3-74
D-928

UCRL-51598

HYBRID-1, A TWO-DIMENSIONAL FOKKER-PLANCK CODE

A. A. Mirin

June 28, 1974

Prepared for U.S. Atomic Energy Commission under contract No. W-7405-Eng-48

**LAWRENCE
LIVERMORE
LABORATORY**

University of California/Livermore

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

NOTES

The report was prepared as an integral part of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, 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 appearing hereon or for the claims, statements, or representations that its use would not infringe privately owned rights.

Printed in the United States of America
Available from

National Technical Information Service

U. S. Department of Commerce

5285 Port Royal Road

Springfield, Virginia 22151

Price: Printed Copy \$ ____ * Microfiche \$0.95

* Pages

1-50
51-150
151-325
326-500
501-1000

NTIS
Selling Price

\$4.00
\$5.45
\$7.60
\$10.60
\$13.60

TID 4500, UC-20
Controlled Thermonuclear,
Processes

LAWRENCE LIVERMORE LABORATORY
University of California, Livermore, California, 94550

UCRL-51598

HYBRID-I, A TWO-DIMENSIONAL FOKKER-PLANCK CODE

A. A. Mirin

MS. date: June 28, 1974

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy 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.

Contents

Abstract	1
Introduction	1
Discussion	1
Basic Equations	1
Normalization	2
Computation of Ion Rosenbluth Potentials	3
Source Terms	4
Charge Exchange	5
Spatially Dependent Functions	5
Reaction Parameter	5
Ambipolar Potential	6
Boundary Conditions	6
Construction of Ion and Electron Meshes	7
Computation of Coefficients of Ion and Electron Equations	7
Advancement of Ion and Electron Equations	8
General Description of Program and Its Subroutines	9
MAIN Program	9
Subroutines	15
BOUNDARY	15
POTSHAPE	15
DENSINPT	15
GETGF	15
TLOOP	15
WHY	16
OUTPUT	16
AMBI	16
GERR	16
COEF	16
GNANDE	16
XSWEPT	16
PREEL	16
GAMMAIN	16
GELECT	16
MOMENTS	16
IONISOT	16
GETCURE	16
MAXE	17
QHOLD2	17
QFUTCH2	17
EVALS	17
GETFG	17
FINDSIG	17
RESTART	17
OFFMON	17
INTGRL	17
MAXMIN	17

Appendix A. Modification of Ion Loss Boundary	18
Appendix B. 2X Version of HYBRID-1	19
Initialization of Ion Distribution	19
Time Changing Magnetic Field	19
Modifications of Source Terms	19
Extra Subroutines and Variables	20
Appendix C. Listing of HYBRID-1	21
Appendix D. Listing of HYBRID-1-2X	51

HYBRID-1, A Two-Dimensional Fokker-Planck Code

Abstract

The basic equations and methods used to solve the two-dimensional Fokker-Planck equation for ions and electrons are summarized. The ion equation is integrated using a second order, alternating direction,

implicit difference scheme. An implicit algorithm is used to integrate the electron equation. A detailed description of the program and a listing are included.

Introduction

HYBRID-1 solves the two-dimensional Fokker-Planck equation for ions and electrons, with the assumption that the electron distribution is isotropic. The main application of the code is to simulate CTR mirror experiments at LLL.

The velocity and angular finite difference meshes usually consist of 45 and 24 points, respectively, although it is possible to operate with up to 22000 total mesh points on the CDC 7600. The field length

for a 45 by 24 mesh is approximately 170,000g, and the running time is approximately 5 min, depending, of course, on the number of time steps necessary. The code operates without disk files or tapes. The necessary input consists merely of data cards. Both CRT plots and HSP printed data make up the output.

Listings of both HYBRID-1 and HYBRID-1-2X are found in appendices C and D.

Discussion

BASIC EQUATIONS

The ion equation is

$$\frac{1}{\Gamma_{i0}} \frac{\partial f_i}{\partial t} = \left[\frac{1}{2} g_{vv} \right] \frac{\partial^2 f_i}{\partial v^2} + \left[\frac{1}{v^2} g_{v\theta} - \frac{1}{v^3} g_{\theta\theta} \right] \frac{\partial^2 f_i}{\partial v \partial \theta} + \left[\frac{1}{2v^4} g_{\theta\theta} + \frac{1}{2v^3} g_{vv} \right] \frac{\partial^2 f_i}{\partial \theta^2} + \left[\frac{1}{2v^3} g_{\theta\theta} \right]$$

