (5.1) for the interaction of injected atoms with the plasma. The computation is performed if the logical variable

\[ \text{NLHSBR} = .\text{TRUE}. \]

Extensive use is made of random number generators in this code, and a random number \( \xi \) is always assumed to lie in the interval \((0,1)\).

A set of \( N_N \) neutral particle co-ordinates are selected from the source computed in the beamline code, described in the previous section. The velocity coordinates thus supplied are normalised (Eq. 6.2) and each neutral must be assigned an energy. This is done using the particle species fractions, \( p^* \) (Eq. 3.5). A random number, \( \xi_j \), is selected, and the energy \( E_j \) of the neutral is set according to

\[
\begin{align*}
\text{if} & \quad \xi_j > \xi_1 & E_j = E \\
\text{if} & \quad \xi_1 > \xi_j > \xi_1 + p^* & E_j = E/2 \\
\text{if} & \quad \xi_1 + p^* > \xi_j > \xi_1 + p^* + p^*_p & E_j = E/3
\end{align*}
\]

where \( E \) represents the injection energy of the beams. The integration of a typical path of an atom from the edge of the plasma to a collision is computed using a method suggested by Putinskij (Ref. 16). The initial coordinates of the neutrals are transposed from their position on the source face such that they lie on a cylindrical surface bounded by the plasma limiter. The smallest mean free path in the plasma, \( \lambda_{\text{min}} \), for each neutral species injected is computed, and then the neutral is advanced a distance

\[
\begin{align*}
\text{d} & = -\ln \left( \xi_2 \right) \lambda_{\text{min}} \\
\end{align*}
\]

where \( \xi_2 \) is a random number. The atom is now located between flux surfaces \( s_{j-1} \) and \( s_j \), and the probability that a collision will occur is

\[
p = \lambda_{\text{min}} / \lambda(s_j, v_j)
\]

where \( \lambda(s_j, v_j) \) (Table 7) is the total collision mean free path, (Eq. 5.2), on the flux surface \( s_j \) for an atom with velocity \( v_j \). Another random number, \( \xi_3 \), is now chosen, and if

\[
p > \xi_3
\]

...
collision is deemed to have occurred. If \( p < p_s \) ("pseudo" collision), the process is repeated until the neutral is either ionised, or leaves the plasma. For a more detailed discussion of this method, the reader is referred to Ref. 6.

The details of all particles are then summarised by the quantities \( H(s) \) and \( f_s \), where \( f_s \) is the total energy loss due to neutrals failing to ionise ("shine through") and \( H(s) \) is defined such that

\[
(1/V_p) \int H(s) dV = 1 - f_s \quad (6.7)
\]

where the integral is over the whole plasma volume \( V_p \). The function \( H(s) \) is thus a function of the flux coordinate \( s \), and corresponds to the \( H(R) \) profile generally referred to in the literature. We define a separate \( H(s) \) and \( f_s \) for each beam and energy species, and then sum them to give global values. If, for any one source, the fraction of "shine through" neutrals exceeds a prescribed limit, \( f_s(\text{max}) \), the program will abort with a message:

EXCESSIVE SHINE THROUGH

6.2.2 Finite Larmor Radius Effects

The exact orbit following (FLR) routine is activated by setting the logical variable

\[ \text{NLFLR} = \text{.TRUE.} \]

The ion space and velocity coordinates are advanced according to Eq. (5.7), using a "leap frog" difference scheme proposed by Buneman (see Ref. 17). Velocity coordinates are defined at half-timesteps \( v^{n+1/2} \), while space coordinates are defined at full timesteps \( x^n \). We then advance the velocity and position of the ion according to

\[
v^{n+1/2} = v^n + (v^n + v^{n+1}) \times 0.5 \times \Omega_e \Delta t \quad (6.8)
\]

\[x^{n+1} = x^n + v^{n+1/2} \Delta t \quad (6.9)\]

where \( \Omega_e = q B/m \) is the ion gyro-frequency. It may readily be observed that this scheme conserves \( v^2 \). In order to ensure that Larmor radius is conserved, it is necessary to replace the term \( 0.5 \times \Omega_e \Delta t \) in Eq. (6.8) by \( \tan(0.5 \times \Omega_e \Delta t) \). For details, the reader is referred to Ref. 17. The timestep \( \Delta t \) is governed by the condition,

\[\Delta t = \xi_e /\Omega_e \quad (6.10)\]

where \( \xi_e \) is a parameter set by the user.

In order to allow for ionisation of the fast ion through its
charge states, we select a random number \( \xi_q \) for each particle and if during the course of its trajectory
\[
\sum_{m} \Delta t_m < \delta \text{ v} > n_q > -\ln(\xi_q) \tag{6.11}
\]
where \( < \delta \text{ v} > n_q \) is the rate coefficient for electron impact ionisation from charge state \( k \) to \( k+1 \) during timestep \( m \), the ion is assumed to lose a further electron, and its charge state \( Z \) is increased accordingly. As soon as \( Z = Z_0 \), the calculation is terminated, and details of the ion coordinates stored for processing in the guiding centre code. The H(s) profile computed from Eq.6.7 thus represents the initial distribution of ions in charge state \( Z_0 \).

### 6.2.3 Sputtering Coefficients

The sputtering coefficients for molybdenum and iron are tabulated in BLOCK DATA, a full description of which is given in Appendix 1. These coefficients are a function of the neutral (or ion) energy, the metal and neutral (or ion) species. The sputtering current is then
\[
i^0_S = \sum_j \int_0 f_S(j,k) c(k,j,n) I(j) \tag{6.12}
\]
where \( I_S \) (amp) is the sputtering current, \( f_S(j,k) \) is the shine through for source \( j \) and energy species \( k \), \( c(k,j,n) \) is the sputtering coefficient from metal \( n \) and \( I(j) \) is the current from source \( j \).

### 6.2.4 Ion Source for Guiding Centre Code

The coordinates of a subset \( N_e \) of fast ions whose creation is computed in this section of the code is stored (Table 7) for later use in the guiding centre code. The birth positions in cylindrical coordinates \( (R_i, \phi_i, z_i) \), the energy \( E \) and the cosine of the "pitch angle" \( \gamma_i = v_i \cdot \nu_i \) (Eq.5.12), are stored for each ion. Further, each ion is assigned a statistical weight, \( c_i \), such that
\[
u_i = I_i / (I_e \xi_i)
\]
where \( I_e \) is the average current from all sources and \( \xi_i \) is the statistical "importance" of the flux surface in which the ion is born. The importance of each flux surface reflects the degree of interest in this region of the plasma. If \( \xi(j) > 1 \), then the flux surface \( j \) is considered to require a more detailed statistical treatment.