$$+ \frac{\cot \theta}{2v^3} g_{\theta} + \frac{1}{v^2} g_v + \frac{\partial \bar{C}}{\partial v} \left] \frac{\partial f_i}{\partial v} + \left[\frac{1}{v^2} \frac{\partial \bar{C}}{\partial \theta} + \frac{1}{v^4 \sin^2 \theta} \left(1 - \frac{\cos^2 \theta}{2} \right) g_{\theta} - \frac{1}{v^3} g_{v\theta} + \frac{\cot \theta}{2v^3} g_v \right] \frac{\partial f_i}{\partial \theta} + \left[4\pi \left(\frac{\gamma_m f_e}{Z_i^2} + f_i \right) - H_c(v) \right] f_i + S_i / \Gamma_{i0} , \quad (1)$$

where f_i is the ion distribution function, which is symmetric about $\theta = \pi/2$, f_e is the electron distribution function, S_i is the ion source, Z_i is the ion atomic number, $H_e(v)$ is a charge exchange term, γ_m is the mass ratio m_i/m_e , and Γ_{io} is given by

$$\Gamma_{io} = \frac{4\pi v^4 Z_i^4}{m_i^2} \quad (2)$$

The Rosenbluth potential $g(v, \theta)$ is expanded as

$$g = \ln \Lambda_{ii} g_i + \frac{\ln \Lambda_{ie} g_e}{Z_i^2} \quad (3)$$

where

$$\nabla^2 g_i = -8\pi f_i \quad (4)$$

and

$$\nabla^2 g_e = -8\pi f_e \quad (5)$$

The quantity $\ln \Lambda_{ab}$ is defined as

$$\ln \Delta_{ab} = \ln \left\{ \left(\frac{m_a m_b}{m_a + m_b} \right) \frac{2\alpha \lambda_D c}{v^2} \sup_{a,b} \sqrt{\frac{2E}{m}} \right\} - \frac{1}{2} \quad (6)$$

where α is the fine structure constant (taken to be

$1/137$), λ_D is the Debye length $\left(\sqrt{\frac{E_e}{6\pi n_e e^2}} \right)$, and E is

the mean energy of species a or b in ergs. The quantity C in Eq. 1 is given by

$$C = (1 - \gamma_m) \frac{\ln \Lambda_{ie}}{Z_i^2} h_e \quad (7)$$

where

$$\nabla^2 h_e = -4\pi f_e \quad (8)$$

The electron equation is

$$\frac{\partial f_e}{\partial t} = \frac{1}{v^2} \frac{\partial}{\partial v} (\alpha f_e + \beta \frac{\partial f_e}{\partial v}) - \frac{\gamma}{v^2} f_e + S_e \quad (9)$$

where the coefficients are

$$\alpha = 4\pi \Gamma_{eo} \sum_{b=i,e} Z_b^2 \ln \Lambda_{eb} \frac{m_e}{m_b} N(\tilde{f}_b) \quad (10)$$

$$\beta = 4\pi \Gamma_{eo} \sum_{b=i,e} Z_b^2 \ln \Lambda_{eb} \left\{ \frac{v^2 \tilde{M}(\tilde{f}_b)}{3} + \frac{\tilde{E}(\tilde{f}_b)}{3v} \right\} \quad (11)$$

$$\text{and } \gamma = 4\pi \Gamma_{eo} \sum_{b=i,e} Z_b^2 \ln \Lambda_{eb} \Delta_e$$

$$\left(\frac{M(\tilde{f}_b)}{3} + \frac{N(\tilde{f}_b)}{2v} - \frac{\tilde{E}(\tilde{f}_b)}{6v^3} \right) \quad (12)$$

Here, the quantities

$$\Gamma_{eo} = 4\pi e^4 / m_e^2 \quad (13)$$

and

$$\Delta_e = 1 / \log_{10} R_{\text{eff}} \quad (14)$$

where

$$R_{\text{eff}} = R / (1 - e\phi / \frac{1}{2} m_e v^2) \quad (15)$$

and ϕ is the ambipolar potential (if $\frac{1}{2} m_e v^2 \ll e\phi$, then $\Delta_e = 0$). The quantity S_e is the electron source. We define

$$\tilde{f}_i(v) = \frac{1}{2} \int_0^\pi f_i(v, \theta) \sin \theta d\theta \quad (16)$$

$$\tilde{f}_e(v) = f_e(v) \quad (17)$$

and the functionals

$$\tilde{M}(f)(v) = \int_v^\infty f(v) v dv \quad (18)$$

$$\tilde{N}(f)(v) = \int_0^v f(v) v^2 dv \quad (19)$$

$$\tilde{E}(f)(v) = \int_0^v f(v) v^4 dv \quad (20)$$

NORMALIZATION

Table 1 lists real variables, normalized variables corresponding to them, and normalization constants (defined as real variables divided by normalized variables).