If an ion is born on a flux surface where \( \xi > \xi_5 \), it is retained or rejected by selecting a random number \( \xi_q \). If \( \xi_q > \xi_5 \) the ion is retained, while if \( \xi_q < \xi_5 \) the ion is rejected ("Russian roulette"). If on the other hand \( \xi \) is an integer greater than one, a set of \( \xi \) identical ion coordinates are stored ("splitting" technique). Thus, the statistical weights may be varied about their default values of approximately 1, for beams of approximately equal currents.
6.3 Ion Slowing Down and Power Deposition in Plasma

6.3.1 Guiding Centre Motion

Computations of fast ion guiding centre motion and slowing down are performed if the logical variable (Table 5)

\[ \text{NLGUID} = \text{.TRUE.} \]

The ion space and velocity coordinates are advanced according to Eqs. (5.8) and (5.11), using a second order Runge-Kutta method. For a given set of variables \( \mathbf{y}(t) \) satisfying the set of differential equations

\[ \frac{d\mathbf{y}(t)}{dt} = \dot{\mathbf{y}}(\mathbf{y}(t)) \]  \hspace{1cm} (6.13)

the solution is advanced according to

\[ \mathbf{y}(t + \Delta t_C) = \mathbf{y}(t) + \frac{1}{2} \Delta t_C \left( \dot{\mathbf{y}}(\mathbf{y}) + \dot{\mathbf{y}}^\wedge(\mathbf{y}) \right) \]  \hspace{1cm} (6.14)

where

\[ \dot{\mathbf{y}}^\wedge(t) = \dot{\mathbf{y}}(t) + \Delta t_C \ddot{\mathbf{y}}(\mathbf{y}(t)) \]  \hspace{1cm} (6.15)

The timestep is governed by the condition

\[ \Delta t_C = \frac{\varepsilon_C 2 \pi R_0}{v_\parallel + v_\perp + v_\text{min}} \]  \hspace{1cm} (6.16)

where \( \varepsilon_C \) and \( v_\text{min} \) are parameters set by the user. The parameter \( v_\text{min} \) has been introduced to prevent the timestep from growing too large for ions with \( v \) close to zero (i.e. "trapped" ions).

6.3.2 The Fokker-Planck Equation

In order to compute the effects of the fast ion interactions with the plasma, a "test particle" solution to Eq. (5.12) is required (cf. Ref. 7).

A typical ion will be displaced in velocity space according to

\[ < \delta v / \delta t > = -v(1 - T_b/E_b) / 2 \tau_b - v(1 + T_b/2E_b) / 2 \tau_b \]  \hspace{1cm} (6.17)

\[ < \delta v_t / \delta t > = -2 \nu_b \gamma \]  \hspace{1cm} (6.18)

The two expressions on the RHS of (6.17) represent fast ion power deposition to plasma electrons and ions respectively. The ion will be further subject to a diffusion spreading, represented over a time interval \( \Delta t_p \) by

\[ < \Delta v^2 > = \nu \Delta t_p \left( T_b / E_b \tau_b + T_i / E_b \tau_i \right) \]  \hspace{1cm} (6.19)

\[ < \Delta v_t^2 > = 2 \nu_b \Delta t_p \left( 1 - \gamma^2 \right) \]  \hspace{1cm} (6.20).
The diffusion in velocity space is thus represented by a pair of Gaussian distributions in \( v \) and \( \delta \), with half widths \( <\Delta v^2> \) and \( <\Delta \delta^2> \) respectively. This process, however, requires a large amount of computer time, and we have adopted a method suggested by Boozer (Ref. 18). The velocity coordinates are advanced in time according to

\[
v' = v - \Delta t_p \frac{\partial v}{\partial t} + \kappa_1 <\Delta v^2> . \quad (6.21)
\]

\[
\delta' = \delta - \Delta t_p \frac{\partial \delta}{\partial t} + \kappa_2 <\Delta \delta^2> . \quad (6.22)
\]

where \( v \) and \( \delta \) represent values at time \( t \), \( v' \) and \( \delta' \) are evaluated at time \( t + \Delta t_p \), and the coefficients \( \kappa_1 \) and \( \kappa_2 \) are either +1 or -1 depending on whether a random number selected for each process is greater or less than 1/2. This process is repeated for each ion until it is either thermalised \( (E = 3T_i/2) \) or leaves the plasma and strikes the wall. The effect of the diffusion terms is only included if the logical variable

\text{NLDIFF} = \text{TRUE}.

and the user is thus able to remove the "Monte-Carlo" sections of the code in order to reduce the problem from a statistical to a deterministic one.

The timestep \( \Delta t_p \) used in this section of the code is such that

\[
\Delta t_p = \chi_p \Delta t_v
\]

where \( \chi_p \) is a "speed up" factor set by the user. Possible "anomalous" effects may be introduced using the constants \( c_1 \), \( c_2 \), and \( c_3 \), where

\[
\tau_e^* = c_1 \tau_e, \quad \tau_i^* = c_2 \tau_i, \quad \tau_p^* = \tau_p / c_3
\]

6.3.3 Charge Exchange and Ionisation

The fast ions are subject to charge exchange with background neutral particles and may either leave the system or re-ionise elsewhere in the plasma. For hydrogen ions, Eq. 5.5 is again used to compute the charge exchange cross sections and the algorithm described in Section 6.2.2 (Eq. (6.11)), with \( b_{\text{ex}} n \) replacing \( <\Delta v^2> \) is used to determine charge exchange events.

If charge exchange occurs, the resulting fast neutral is "split" into \( N_e \) particles with identical coordinates in velocity space and with statistical weights \( \omega_i / N_e \). They are then given an initial direction, randomly selected from the cone \( \gamma = \gamma_e / v \), and then treated as "re-injected" neutrals in the manner described in Section 6.2. If a neutral particle re-ionises, its new coordinates are stored and its further motion through the plasma computed. If the statistical weight of a particle falls below a minimum \( \omega_{\text{min}} \), set by the user, the splitting process is discontinued and the neutral is given only one chance to re-ionise or be lost.
In the case of heavy ions, an identical procedure to that described in Section 6.2.2 is followed, until the ion is fully ionised or thermalises in the plasma. Only ionisation from lower to higher ionisation states is allowed in this process.

6.3.4 Particle and Power Balance

The quantities computed in the Guiding Centre code to describe the distribution of injected power in the plasma are listed in Table 7. The deposition of power from the fast ions to the plasma ions and electrons, \( p_\alpha(s) \) and \( p_\beta(s) \) respectively, are stored as a function of the flux surfaces \( s \). The fractional power losses due to unconfined orbits, \( f_0 \), and charge exchange, \( f_\lambda \), are also computed and stored.

In the case of heavy ions, a function \( H(s,k) \) is computed, where \( s \) represents flux surfaces and \( k \) the charge state of the ion. This function represents the distribution of thermalised ions in the plasma, and is derived from a particle balance, as distinct from the power distribution used to compute \( H(s) \). \( H(s,k) \) is normalised such that

\[
\left( \frac{1}{V_\alpha} \right) \int_{s_\lambda} H(s,k) \, dV = 1 - f_{\text{loss}} \quad (6.23)
\]

where \( f_{\text{loss}} \) represents the total fraction of particles which are lost to the system during the initial injection, \( f_\lambda \), and slowing down of the fast ions, \( f_0 \).

Finally, the sputtering current \( I_s \) due to particle losses during fast ion slowing down is computed in an identical manner to that described in Section 6.2.3 (Eq.6.12).

6.4 Numerical Functions and Control Variables

The variables required for control of the numerical schemes are listed in Table 5 and default values are given in Table 6.

A variable NRAN is also included, which initialises the random number generator RNFL by calling it NRAN times in (1.5) AUXVAL. This enables the user to control statistical variations in the results. The function (2.1) GAUSS uses the Box-Muller technique (Ref.15) to select from a Gaussian distribution in the beamline code (Section 6.1).
7. Output Package

7.1 General

The Output Package consists of a number of routines for line printer output, activated by setting the logical variable (Table 10)

\[ \text{NLPRT} = \text{.TRUE.} \]

The functions of the various subroutines are listed in Section 2.6. Output from any of these subroutines may be suppressed by setting the OLYMPUS logical variable

\[ \text{NLOMT3(k)} = \text{.TRUE.} \]

where \( k \) is the index of the routine to be suppressed. The meaning of most of the printout provided should be self evident from the text. However, where ambiguities may occur, a fuller description of some output quantities is provided in the following sections.

The first page of output lists four lines of comments supplied by the user (see Appendix 2) and a list of all NAMELIST variables in standard format.