Table 1. Real vs normalized variables

Real	Normalized	Normalization constant
n_i	$\tilde{n}_i$	$\tilde{n}$
v	x	$\tilde{v}$
t	τ	$\tilde{t} = \tilde{v}^3 / \tilde{n} \Gamma_{i0}$
f_i	F_i	$\tilde{n} / \tilde{v}^3$
f_e	F_e	$\tilde{n} / \tilde{v}^3$
g	G	$\tilde{n} \tilde{v}$
g_i	G_i	$\tilde{n} \tilde{v}$
C	C	$\tilde{n} / \tilde{v}$
h_e	H_e	$\tilde{n} / \tilde{v}$
M	M	$\tilde{n} / \tilde{v}$
$\tilde{N}$	N	$\tilde{n}$
$\tilde{E}$	E	$\tilde{n} \tilde{v}^2$
S_i	s_i	$s = \tilde{n} / \tilde{v}^3 \tilde{t}$
S_e	s_e	s
α	α	$\tilde{\alpha} = \tilde{v}^3 / \tilde{t}$
β	β	$\tilde{\beta} = \tilde{v}^4 / \tilde{t}$
γ	γ	$\tilde{\gamma} = \tilde{v}^2 / \tilde{t}$
$\tilde{E}_i$	E_i	$\tilde{E} = 1/2 m_i \tilde{v}^2$
$\tilde{E}_e$	E_e	$\tilde{E}$
Γ_{i0}	Γ_{i0}	Γ_{i0}
Γ_{e0}	$\Gamma_{e0} / \Gamma_{i0}$	Γ_i
$\tilde{f}_i$	$\tilde{F}_i$	$\tilde{n} / \tilde{v}^3$
$\tilde{f}_e$	$\tilde{F}_e$	$\tilde{n} / \tilde{v}^3$
g_e	G_e	$\tilde{n} \tilde{v}$

The normalized ion density $\tilde{n}_i$ is defined as

$$\tilde{n}_i = 2\pi \int_0^\infty x^2 dx \int_0^\pi F_i(x, \theta) \sin \theta d\theta \quad (21)$$

and the real ion density n_i is given by

$$n_i = 2\pi \int_0^\infty v^2 dv \int_0^\pi f_i(v, \theta) \sin \theta d\theta \quad (22)$$

The normalized ion energy $\tilde{E}_i$ is defined as

$$\tilde{E}_i = \langle x^2 \rangle = \frac{2\pi}{\tilde{n}_i} \int_0^\infty x^4 dx \int_0^\pi F_i(x, \theta) \sin \theta d\theta \quad (23)$$

and the real ion energy E_i is given by

$$E_i = \frac{1}{2} m_i \langle v^2 \rangle = \frac{1}{2} m_i \tilde{v}^2 \tilde{E}_i \quad (24)$$

The density and energy of the isotropic electron distribution are defined and normalized analogously

(except that $E_e = \frac{m_e}{m_i} \langle x^2 \rangle$).

When transformed completely to normalized variables, Eqs. (1), (3), (4), (5), (7), (8), (9), (10), (11), and (12) still hold.

COMPUTATION OF ION ROSENBLUTH POTENTIALS

Given $F_i(x, \theta)$, we wish to compute $G_i(x, \theta)$, where

$$\nabla^4 G_i = -8\pi F_i \quad (25)$$

We expand in Legendre polynomials:

$$F_i(x, \theta) = \sum_{j=0}^{2M} V_j(x) P_j(\cos \theta) \quad (26)$$

and

$$G_i(x, \theta) = \sum_{j=0}^{2M} W_j(x) P_j(\cos \theta) \quad (27)$$

We define the functionals

$$M_j(f)(x) = \int_x^\infty f(x) x^{-1-j} dx \quad (28)$$

$$N_j(f)(x) = \int_0^x f(x) x^{2+j} dx \quad (29)$$

$$R_j(f)(x) = \int_x^\infty f(x) x^{-3-j} dx \quad (30)$$

and

$$E_j(f)(x) = \int_0^x f(x) x^{4+j} dx \quad (31)$$

The coefficients of Eq. (27) may then be expressed in terms of the coefficients of Eq. (26) by

$$W_j(x) = \left(\frac{-4\pi}{2j+1} \right) \left\{ \left(\frac{1}{2j+3} \right) \left(x^{-j-1} E_j(V_j) + x^{j+2} M_j(V_j) \right) - \left(\frac{1}{2j-1} \right) \left(x^{-j+1} N_j(V_j) + x^j R_j(V_j) \right) \right\} \quad (32)$$

Derivatives of W_j , which are used in computing derivatives of G_i , are calculated by differentiating under the integral sign.