7.2 Beamline Code Output

In subroutine (3.3) BEAMP, all beamline data required for the running of the code is printed. The first page is devoted to individual source details, and the following computed quantities are also printed:

- Power from neutral source \( P_N \) Eq (3.2)
- Power entering torus \( P_T \) Eq (3.3)
- Fraction scraper loss \( f_L \) Eq (3.6)
- Neutral energy fractions \( p_n \) Eq (3.4)

The following pages deal with details of each beamline, including power losses to the scrapers, while the final page describes the geometry of the beamline coupling with the torus. Details of the quantities printed are given in Section 3. Finally, a summary of
total power and neutral current entering the torus, together with total power losses, is printed.

7.3 Plasma Details

In subroutine (3.5) PLASMP all relevant details regarding the plasma are printed. The equations selected for \( n_e, T_e, T_i \) (Eq. 4.1 or 4.2) and \( n_0 \) (Eq. 4.7) are printed, together with profiles of all relevant quantities described in Section 4 as a function of the flux function \( s/s(N_e) \).

Finally, the following useful quantities are printed:

\[
\text{Plasma electron energy} = \int_{s_j}^{s_{j-1}} n_e \cdot T_e \cdot dv_j
\]

\[
\text{Plasma ion energy} = \int_{s_j}^{s_{j-1}} n_i \cdot T_i \cdot dv_j
\]

\[
\int_{ndl} = \int_{s_j}^{s_{j-1}} n_e \cdot ds_j / S(N_e)
\]

where \( dv_j \) is the volume enclosed between flux surfaces \( s_j \) and \( s_{j-1} \), and \( ds_j = s_j - s_{j-1} \).

7.4 Magnetic Field

Magnetic field details are printed in subroutine (3.5) FIELDP. This routine will normally need to be adapted to the plasma experiment being modelled. The default routine is appropriate to the W7-AS configuration, where the toroidal field, \( B_T \), the vertical field, \( B_V \), and the set of coefficients computed by Domaschek (Ref. 2) are printed.

7.5 The Numerical Scheme

Numerical control variables are printed in subroutine (3.6) NUMERP. Quantities printed should be self explanatory from Section 6 and Table 5.

7.6 H(R) Profiles

The \( H(R) \) (or \( H(s) \)) profiles discussed in Section 6.2 are printed in subroutine (3.7) HSUBRP, firstly for each neutral source and energy species separately, and then they are summed to give a total power deposition for the birth profile of fast ions and the total "shine through". In addition the integral of \( H(s) \) contained within each flux surface \( s \), is printed, where

\[
\tilde{H}(s) = \frac{1}{V_p} \int H(s) \cdot dv
\]
Profiles from individual sources will not be printed if the logical variable (Table 10)

\[ \text{NLTTY} = \text{.TRUE.} \]

Finally, the sputtering current, \( I_s \) (Eq. 6.12) due to the escaping neutrals striking the wall is printed.

### 7.7 Guiding Centre Code

Details of the final power balance after slowing down of the fast ions are printed from subroutine (3.8) GUIDCP. Profiles of power deposition to electrons, \( p_e(s_j) \), and ions, \( p_i(s_j) \), (in Watt/cc), are printed, together with the total power injected into the torus. The initial \( H(s) \) profile is also reprinted for comparison with the "Power Deposition H(R)". An energy balance is also printed, including the fraction of ion power corresponding to ions whose slowing down was not completed at the time the job was terminated. The sputtering current due to lost ions and charge exchange neutrals, \( I_s \), is printed, followed by the total sputtering current, including that due to "shine through" neutrals.

The final page of output is only printed when heavy ions are being followed, and gives a particle deposition profile and particle balance for all states of thermalised ions (Eq. 6.23).
8. Diagnostics and Restart Facility

This section describes subroutines used in the case of abnormal termination of the code, (4.3) CRASH and (5.4) ERROR. The former is called from the Guiding Centre Code if CPU Time Limit for the job is exceeded, while the latter is used to abort the code in the case where intermediate computations indicate errors in input data supplied.

If power losses in the beamline exceed a prescribed limit (Section 6.1), a message is printed:

BEAMLINE FOCUSSING FAULTY
BEAM NO k

If "shine through" losses exceed a prescribed limit (Section 6.2), a message is printed:

EXCESSIVE SHINE THROUGH
BEAM NO k

If the user attempts to compute an ionisation or charge exchange process which has not been tabulated, a message is printed.

CROSS SECTIONS FOR THIS PROCESS ARE NOT PROGRAMMED

In order to make use of the recovery facility CRASH, the user must set the logical variable (Table 10)

NLPRV = .TRUE.

In the event that time limit is exceeded, the print routines HSUBRP and GUIDCP are called, and the fast ion details necessary to restart the run are written onto a CRAY Permanent File RESTART, using Subroutine

(8.5) SUBROUTINE RECORD

which is also used to read from the file when the job is restarted. The job may be restarted using the Control Card Deck listed in Appendix 2. The user must re-submit the original DATA set, including the logical variable

NLREST = .TRUE.

If more than one job is to be restarted, the ASSIGN statement in the Control Deck must be modified to include the Edition Number (ED=) of the file from which the job is to be restarted.
9. Tables of COMMON Variables

The following tables list all COMMON variables considered relevant to running of the FAFNER Code. Default values for all non zero input variables are given in separate tables.

In the Tables, the following subscripts are used:

Table 1: Beamline Code
- \( j \) the beam number,
- \( k \) represents the energy species,
- \( m \) the aperture number,
- \( n \) the beamline number.

Table 3: Plasma and Magnetic Field
- \( j \) represents the flux coordinate
- \( j, j, j \_b \) arrays in \((R, \phi, z)\)
- \( j_v \) the variable number (Section 4.3)
- \( k \) ion species in plasma

Table 5: Numerical scheme
- \( i \) represents the particle number
- \( j \) represents the flux coordinate
- \( k \) represents the energy species.

Parameters marked (#) are for internal use in code only

In the list of default parameters, only those variables which are preset to be non zero are included. Logical variables are assumed .FALSE. unless otherwise included.
Table 1: Neutral Beam Parameters

<table>
<thead>
<tr>
<th>Variable</th>
<th>Symbol</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB#</td>
<td>$A_b$</td>
<td></td>
<td>Atomic weight of injected neutrals</td>
</tr>
<tr>
<td>NBLINE(n)</td>
<td>$n_s$</td>
<td></td>
<td>Number of sources per beamline</td>
</tr>
<tr>
<td>NSINJ</td>
<td>$N_J$</td>
<td></td>
<td>Species injected (Ref. Table 8)</td>
</tr>
<tr>
<td>NTBEAM#</td>
<td>$n_b$</td>
<td></td>
<td>Total number of neutral sources</td>
</tr>
<tr>
<td>NTBLIN</td>
<td>$N_T$</td>
<td></td>
<td>Total number of beamlines</td>
</tr>
<tr>
<td>XZBF#</td>
<td>$Z_b$</td>
<td></td>
<td>Atomic number of injected neutrals</td>
</tr>
<tr>
<td>ALFAB(j)</td>
<td>$\alpha$</td>
<td>deg.</td>
<td>Angle of beam to x axis in plane x-y</td>
</tr>
<tr>
<td>BETAB(j)</td>
<td>$\beta$</td>
<td>deg.</td>
<td>Angle of beam to horizontal</td>
</tr>
<tr>
<td>BHGHT(j)</td>
<td>$h_b$</td>
<td>cm.</td>
<td>Vertical dimension of source</td>
</tr>
<tr>
<td>BWDTH(j)</td>
<td>$w_b$</td>
<td>cm.</td>
<td>Width of source</td>
</tr>
<tr>
<td>NLBEAM(j)</td>
<td></td>
<td></td>
<td>.TRUE. if source j switched on</td>
</tr>
<tr>
<td>NLBTYP(n)</td>
<td></td>
<td></td>
<td>.TRUE. if beamline n differs from beamline n-1</td>
</tr>
<tr>
<td>NSHAPE(j)</td>
<td></td>
<td></td>
<td>Shape of source (1.circle 2.rectangle 3.pencil)</td>
</tr>
<tr>
<td>PHID(n)</td>
<td>$\phi_0$</td>
<td>deg.</td>
<td>$\phi$ coordinate of injection duct</td>
</tr>
<tr>
<td>QX(j)</td>
<td>$x_q$</td>
<td>cm.</td>
<td>x coordinate of source</td>
</tr>
<tr>
<td>QY(j)</td>
<td>$y_q$</td>
<td>cm.</td>
<td>y coordinate of source</td>
</tr>
<tr>
<td>QZ(j)</td>
<td>$z_q$</td>
<td>cm.</td>
<td>z coordinate of source</td>
</tr>
<tr>
<td>RMJD(n)</td>
<td>$R_0$</td>
<td>cm.</td>
<td>Major radius of injection duct</td>
</tr>
<tr>
<td>RMJS(n)</td>
<td>$R_S$</td>
<td>cm.</td>
<td>Major radius point S (Fig 2a)</td>
</tr>
<tr>
<td>RTANG(j)#</td>
<td>$R_T$</td>
<td>cm.</td>
<td>Radius of tangency of source</td>
</tr>
<tr>
<td>THETAI (n)</td>
<td>$\theta_i$</td>
<td>deg.</td>
<td>Angle of x-axis to OD (Fig 2a)</td>
</tr>
<tr>
<td>THETAT (j)</td>
<td>$\theta_r$</td>
<td>deg.</td>
<td>Rotation of source</td>
</tr>
</tbody>
</table>
(3) Ion Source

BAMPS(j)  \( I_s \) Amp.  Current in ion source
BEFF(j,k)  \( E \) Efficiency of source
EBEAM(j)  \( V_s \) volt Voltage on ion source
or E eV.  Full energy of neutral beam
EFFNEU(k) \( \eta \) Neutralisation efficiency
PIFRAC(j,k) \( p_i^+ \), \( H^+ \), \( H_2^+ \), \( H_3^+ \) power fractions

(4) Neutral Beam currents, power, energy

BAMP(j)#  \( I_s \) Amp.  Neutral Beam current
BAMPT#  \( I_T \) Amp.  Total neutral current, all sources
P0#  \( P_T \) watt  Total neutral power, all sources
PEFRAC(j,k) \( P_i^+ \) Neutral power fractions E,E/2,E/3
PINJ(j)  \( P_i^+ \) watt  Neutral beam power entering torus
PINJN(j)  \( P_i^+ \) watt  Power from source after neutraliser
PNFRAC(j,k)# \( P_i^+ \) Neutral species particle fractions

(5) Beam focussing, divergence

BDVGHZ(j)  \( \delta_H \) deg.  Horizontal beam divergence
BDVGVVT(j)  \( \delta_V \) deg.  Vertical beam divergence
BHZFOC(j)  \( F_H \) cm.  Horizontal focal length
BVTFOC(j)  \( F_V \) cm.  Vertical focal length
SIGHZ(j)#  \( \delta_H^1 \) rad.  Gaussian horizontal divergence
SIGVT(j)#  \( \delta_V^1 \) rad.  Gaussian vertical divergence
### Scrapers and Ducts

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGHGT(n,m)</td>
<td>$h_n$</td>
</tr>
<tr>
<td>ALFAA(n,m)</td>
<td>$\alpha_n$</td>
</tr>
<tr>
<td>AWDTH(n,m)</td>
<td>$w_n$</td>
</tr>
<tr>
<td>BETAA(n,m)</td>
<td>$\beta_n$</td>
</tr>
<tr>
<td>FLOSS(j)#</td>
<td>$f_j$</td>
</tr>
<tr>
<td>PBLOST#</td>
<td>watt</td>
</tr>
<tr>
<td>PSLOSS(n,m)#</td>
<td>watt</td>
</tr>
<tr>
<td>NLSLCP(j,m)</td>
<td>TRUE, if beam l uses aperture m</td>
</tr>
<tr>
<td>NSCRAP(n,m)</td>
<td>Shape of scrapers (1 Circle 2 Ellipse 3 Rect. 4 W7AS)</td>
</tr>
<tr>
<td>NTSCRAP(n)</td>
<td>$n_n$</td>
</tr>
<tr>
<td>THETAA(n,m)</td>
<td>$\theta_n$</td>
</tr>
<tr>
<td>XAP(n,m)</td>
<td>$x_n$</td>
</tr>
<tr>
<td>XZAP(n,m)</td>
<td>$z_n$</td>
</tr>
<tr>
<td>YAP(n,m)</td>
<td>$y_n$</td>
</tr>
</tbody>
</table>
Table 2: Neutral Beam Default Parameters

(Parameters correspond to H Injection in WVII-AS)

(1) General

NBLINE(1) = 4   NSINJ= 1   NTBLIN = 2

(2) Source Geometry

ALFAB = 3.68  -3.68  -3.68  3.68
BETAB = 2.86  2.86  -2.86  -2.86
BWIDTH = 4x16.0  NSHAPE = 4x1
NLBEAM(2) = .T.  NLBEAM(3) = .T.  NLBEAM(5) = .T.  NLBEAM(8) = .T.
NLBTYP(2) = .T.
PHID = -36.0  36.0
QX = 4x376.5
QY = 23.08  -23.08  -23.08  23.08
QZ = 22.95  22.95  -22.95  -22.95
RMJ0 = 242.0  RMJS = 242.0
THETA1 = 48.0  -48.0

(3) Ion Source

BEFF(J,K) = 1.0
PIFRAC(J,1) = .6  PIFRAC(J,2) = .2  PIFRAC(J,3) = .2

(4) Neutral Beam currents, power, energy

EBEAM = 4x4.0E4  PINJN = 4x3.75E5

(5) Beam focussing, divergence

BDVGHZ = 4x1.2  BDVGVT = 4x1.2
BHZFOC = 4x320.0  BVTFOC = 4x320.0

(6) Scrapers and Ducts

AWIDTH(1,1) = 40.0  AWIDTH(1,2) = 24.0  AWIDTH(1,3) = 31.0
NLSCRAP(1,1) = 12x.T.
NSCRAP(1,1) = 1  NSCRAP(1,2) = 4  NSCRAP(1,3) = 5
NTSCRAP(1) = 3  THETAA(1,2) = 19.0  THETAA(1,3) = 19.0
XAP(1,1) = 105.0  XAP(1,2) = 84.5  XAP(1,3) = 42.5

(7) Magnetic Field Configuration

B0 = gauss  Toroidal magnetic field on axis
BVERT = gauss  Vertical magnetic field
HSTELL = a value for stellator
### Table 3: Plasma and Magnetic Field Parameters