The quantity G_e (which satisfies $\nabla^4 G_e = -8\pi F_e$) and its derivatives are similarly computed, but only a trivial Legendre expansion is required since F_e is independent of θ . The term $\frac{\partial H_e}{\partial x}$, which is used in $\frac{\partial C}{\partial x}$, is given by

$$\frac{\partial H_e}{\partial x} = \frac{-4\pi}{x^2} N(\bar{F}_e) \quad (33)$$

SOURCE TERMS

There can be several ion source terms: each one is of the form

$$S_i(x, \theta) = J_i \cdot S_i(x, \theta) / \bar{S}_i \bar{S}_i^3 \quad (34)$$

where J_i is the current in particles/cm³ and S_i is a shape function. The term $\bar{S}_i$ is given by

$$\bar{S}_i = 2\pi \int_0^\infty x^2 dx \int_0^\pi S_i(x, \theta) \sin \theta d\theta \quad (35)$$

The quantity $S_i(x, \theta)$ is taken to be a product

$$S_i(x, \theta) = S_1(x) S_2(\theta) \quad (36)$$

where $S_1(x)$ is a Gaussian in x , and $S_2(\theta)$ is either a Gaussian in θ centered at $\theta = \pi/2$, a Gaussian in $\cos \theta$,

or an approximation to a normal mode.¹ The current, J_i , is written as

$$J_i = A_i^{(s)} + B_i^{(s)} \sqrt{n_i} + C_i^{(s)} n_i \quad (37)$$

where n_i is the ion density, and $A_i^{(s)}$, $B_i^{(s)}$, and $C_i^{(s)}$ are constants. The above terms are rewritten in the form

$$s_i(x, \theta) = s_0 S_i(x, \theta) \quad (38)$$

where

$$s_0 = R_{si}(1 + D_{ci} n_i) + D_{12} \sqrt{n_i} \quad (39)$$

$$R_{si} = A_i^{(s)} \bar{i} / \bar{n} \bar{S}_i \quad (40)$$

$$D_{ci} = B_i^{(s)} \bar{i} / A_i^{(s)} \quad (41)$$

and

$$D_{12} = C_i^{(s)} \bar{i} / \sqrt{\bar{n}} \bar{S}_i \quad (42)$$

The electron source term is of the form

$$S_e(x) = J_e S_e(x) / \bar{S}_e \bar{S}_e^3 \quad (43)$$

where J_e and S_e correspond to J_i and S_i . The term $\bar{S}_e$ is given by

$$\bar{S}_e = 4\pi \int_0^\infty x^2 S_e(x) dx \quad (44)$$

The quantity $S_e(x)$ is taken to be a Gaussian, and the current J_e is written as

$$J_e = A_e^{(s)} + B_e^{(s)} \sqrt{n_e} + C_e^{(s)} n_e \quad (45)$$

where n_e is the electron density, and $A_e^{(s)}$, $B_e^{(s)}$, and $C_e^{(s)}$ are constants. The term $C_e^{(s)}$ may be adjusted to take into consideration the ionization of neon by cold electrons.

¹A. H. Futch, Jr., J. P. Holdren, J. Killeen, and A. A. Mirin, *Multi-Species Fokker-Planck Calculations for D-T and D-He³ Mirror Reactors*, Lawrence Livermore Laboratory, Rept. UCRL-73226 (1971) p. 36.

CHARGE EXCHANGE

The coefficient $H_c(v)$ in Eq. (1) is given by

$$H_c(v) = n_0 v \frac{1 - e^{-\beta v^4}}{a + b \ln v/v_1 + c(v - v_1)^2 + d v^8} + \alpha \quad (46)$$

The first term in Eq. (46) represents charge exchange with molecular hydrogen or deuterium, and the second term represents charge exchange with the beam. The quantities n_0 and α are specified constants, and the other variables in Eq. (46) are given by

$$\beta = 3.491 \times 10^{-31} \quad (47)$$

$$a = 1.111 \times 10^{15} \quad (48)$$

$$b = 1.351 \times 10^{14} \quad (49)$$

$$c = 5.099 \times 10^{-2} \quad (50)$$

$$d = 2.092 \times 10^{-53} \quad (51)$$

and

$$v_1 = 1.000 \times 10^8 \quad (52)$$

SPATIALLY DEPENDENT FUNCTIONS

We let $\psi = B/B_0$, where B_0 is the magnetic field at the center of the mirror machine, and we let

$$\Phi_{\max} = \frac{Z_i e \phi}{\frac{1}{2} m_i \tilde{v}^2}. \quad \text{We consider values of } \psi \text{ between 1}$$

and R , where R is the mirror ratio, and values of Φ between 0 and $\Phi_{\max}$. Letting $\mu = \cos \theta$, $w^2 = \psi(1 - \mu^2)$, and $s^2 = x^2 w^2 / x^2 + \Phi$, we compute the densities

$$n_{zi}(\psi, \Phi) = 4\pi \tilde{n} \int_0^\infty x \sqrt{x^2 + \Phi} dx$$

$$\int \frac{\min\left\{1, \frac{x}{\sqrt{x^2 + \Phi}} \sqrt{\psi}\right\}}{\sqrt{x^2 + \Phi}} \sqrt{\frac{\psi}{R}} \frac{s ds F(x, \theta)}{\sqrt{1 - s^2}} \quad (53)$$