<table>
<thead>
<tr>
<th>Variable</th>
<th>Symbol</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP(k)#</td>
<td>$A_k$</td>
<td></td>
<td>Atomic weight of plasma ions</td>
</tr>
<tr>
<td>NPIDN(k)</td>
<td>$N_k$</td>
<td></td>
<td>Plasma species (ref. Table 8)</td>
</tr>
<tr>
<td>NSPEC</td>
<td>$n_p$</td>
<td></td>
<td>Number of ion species in plasma</td>
</tr>
<tr>
<td>NZW</td>
<td>$N_D$</td>
<td></td>
<td>Type of metal in torus (Table 9)</td>
</tr>
<tr>
<td>RMJ</td>
<td>$R_M$</td>
<td>cm</td>
<td>Major radius of torus</td>
</tr>
<tr>
<td>RV</td>
<td>$r_V$</td>
<td>cm</td>
<td>Minor radius of torus</td>
</tr>
<tr>
<td>XZP(k)#</td>
<td>$Z_{n_p}$</td>
<td></td>
<td>Atomic number of plasma ions</td>
</tr>
<tr>
<td>HGTP</td>
<td>$h_0$</td>
<td>cm</td>
<td>$z$ coordinate of plasma axis</td>
</tr>
<tr>
<td>NPHIS</td>
<td>$N_{\phi}$</td>
<td></td>
<td>Number of $\phi$ points on mesh</td>
</tr>
<tr>
<td>NRS</td>
<td>$N_R$</td>
<td></td>
<td>Number of $R$ points on mesh</td>
</tr>
<tr>
<td>NZONES</td>
<td>$N_F$</td>
<td></td>
<td>Number of flux surfaces</td>
</tr>
<tr>
<td>NZS</td>
<td>$N_z$</td>
<td></td>
<td>Number of $z$ points on mesh</td>
</tr>
<tr>
<td>PHIS(j,\phi)</td>
<td>$\phi$</td>
<td>rad</td>
<td>$\phi$- coordinate on plasma mesh</td>
</tr>
<tr>
<td>RMAJS(j,R)</td>
<td>$R_m$</td>
<td>cm</td>
<td>R-coordinate on plasma mesh</td>
</tr>
<tr>
<td>RMJ</td>
<td>$R_0$</td>
<td>cm</td>
<td>Major radius of plasma axis</td>
</tr>
<tr>
<td>S(j)</td>
<td>$s(j)$</td>
<td>cm</td>
<td>Flux coordinate</td>
</tr>
<tr>
<td>VOL(j)</td>
<td>$V_j$</td>
<td>cm</td>
<td>Volume enclosed by $s(j)$</td>
</tr>
<tr>
<td>VOLP#</td>
<td>$V_P$</td>
<td>cm</td>
<td>Total plasma Volume</td>
</tr>
<tr>
<td>XZS(j,z)</td>
<td>$z$</td>
<td>cm</td>
<td>$z$- coordinate on plasma mesh</td>
</tr>
</tbody>
</table>
(3) Plasma Variables

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXALFA(j)</td>
<td>First exponent for variable j</td>
</tr>
<tr>
<td>EXBETA(j)</td>
<td>Second exponent for variable j</td>
</tr>
<tr>
<td>FRACNI(k)</td>
<td>Fraction of species k</td>
</tr>
<tr>
<td>G0(j)</td>
<td>Value of variable j on plasma axis</td>
</tr>
<tr>
<td>GA(j)</td>
<td>Value of variable j on plasma edge</td>
</tr>
<tr>
<td>NLEXPT(j)</td>
<td>.TRUE. if expt.data used, variable j</td>
</tr>
<tr>
<td>NPROF(j)</td>
<td>Type of profile, variable j</td>
</tr>
<tr>
<td>RG(j)</td>
<td>Scaling radius, variable j</td>
</tr>
<tr>
<td>TE(j)</td>
<td>Electron temperature</td>
</tr>
<tr>
<td>TI(j)</td>
<td>Ion temperature</td>
</tr>
<tr>
<td>XNE(j)</td>
<td>Electron density</td>
</tr>
<tr>
<td>XNIS(j,k)</td>
<td>Ion density, species j</td>
</tr>
</tbody>
</table>

(4) Impurities

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIMP</td>
<td>Atomic weight of effective impurity</td>
</tr>
<tr>
<td>XNI(j)#</td>
<td>Total ion density</td>
</tr>
<tr>
<td>XNIMP(j)#</td>
<td>Effective impurity density</td>
</tr>
<tr>
<td>XZEFF(j)</td>
<td>Zeff of plasma (Eq.4.3)</td>
</tr>
<tr>
<td>XZEFF#</td>
<td>Zeff constant for plasma</td>
</tr>
<tr>
<td>XZIMP</td>
<td>Atomic number of effective impurity</td>
</tr>
<tr>
<td>XZNBAR(j)</td>
<td>Z (Equation 4.4)</td>
</tr>
</tbody>
</table>

(5) Neutral Density

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXPN#</td>
<td>Exponent for neutral density</td>
</tr>
<tr>
<td>RLIM</td>
<td>Radius of limiter</td>
</tr>
<tr>
<td>XNOLIM</td>
<td>Neutral density at limiter</td>
</tr>
</tbody>
</table>

(6) Magnetic Field Configuration

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B0</td>
<td>Toroidal magnetic field on axis</td>
</tr>
<tr>
<td>BVERT</td>
<td>Vertical magnetic field</td>
</tr>
<tr>
<td>MSTELL</td>
<td>m value for stellarator</td>
</tr>
</tbody>
</table>
Table 4: Plasma Default Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPION(1)</td>
<td>2</td>
</tr>
<tr>
<td>NSPEC</td>
<td>1</td>
</tr>
<tr>
<td>NZW</td>
<td>1</td>
</tr>
<tr>
<td>RMJV</td>
<td>200.0</td>
</tr>
<tr>
<td>RV</td>
<td>40.0</td>
</tr>
</tbody>
</table>

(2) Flux Surfaces

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMJ</td>
<td>200.0</td>
</tr>
</tbody>
</table>

(3) Plasma Variables

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXALFA(2)</td>
<td>2.0</td>
</tr>
<tr>
<td>EXALFA(4)</td>
<td>2.0</td>
</tr>
<tr>
<td>EXALFA(5)</td>
<td>2.0</td>
</tr>
<tr>
<td>EXBETA(4)</td>
<td>2.0</td>
</tr>
<tr>
<td>EXBETA(5)</td>
<td>0.5</td>
</tr>
<tr>
<td>FRACNI(1)</td>
<td>1.0</td>
</tr>
<tr>
<td>G(2)</td>
<td>4.0E13</td>
</tr>
<tr>
<td>G(4)</td>
<td>2.5E3</td>
</tr>
<tr>
<td>G(5)</td>
<td>1.6E3</td>
</tr>
<tr>
<td>GA(4)</td>
<td>10.0</td>
</tr>
<tr>
<td>G(5)</td>
<td>10.0</td>
</tr>
<tr>
<td>NPROF(4)</td>
<td>1</td>
</tr>
<tr>
<td>NPROF(5)</td>
<td>1</td>
</tr>
<tr>
<td>NPROF(3)</td>
<td>3</td>
</tr>
<tr>
<td>RG(2)</td>
<td>12.00</td>
</tr>
<tr>
<td>RG(4)</td>
<td>22.31</td>
</tr>
<tr>
<td>RG(5)</td>
<td>22.31</td>
</tr>
</tbody>
</table>

(4) Impurities

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIMP</td>
<td>16.0</td>
</tr>
<tr>
<td>XZIMP</td>
<td>8.0</td>
</tr>
<tr>
<td>XZEFF0</td>
<td>1.5</td>
</tr>
</tbody>
</table>

(5) Neutral Density

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXPN0</td>
<td>0.16</td>
</tr>
<tr>
<td>RLIM</td>
<td>25.0</td>
</tr>
<tr>
<td>XNOLIM</td>
<td>1.0E10</td>
</tr>
</tbody>
</table>

(6) Magnetic Field

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BO</td>
<td>3.0E4</td>
</tr>
<tr>
<td>MSTELL</td>
<td>5</td>
</tr>
</tbody>
</table>
Table 5: Input Data for the Numerical Scheme