and

$$n_{ze}(\psi, \Phi) = 4\pi n \cdot \int \frac{\sqrt{\frac{m_i \Phi_{\max}}{m_e Z_i}}}{\sqrt{\frac{m_j \Phi}{m_e Z_i}}}$$

$$x^2 dx F_c(x) \sqrt{1 - \frac{m_i \Phi}{m_e x^2 Z_i}} + \int \frac{\sqrt{\frac{m_i \Phi_{\max}}{m_e Z_i}}}{\sqrt{\frac{m_j \Phi}{m_e Z_i}}} \quad (54)$$

$$x^2 dx F_c(x) \sqrt{1 - \frac{\psi}{R} + \frac{m_i}{m_e x^2 Z_i} \left(\frac{\psi}{R} \Phi_{\max} - \Phi \right)} \quad (54)$$

Chebyshev quadrature is used for Eq. (53). For each ψ , we find a value of $\Phi = \Phi^*(\psi)$ such that

$$n_{ze}(\psi, \Phi) = Z_i n_{zi}(\psi, \Phi) \quad (55)$$

We then compute the function

$$n_{zi}^*(\psi) = n_{zi}(\psi, \Phi^*(\psi)) \quad (56)$$

The pressures $p_{\parallel}$ and $p_{\perp}$ are also computed. The equation for $p_{\parallel}$ is almost the same as Eq. (53), but a factor $f_{\parallel}$ given by

$$f_{\parallel} = m_i \tilde{v}^2 \left\{ x^2 [1 - \psi(1 - \mu^2)] + \Phi \right\} \quad (57)$$

must be inserted in the integrand. To obtain $p_{\perp}$ one inserts the factor $f_{\perp}$, given by

$$f_{\perp} = \frac{1}{2} m_i \tilde{v}^2 \psi x^2 (1 - \mu^2) \quad (58)$$

Finally, the quantity

$$\psi_{vac}(\psi) = \sqrt{\psi^2 + k_m P_{\perp}(\psi)} \quad (59)$$

is computed, where k_m is the largest number such that ψ_{vac} is non-decreasing.

REACTION PARAMETER $(\sigma v)_{DT}$

If the ion species is deuterium, one can compute $(\sigma v)_{DT}$ by two different methods, where it is assumed

that there exists a tritium distribution at the same density and energy as the deuterium distribution.¹ The simpler method involves merely one-dimensional integration:

$$(\sigma v)_{DT} = \sqrt{\frac{5}{3}} \cdot \frac{\int_0^\infty v^3 \sigma_{DT}(E) \tilde{f}_T(v) dv}{\int_0^\infty v^2 \tilde{f}_T(v) dv} \quad (60)$$

$$\text{where } E = \frac{5}{6} m_i v^2 \text{ (in keV)} \quad (61)$$

The quantity $\sigma_{DT}(E)$ is given by

$$10^{24} \times \sigma = 2.25 \times 10^4 \exp(-44.4/\sqrt{E})/E \quad (62)$$

if $E < 20$,

$$10^{24} \times \sigma = 4.62 \times 10^{-4} E^2 - 0.137 \text{ if} \quad (63)$$

$20 \leq E < 30$,

$$10^{24} \times \sigma = 6.7 \times 10^{-4} E^2 - 0.32 \text{ if } 30 \leq E \quad (64)$$

< 50 ,

$$10^{24} \times \sigma = 8.55 \times 10^{-2} E - 2.93 \text{ if } 50 \leq E \quad (65)$$

< 87 ,

$$10^{24} \times \sigma = 1.98 \times 10^4 / [(E - 107)^2 + 4000] \text{ if} \quad (66)$$

$87 \leq E < 107$,

$$10^{24} \times \sigma = 3.56 \times 10^4 / [(E - 107)^2 + 7200] \text{ if} \quad (67)$$

$107 \leq E < 137$,

$$10^{24} \times \sigma = 8.7 - 0.0314E \text{ if } 137 \leq E < 185 \quad (68)$$

$$10^{24} \times \sigma = 7.05 \times 10^3 / E^{3/2} \text{ if } 185 \leq E < 400 \quad (69)$$

and

$$10^{24} \times \sigma = 255/(E - 110) \text{ if } E \geq 400 \quad (70)$$

The more accurate method for computing $(\sigma v)_{DT}$ involves three-fold velocity integration:

$$(\sigma v)_{DT} = \frac{\int_0^\infty d\hat{s} \int_0^{\hat{s}} u^2 du \int_0^u (\hat{s}^2 - \hat{t}^2) \left\{ \tilde{F}_1\left(\frac{\hat{s} + \hat{t}}{2}\right) \tilde{F}_2\left(\frac{\hat{s} - \hat{t}}{2}\right) + \tilde{F}_1\left(\frac{\hat{s} - \hat{t}}{2}\right) \tilde{F}_2\left(\frac{\hat{s} + \hat{t}}{2}\right) \right\} \sigma(E) d\hat{t}}{\int_0^\infty d\hat{s} \int_0^{\hat{s}} u du \int_0^u (\hat{s}^2 - \hat{t}^2) \left\{ \tilde{F}_1\left(\frac{\hat{s} + \hat{t}}{2}\right) \tilde{F}_2\left(\frac{\hat{s} - \hat{t}}{2}\right) + \tilde{F}_1\left(\frac{\hat{s} - \hat{t}}{2}\right) \tilde{F}_2\left(\frac{\hat{s} + \hat{t}}{2}\right) \right\} d\hat{t}} \quad (71)$$

where σ is given by Eqs. (62-70) and

$$E = \frac{1}{2} m_i \hat{t}^2 \text{ (in keV)} \quad (72)$$

Here, $\tilde{F}_2(v)$ is the tritium distribution appropriately scaled so that its density and energy equal those of deuterium.

These formulae can easily be modified to consider reactions other than the DT reaction.

AMBIPOLAR POTENTIAL

The ambipolar potential ϕ is computed at each time step by requiring charge neutrality. Suppose we are advancing the ion and electron equations from t to $t + \Delta t$. Let

$$d_i = Z_i (n_i(t + \Delta t) - n_i(t)) \quad (73)$$

and

$$d_e = n_e(t + \Delta t) - n_e(t) \quad (74)$$

If $Z_i n_i(t) > n_e(t)$ and $d_i \geq d_e$, we raise ϕ by an amount $\Delta\phi$ and re-advance the distributions from t to $t + \Delta t$. If $Z_i n_i(t) < n_e(t)$ and $d_i \leq d_e$, we lower ϕ by an amount $\Delta\phi$ and re-advance the distribution from t to $t + \Delta t$. This procedure is repeated until neither of the above two tests is positive. If these charge neutrality requirements are not satisfied after a certain specified number of iterations, the calculation is stopped.

BOUNDARY CONDITIONS

We let

$$x_{\min} = x_c / \sqrt{R - 1} \quad (75)$$

where

$$Z_i \phi = \frac{1}{2} m_i \bar{v}^2 x_c^2 \quad (76)$$

Any ions satisfying $x \leq x_{\min}$ are lost; so for such x , $F_i(x, \theta)$ is set to 0. For each x , we compute θ to satisfy the equation

$$\sin^2 \theta = (1 + x_c^2/x^2)/R, \quad (77)$$

and if $0 \leq \theta \leq \bar{\theta}$, $F_i(x, \theta)$ is set to 0. At $x = \infty$, (the last ion mesh point) $F_i(x, \theta)$ is set equal to 0, and at $\theta = \pi/2$, $\frac{\partial F_i}{\partial \theta}$ is set equal to 0.

The boundary conditions for the electron distribution function are the same as the conservation conditions previously described in Ref. 1. The quantity

$$\hat{G} = \alpha F_e(x) + \beta \frac{\partial F_e}{\partial x} \quad (78)$$

is set equal to zero, one half mesh width away from $x=0$ and $x=\infty$ (the last electron mesh point). The limiting case of Eq. (78), as the mesh becomes longer and finer, is

$$\frac{\partial F_e}{\partial x} = 0, \text{ at } x = 0 \quad (79)$$

and

$$F_e = 0, \text{ at } x = \infty. \quad (80)$$

CONSTRUCTION OF ION AND ELECTRON MESHES

The ion velocity mesh $\{x_j\}_{j=1}^J$ ranges from $x=0$ to $x=x_{\max}$, where $x_{\max}$ is an input parameter. The angular mesh $\{\theta_j\}_{j=1}^J$ ranges from θ_{LC} to $\pi/2$, where

$$\sin^2 \theta_{LC} = 1/R, \quad (81)$$

and R is the mirror ratio. One extra angular point θ_{1+1} is provided for reflection at $\pi/2$.

The electron velocity mesh $\{w_j\}_{j=1}^J$ ranges from $x=0$ to $x=x_{\max}$, where $x_{\max}$ is an input parameter chosen independently of and usually greater than $x_{\max}$. The ion and electron meshes are related only in the sense that they refer to the same normalized velocity x . To go from one mesh to the other, one must usually interpolate.

Any of the three meshes described above may be non-uniform. To integrate any mesh function over

two consecutive mesh-widths, an extension of Simpson's technique is used:

$$\int_{z_{i-1}}^{z_{i+1}} f(z) dz \approx C_1 f(z_{i-1}) + C_2 f(z_i) + C_3 f(z_{i+1}), \quad (82)$$

where

$$C_1 = \frac{2h_+^3 - h_+^3 + 3h_-^2 h_+}{6h_+ (h_+ + h_-)}, \quad (83)$$

$$C_2 = \frac{h_+^3 + h_-^3 + 3h_+ h_- (h_+ + h_-)}{6h_+ h_-}, \quad (84)$$

$$C_3 = \frac{2h_+^3 - h_-^3 + 3h_+^2 h_-}{6h_+ (h_+ + h_-)}, \quad (85)$$

$$h_+ = z_{i+1} - z_i, \quad (86)$$

and

$$h_- = z_i - z_{i-1} \quad (87)$$

COMPUTATION OF COEFFICIENTS OF ION AND ELECTRON EQUATIONS

The quantity G_i and its derivatives (see *Computation of Ion Rosenbluth Potentials*) are computed as follows. The V_j are calculated using trapezoidal integration with respect to θ . The functionals M_j , N_j , R_j , and F_j are then computed using trapezoidal integration with respect to x . Next, W_j and its first two derivatives are computed using Eq. (32), and the first two derivatives of Π_j are calculated using well known recursion relations. Finally, G_i and its derivatives through order 2 are computed using Eq. (27) and appropriate derivatives of its right hand side, respectively. The G_i and its derivatives are not necessarily recomputed each cycle since they are slowly changing, but they are renormalized by the ion density if it changes.

In computing G_e and H_e , the electron distribution function F_e is interpolated onto the ion mesh, and the integral

$$I_{me} = \int_{x_{\max}}^{x_{\min}} F_e(x) x dx \quad (87)$$

is gotten by integrating F_e over the electron mesh. The quantity $x_{\max}$ is not necessarily an electron mesh point, so interpolation of the integral must be performed at the lower limit. The functionals $M_0(F_e)$, $N_0(F_e)$ and $E_0(F_e)$ are then computed by trapezoidal integration over the ion mesh. In the case of M_0 , integration is performed up to $x_{\max}$, and I_{me} is then added to the integral.

The coefficients α , β , and γ of the electron equation depend on the functionals M , N , and E (which are really the same as M_0 , N_0 , and E_0). To get these functionals trapezoidal integration is used. In the case of the ion-dependent functionals, the ion distribution function F_i is first interpolated onto the electron mesh, and the integration is then performed.

ADVANCEMENT OF ION AND ELECTRON EQUATIONS

We first rewrite the ion equation in the form

$$\begin{aligned} \frac{1}{n_i} \frac{\partial F_i}{\partial \tau} = & (a_1^i + a_1^e) \frac{\partial^2 F_i}{\partial x^2} + a_2^i \frac{\partial^2 F_i}{\partial x \partial \theta} \\ & + (a_3^i + a_3^e) \frac{\partial^2 F_i}{\partial \theta^2} + (b_1^i + b_1^e) \frac{\partial F_i}{\partial x} \\ & + (b_2^i + b_2^e) \frac{\partial F_i}{\partial \theta} + (c^i + c^e) F_i + d, \end{aligned} \quad (88)$$

where

$$a_1^i = \frac{1}{2n_i} \frac{\partial^2 G_i}{\partial x^2} \cdot \ell n \Lambda_{ii}, \quad (89)$$

$$a_1^e = \frac{1}{2n_i} \frac{\partial^2 G_e}{\partial x^2} \cdot \frac{\ell n \Lambda_{ie}}{Z_i^2}, \quad (90)$$

$$a_2^i = \frac{1}{n_i} \left\{ \frac{1}{x^2} \frac{\partial^2 G_i}{\partial x \partial \theta} - \frac{1}{x^3} \frac{\partial G_i}{\partial \theta} \right\} \cdot \ell n \Lambda_{ii}, \quad (91)$$

$$a_3^i = \frac{1}{n_i} \left\{ \frac{1}{2x^4} \frac{\partial^2 G_i}{\partial \theta^2} + \frac{1}{2x^3} \frac{\partial G_i}{\partial x} \right\} \cdot \ell n \Lambda_{ii}, \quad (92)$$

$$a_3^e = \frac{1}{n_i} \cdot \frac{1}{2x^3} \frac{\partial G_e}{\partial x} \cdot \frac{\ell n \Lambda_{ie}}{Z_i^2}, \quad (93)$$

$$b_1^i = \frac{\ell n \Lambda_{ii}}{n_i} \left\{ \frac{1}{2x^3} \left(\frac{\partial^2 G_i}{\partial \theta^2} + \cot \theta \frac{\partial G_i}{\partial \theta} \right) + \frac{1}{x^2} \frac{\partial G_i}{\partial x} \right\}, \quad (94)$$

$$b_1^e = \frac{1}{n_i} \left\{ \frac{1}{x^2} \frac{\partial G_e}{\partial x} + \frac{\partial C_e}{\partial x} \right\} \cdot \frac{\ell n \Lambda_{ie}}{Z_i^2}, \quad (95)$$