<table>
<thead>
<tr>
<th>Variable</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLMAX</td>
<td>$f_L^{(\text{max})}$</td>
<td>Maximum scraper loss (fraction)</td>
</tr>
<tr>
<td>NPBEAM</td>
<td>$N_p$</td>
<td>Number of neutrals per source</td>
</tr>
<tr>
<td>NSBEAM</td>
<td>$N_0$</td>
<td>Number of neutrals stored per source</td>
</tr>
<tr>
<td>FLHMAX</td>
<td>$f_s^{(\text{max})}$</td>
<td>Max. shine through fraction allowed</td>
</tr>
<tr>
<td>NLFLR</td>
<td>.TRUE.</td>
<td>.TRUE. if FLR effects considered</td>
</tr>
<tr>
<td>NLHSSBR</td>
<td>.TRUE.</td>
<td>.TRUE. if H(R) computation required</td>
</tr>
<tr>
<td>NTNEUP</td>
<td>$N_N$</td>
<td>Number of neutrals / source for H(R)</td>
</tr>
<tr>
<td>XCHFL</td>
<td>$\xi_F$</td>
<td>Timestep control, FLR Program</td>
</tr>
<tr>
<td>XZB0</td>
<td>$z_0$</td>
<td>Ion charge after FLR computation</td>
</tr>
</tbody>
</table>

(3) Guiding Centre Code

<table>
<thead>
<tr>
<th>Variable</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTAUE</td>
<td>$c_0$</td>
<td>Anomalous slowing down, electrons</td>
</tr>
<tr>
<td>CTAUI</td>
<td>$c_1$</td>
<td>Anomalous slowing down, ions</td>
</tr>
<tr>
<td>CTAUS</td>
<td>$c_2$</td>
<td>Anomalous pitch angle scattering</td>
</tr>
<tr>
<td>FCHMIN</td>
<td>$\mu_{\text{min}}$</td>
<td>Min. weight for ch. ex. splitting</td>
</tr>
<tr>
<td>NCHXEF</td>
<td>$N_x$</td>
<td>Splitting for charge exchange</td>
</tr>
<tr>
<td>NLDIFF</td>
<td></td>
<td>Velocity diffusion switch</td>
</tr>
<tr>
<td>NLGUID</td>
<td>.TRUE.</td>
<td>.TRUE. if GC Code run</td>
</tr>
<tr>
<td>NRAN</td>
<td></td>
<td>Random number initiator</td>
</tr>
<tr>
<td>NTPART</td>
<td>$N_6$</td>
<td>Number of ions followed in G.C. Code</td>
</tr>
<tr>
<td>XCHGC</td>
<td>$\xi_G$</td>
<td>Timestep control, GC motion</td>
</tr>
<tr>
<td>XCPMIN</td>
<td>$\gamma_{\text{min}}$</td>
<td>Pitch angle control, GC motion</td>
</tr>
<tr>
<td>XIMPTC(j)</td>
<td>$\ell$</td>
<td>Statistical importance of zone j</td>
</tr>
<tr>
<td>XSPEED</td>
<td>$\chi_p$</td>
<td>Speed up factor, power deposition</td>
</tr>
</tbody>
</table>
Table 6: Numerical Scheme Default Parameters

1. **Beamline Code**
   - FLMAX = 0.5
   - NPBEAM = 2000
   - NSBEAM = 250

2. **H(R) Computation**
   - FLHMAX = 0.5
   - XCHFL = 0.05
   - XZBO = 1.0

3. **Guiding Centre Code**
   - CTAUE = 1.0
   - CTAUI = 1.0
   - CTAUS = 1.0
   - FCHMIN = 0.2
   - NCHXEF = 5
   - NLDIFF = T.
   - NRAN = 1000
   - NTPART = 100
   - XCHGC = 0.005
   - XCPMIN = 0.005
   - XIMPTC = 105*1.
   - XSPEED = 10.
Table 7: Numerical Scheme Computed Quantities

1) Neutral Source for H(R)

- BNX(i): x-coord. of atoms stored for heating
- BNY(i): y-coord. of atoms stored for heating
- BNZ(i): z-coord. of atoms stored for heating
- BNVX(i): v component (normalised) of atom
- BNYY(i): v component (normalised) of atom
- BNVZ(i): v component (normalised) of atom

2) Collision Cross Sections

- CLAMDA(k,j): \( \lambda_{\text{cm}^{-1}} \) 1/Total collision mean free path
- CLMCHX(k,j): \( \lambda_{\text{cm}^{-1}} \) 1/Charge exchange mean free path
- XLMIN(k): \( \lambda_{\text{cm}} \) Minimum mean free paths in plasma

3) H(R) and Shine Through Details

- FSHINE: Particle shine through fraction
- HR(j): H(s) Total H(R)
- HRB(j,l,k): H(R) for individual beams, energies
- HRLOSS(j,k): Power fraction shine through
- HRLOST: Total power fraction shine through
- SPAMP0(m): \( I_0 \) Amp Sputtering Current, metal m

4) Ion Source for Guiding Centre Code

- BPE(i): Initial fast ion energy
- BPHI(i): Initial \( \phi \) co-ordinate of ion
- BPRCYL(i): Initial R co-ordinate of ion
- BPSIG(i): Cosine of pitch angle (\( v_\parallel /v \))
- BPWHT(i): Statistical weight of ion
- BPZ(i): Initial z co-ordinate of ion
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOLOST</td>
<td>Particle fraction of lost ions</td>
</tr>
<tr>
<td>HRI(k,j)</td>
<td>Particle deposition profile</td>
</tr>
<tr>
<td>SPAMPI(m)</td>
<td>Sputtering current from ions</td>
</tr>
<tr>
<td>WECHEX</td>
<td>Fraction charge exchange loss</td>
</tr>
<tr>
<td>WELSS</td>
<td>Fraction orbit loss</td>
</tr>
<tr>
<td>WPE(j)</td>
<td>$P_e(s)$ Watt/cc Power deposition to electrons</td>
</tr>
<tr>
<td>WPETOT</td>
<td>Watt Total power to electrons</td>
</tr>
<tr>
<td>WPI(j)</td>
<td>$P_i(s)$ Watt/cc Power deposition to ions</td>
</tr>
<tr>
<td>WPITOT</td>
<td>Watt Total power to ions</td>
</tr>
</tbody>
</table>
Table 8: Table of Elements Used in Code

<table>
<thead>
<tr>
<th>Integer</th>
<th>Element</th>
<th>Symbol</th>
<th>A</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hydrogen</td>
<td>H</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Deuterium</td>
<td>D</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Tritium</td>
<td>T</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Helium 3</td>
<td>He</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>Helium 4</td>
<td>He</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>Oxygen</td>
<td>O</td>
<td>16</td>
<td>8</td>
</tr>
</tbody>
</table>

* The integer numbers correspond to NSINJ in Table 1 and NPION in Table 3.

Table 9: Metals for Sputtering

<table>
<thead>
<tr>
<th>N</th>
<th>Metal</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Iron</td>
<td>Fe</td>
</tr>
<tr>
<td>2</td>
<td>Molybdenum</td>
<td>Mo</td>
</tr>
</tbody>
</table>

Table 10: Output and Diagnostic Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLPRNT</td>
<td>.T.</td>
<td>.TRUE. for Line Printer Output</td>
</tr>
<tr>
<td>NLREST</td>
<td>.F.</td>
<td>.T. if job restarted from files</td>
</tr>
<tr>
<td>NLRPRV</td>
<td>.T.</td>
<td>Files saved after CPU time limit</td>
</tr>
<tr>
<td>NLTTY</td>
<td>.F.</td>
<td>.T. for shortened H(R) Output</td>
</tr>
</tbody>
</table>
Acknowledgements

The author is indebted to Dr. J. C. Cordey of the JET Joint Undertaking and Dr. R. G. Goldston and Dr. D. E. Post of Princeton Plasma Physics Laboratory for advice and encouragement over many years in the development of this work. The support of the Neutral Beam Injection Group at IPP in Garching, in particular from Dr. E. Speth, Dr. F. P. Penningfeld and Dr. W. Ott is also gratefully acknowledged.

References

(1) J. P. Christiansen and K. V. Roberts, 1974, Comp. Phys. Comm., 7, 245
(2) W. Domaschk, 1978, IPP Report IPP 0/38
(3) T. H. Stix, 1972, Plasma Physics, 14, 367
(4) J. A. Rome, J. D. Callen and J. F. Clarke, 1974, Nuc. Fus., 14, 141
(7) R. J. Goldston et al., 1981, JCP, 43, 619.
(8) R. Freeman and E. Jones, 1974, CLM-R137, Culham Laboratory
(10) Atomic Data for Controlled Fusion Research, 1977, ORNL 5206 and 5207
(11) A. C. Riviere, 1971, Nuc. Fus., 11, 363
(13) H. L. Bay et al., 1977, J. Appl. Phys., 48, 4722
(14) L. Spitzer, 1962, Physics of Fully Ionised Gases, Interscience, p127
(15) M. E. Muller, 1958, Math. Tab. Aids Comput., 12, 167
(16) S. V. Putvinski, private communication
(18) A. Boozer, Private communication
Appendix 1: Coefficients Tabulated in BLOCK DATA

The BLOCK DATA used in FAFNER is reproduced below. The quantities tabulated are the sputtering coefficients, discussed in Section 5.1.4, the ion species in the plasma, discussed in Section 4.3, the neutralisation efficiencies of the ions which pass from the ion source through the neutraliser, discussed in Section 3.3, and the rate coefficients for ion-electron impact ionisation, discussed in Section 5.1.2. The sputtering coefficients tabulated are for hydrogen and deuterium impact on iron and molybdenum walls as a function of the impact energy (100eV to 100keV). We assume that the coefficients for atom and ion impact to be identical. The names of the elements are also stored as Hollerith variables for use in the Output Routines.

The atomic mass and charge numbers of the elements listed in Table 8, together with their names as Hollerith Variables, are tabulated for use in constructing the plasma model used in the code.

The neutralisation efficiencies are also tabulated for hydrogen injection as a function of energy, (0-200keV). For deuterium injection, a conversion is made in the code so that the coefficients chosen are those for ions whose energies correspond to those of H ions with the same velocity. Tables of rate coefficients for electron impact ionisation with ions are tabulated as a function of electron temperature (0 to 5keV). Rate coefficients for H→H and D→D+→...08+ are tabulated, and addition of other ions, such as He, are readily be programmed using the scheme of Table 8.
BLOCK DATA

BLOCK DATA: 1. Sputtering Coefficients
2. Plasma Species
3. Neutralisation Efficiencies
4. Ion-Electron Impact Cross Sections

Version 1* 01/Sept/84  GGL

Common for Block Data

VERSION 1* 01/Jul/84  GGL
COMMON /COMDAT/
R APLAS, CIONE, CSPUTI, CTE, ESPUT, XEFFB,
R XENGB,
I NELEM, NENGB, NPLAS, NSPUT, NTE, NZPLAS

DIMENSION
R APLAS(6), CIONE(18,13), CSPUTI(14,2,2), CTE(18),
R ESPUT(14), XEFFB(22,1), XENGB(22),
I NELEM(2), NPLAS(6), NZPLAS(6)

APLAS(k)  Atomic weight of element k
CIONE(j,k)  Electron impact rate coefficient, Te(j)
CSPUTI(m,k1,k2) Sputtering coefficient
CTE(j)  Electron Temperatures
ESPUT(m)  Energies for sputtering
XEFFB(m1)  Neutralisation efficiencies
XENGB(m1)  Energies for neutralisation efficiencies
NELEM  Name of metals for sputtering
NENGB  Number of energy bands for neutralisation
NPLAS  Name of plasma species k
NSPUT  Number of energy bands for sputtering
NTE  Number of Temperature bands
NZPLAS(k)  Atomic number of element k

Cl  1.  Sputtering Coefficients

Cl  1.1  Energies (eV)

DATA NSPUT /14/
DATA ESPUT /
. 1.0E2, 2.0E2, 4.0E2, 6.0E2, 1.0E3, 2.0E3,
. 4.0E3, 8.0E3, 1.0E4, 2.0E4, 4.0E4, 6.0E4,
. 8.0E4, 1.0E5 /

Cl  1.2  Names of Elements

DATA NELEM /
. ZHFE, 2HMO /

DATA CSPUTI /

Cl  1.3  Sputtering from Iron

Cl  Ref: Scherzer, B.M.U., J. Vac. Sci. Technol., 1976, 13, 420

Cl  1.3.1  Hydrogen-Iron Sputtering
\begin{verbatim}
. 0.0, 6.0E-4, 3.0E-3, 4.0E-3, 6.0E-3, 8.0E-3,
. 8.0E-3, 6.0E-3, 5.0E-3, 4.0E-3, 2.5E-3, 2.0E-3,
. 1.5E-3, 1.1E-3.

1.3.2 Deuterium-Iron Sputtering

. 6.0E-4, 4.0E-3, 8.0E-3, 1.5E-2, 2.0E-2, 3.0E-2,
. 3.2E-2, 3.8E-2, 2.5E-2, 2.0E-2, 1.2E-2, 9.0E-3,
. 8.0E-3, 6.0E-3.

1.4 Sputtering from Molybdenum

Ref: Bay et al., J. Appl. Phys., 1977, 48, 4722

1.4.1 Hydrogen-Molybdenum Sputtering

. 0.0, 8.0, 2.0E-4, 5.0E-4, 1.0E-3, 2.0E-3,
. 2.0E-3, 2.0E-3, 1.0E-3, 6.0E-4, 3.0E-4,
. 2.5E-4, 2.0E-4.

1.4.2 Deuterium-Molybdenum Sputtering

. 0.0, 6.0E-4, 2.5E-3, 4.0E-3, 6.0E-3, 8.0E-3,
. 8.0E-3, 4.0E-3, 3.0E-3, 1.5E-3, 9.0E-4, 6.0E-4,
. 5.0E-4, 4.0E-4 /

2. Plasma Species

1 H
2 D
3 T
4 He3
5 HE4
6 O

2.1 Names of Elements

DATA NPLAS /
. 3HH, 3HD, 3HT, 3HHE3, 3HHE4, 3HO /

2.2 Atomic Weights

DATA APLAS /
. 1.0, 2.0, 3.0, 4.0, 16.0/

2.3 Atomic Numbers

DATA NZPLAS /
. 1, 1, 1, 2, 2, 8 /

3. Tables of Neutralisation Efficiencies

3.1 Injection Energies (eV)

DATA NENGB /22/
DATA XENGB /
. 0.0, 5.0E3, 1.0E4, 2.0E4, 3.0E4, 4.0E4,
. 5.0E4, 6.0E4, 7.0E4, 8.0E4, 9.0E4, 1.0E5.
\end{verbatim}
3.2 Ideal Neutralisation Efficiencies

DATA XEFFB /

3.2.1 H Injection
Note: For D or T Injection, Coefficients Correspond to energies of H atoms with the same velocity

<table>
<thead>
<tr>
<th>Energy (eV)</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.850</td>
<td>0.910</td>
</tr>
<tr>
<td>0.530</td>
<td>0.450</td>
</tr>
<tr>
<td>0.140</td>
<td>0.025</td>
</tr>
<tr>
<td>0.030</td>
<td>0.005</td>
</tr>
</tbody>
</table>

4. Electron-Impact Ionisation Tables
(Sigma v)

4.1 Electron Temperatures

DATA NTE /18/
DATA CTE /

<table>
<thead>
<tr>
<th>Temperature (eV)</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>70.0</td>
<td>100.0</td>
</tr>
<tr>
<td>700.0</td>
<td>1.0E3</td>
</tr>
<tr>
<td>2.0E3</td>
<td>3.0E3</td>
</tr>
<tr>
<td>4.0E3</td>
<td>5.0E3</td>
</tr>
</tbody>
</table>

DATA CIONE /

4.2 Hydrogen Isotopes
Ref: R. Freeman and E. Jones, C1M-R137, Ulham Lab., 1974

<table>
<thead>
<tr>
<th>Isotope</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>9.5E-13</td>
</tr>
<tr>
<td>T</td>
<td>3.1E-13</td>
</tr>
<tr>
<td>2H</td>
<td>2.2E-13</td>
</tr>
</tbody>
</table>

4.3 Helium Values Not Set

4.6 Oxygen

4.6.1 0 -> 0+

<table>
<thead>
<tr>
<th>Transition</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0E-9</td>
<td>1.0E-8</td>
</tr>
<tr>
<td>8.0E-8</td>
<td>9.0E-8</td>
</tr>
<tr>
<td>9.0E-7</td>
<td>8.0E-7</td>
</tr>
</tbody>
</table>

4.6.2 0+ -> 0 2+

<table>
<thead>
<tr>
<th>Transition</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>3.0E-10</td>
</tr>
<tr>
<td>1.8E-8</td>
<td>3.0E-8</td>
</tr>
<tr>
<td>3.1E-8</td>
<td>2.0E-8</td>
</tr>
</tbody>
</table>

4.6.3 0 2+ -> 0 3+

<table>
<thead>
<tr>
<th>Transition</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>2.0E-11</td>
</tr>
<tr>
<td>6.5E-9</td>
<td>5.0E-9</td>
</tr>
<tr>
<td>1.5E-8</td>
<td>1.4E-8</td>
</tr>
</tbody>
</table>
c
Cl

4.6.4 0 3+ -> 0 4+

.0.0, 1.0E-12, 7.0E-11, 3.0E-10, 7.0E-10, 1.2E-9,
.2.0E-9, 3.0E-9, 5.0E-9, 6.0E-9, 6.7E-9, 7.0E-9,
.7.0E-9, 7.0E-9, 6.0E-9, 5.0E-9, 6.0E-9, 5.0E-9.

Cl

4.6.5 0 4+ -> 0 5+

.0.0, 1.0E-14, 6.0E-12, 5.0E-11, 1.5E-10, 2.5E-10,
.5.0E-10, 9.0E-10, 1.8E-9, 2.3E-9, 2.6E-9, 2.8E-9,
.3.0E-9, 3.0E-9, 3.0E-9, 3.0E-9, 3.0E-9, 3.0E-9.

DATA (ClO(J), J=181,234) /

Cl

4.6.6 0 5+ -> 0 6+

.0.0, 0.0, 6.0E-13, 8.0E-12, 3.0E-11, 6.0E-11,
.1.5E-10, 3.0E-10, 6.0E-10, 8.0E-10, 9.0E-10, 1.0E-9,
.1.1E-9, 1.2E-9, 1.2E-9, 1.2E-9, 1.2E-9, 1.2E-9.

Cl

4.6.7 0 6+ -> 0 7+

.0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.0.0, 7.0E-14, 3.0E-12, 1.4E-11, 2.6E-11, 4.0E-11,
.7.0E-11, 8.0E-11, 1.5E-10, 1.6E-10, 1.6E-10, 1.6E-10.

Cl

4.6.8 0 7+ -> 0 8+

.0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
.0.0, 0.0, 6.0E-13, 6.0E-12, 7.0E-12, 1.1E-11,
.2.0E-11, 2.5E-11, 4.0E-11, 5.0E-11, 5.5E-11, 6.0E-11.
Appendix 2: Job Control Decks for FAFNER

The Job Control Cards necessary to submit a FAFNER run to the Cray are listed below:

```
/JOB R=1024 T=2:00 PR=13
ASSIGN (DN=$OUT,A=FT06)
ASSIGN (DN=$PLOT,A=FT07)
ASSIGN (DN=A,A=FT08)
CFT, DN=HZ, OFF=P.
LD,LIB=IPPLIB, SET=INDEF.
SAVE, DN=A, PDN=RESTART, ID=GGL.
EXIT.
DUMPJOB.
DEBUG, TRACE.
/EOF
$$ GGL: FAFNER
$$ GGL: WVIIAS
$$ GGL: OLYMPUS.CRAY
/EOF
$$ GGL: W7ASRUN.DATA, NOLABEL
/EOF
```

An example of a DATA File is

```
RUN 07/10/84  01
HYDROGEN TEST RUN
RUN FROM START
DEFAULT PARAMETERS
$NEWRUN
NLREST=F.
$END
$$ JOK: AMOSPNCNH. JOK744<20:*>
  2.00E02  3.00E04
$$ FPP: AMOSPNCNH. FPP039
```

It should be noted that in accordance with OLYMPUS convention (Ref.1), the first 4 data cards are used for comments describing the run, and the Input Variables described in Tables 1 through 10 are changed from their default values using the NAMELIST NEWRUN.
The example above is for hydrogen injection. If we wish to study oxygen injection, the DATA file would be changed.

RUN 07/10/84 02
OXYGEN TEST RUN
RUN FROM START
NO SPEED UP, FLR EFFECTS TO Z=4
$NEWRUN
XSPEED=1.0,
NLFLR=T,
XZB0=4,
NTPART=100, NTNEUP=100,
NLREST=F,
NSINJ=6,
$END
$$ JOK:AMOSPNC1.JOK744<20:*> 2.00E02 3.00E04
$$ FPP:AMOSPNC1.FPP099

In the case that the Job Time Limit is exceeded, the SAVE statement on the Control Deck ensures that all necessary information required to restart the job is contained on File RESTART. The job may be restarted by submitting the following Deck.

/JOB R=1024 T=30:00 PR=00
ASSIGN (DN=$OUT, A=FT06)
ASSIGN (DN=$PLOT, A=FT07)
ASSIGN (DN=A, A=FT08)
ACCESS, DN=B, PDN=RESTART, ID=GGL, UQ.
ASSIGN (DN=B, A=FT09)
CFT, DN=HZ, OFF=P.
LDR, LIB=IPPLIB, SET=INDEF.
DELETE, DN=B.
SAVE, DN=A, PDN=RESTART, ID=GGL.
EXIT.
DUMPJOB.
DEBUG, TRACE.
/EOF
$$ GGL:FAFNER
$$ GGL:WVIIAS
$$ GGL:OLYMPUS.CRAY
/EOF
$$ GGL:W7ASRUN.DATANOLABEL
/EOF

and setting NLREST=.TRUE. in the Control Deck.
Fig. 1 Flowchart of Code FAFNER

(1) Light ions
(2) Heavy ions
(3) CPU time limit exceeded
Fig. 2a  Beamline Coordinate System - Mid plane of Torus
**Fig. 2b**  Beamline - Side View

**Fig. 2c**  Tilt of injectors ($x_B$-axis out of page)
Fig. 3 Description of Scapers

Fig. 3a Scraper geometry

Fig. 3b Scraper Shapes