$$\begin{aligned} b_2^i = & \frac{\ell n \Lambda_{ii}}{n_i} \left\{ \left(\frac{\csc^2 \theta - \frac{1}{2} \cot^2 \theta}{x^4} \right) \frac{\partial G_i}{\partial \theta} \right. \\ & \left. - \frac{1}{x^3} \frac{\partial^2 G_i}{\partial x \partial \theta} + \frac{\cot \theta}{2x^3} \frac{\partial G_i}{\partial x} \right\}, \end{aligned} \quad (96)$$

$$b_2^e = \frac{1}{n_i} \frac{\cot \theta}{2x^3} \frac{\partial G_i}{\partial x} \cdot \frac{\ell n \Lambda_{ie}}{Z_i^2}, \quad (97)$$

$$c^i = \frac{4\pi F_i \ell n \Lambda_{ii}}{n_i}, \quad (98)$$

$$c^e = \frac{4\pi \gamma_m F_e \ell n \Lambda_{ie}}{Z_i^2 n_i}, \quad (99)$$

and

$$d = s_i/n_i. \quad (100)$$

An alternating direction implicit (ADI) method is then used to advance the equation from τ to $\tau + \Delta\tau$. The difference approximations along with the ADI algorithm have been fully described.² It should be made clear that integrations along $x = \text{constant}$ and $\theta = \text{constant}$ lines begin at the loss-cone boundary rather than at the end of the mesh ($x = 0$ or $\theta = \theta_{LC}$).

The electron equation is advanced using fully implicit differencing. The complete algorithm is described in Ref. 1, Appendix B, where, A, B, C and D replace α , β , γ , and s_e , and the parameter ρ is taken to be 1.

²J. Killeen and K. D. Marx, *The Solution of the Fokker-Planck Equation For a Mirror-Confined Plasma*, Lawrence Livermore Laboratory, Rept. UCRL-71662 (1969).

GENERAL DESCRIPTION OF PROGRAM AND ITS SUBROUTINES

2 shows the variables which are used by MAIN, and which are contained in the labeled common blocks accessible from each subroutine.

MAIN Program

MAIN controls the initialization of the code. Table

Table 2. MAIN program and common block variables.

Variable	Function
IYA	Number of θ meshpoints
JXA	Number of x meshpoints for ions
JXE	Number of x meshpoints for electrons
NSOR	Number of ion sources
MX	M [Eq. (26)] (maximum M allowed)
MMX	$2*MX$
MXP3M	$-2*MX-3$
MXP4	$2*MX+4$
JXEP1	$JXE + 1$
IYA2	$2*IYA - 1$
IYAO2	$IYA / 2$
IYAM1	$IYA - 1$
IYAP1	$IYA + 1$
IYJXA	$IYA + JXA$
NCNCH	No. of time steps which can be simultaneously plotted for time dependent plots
PTIME	Time dependent <i>Real Time</i> array
PDENS	Time dependent <i>Density</i> array
PENGY	Time dependent <i>Energy</i> array
NCH	Index for Time dependent plots
TIME	τ (Table 1)
STOP	Time (seconds) at which code should stop
DT	$\Delta\tau$
AMU	1.67165×10^{-24}
ECON	1.602×10^{-9}
RHOP	$2/\Delta\tau$
PIO2	$\pi/2$
TWOPI	2π
FOURPI	4π
CHARGE	4.803×10^{-10}
KSET	Output every KSET cycles
KWAIT	Recompute $\tilde{G}$ every KWAIT cycles

Table 2. MAIN program and common block variables. (continued)

Variable	Function
K	Index for output
KGC0	Index for recomputing $\tilde{G}$
PI	π
XEMAX	$x_{\max}$ (see <i>Construction of Ion and Electron Meshes</i>)
TWGOPI	$2/\pi$
XMAX	$x_{\max}$ (see <i>Construction of Ion and Electron Meshes</i>)
KGETGFK	Equals 0 until subroutine GETGF is entered once
DTR	Δt
CXRXB(4)	Contains n_0 , neon const., α (see <i>Source Terms and Charge Exchange</i>)
CXLOSS	Number of particles lost due to charge exchange in one cycle
CXLOST	Total number of particles lost due to charge exchange since beam turnoff
HCXR (JXA)	$H_c(v)$
CPB2 (IYAP1,JXA)	$\frac{1}{2} c^i$ [Eq. (98)]
A2P (IYAP1,JXA)	a_2^i [Eq. (91)]
ALPHAP (IYAP1,JXA)	Ion contrib. to α_{ij} (Ref., App. B)
BETAP (IYAP1,JXA)	Ion contrib. to β_{ij} (Ref., App. B)
GAMMAP (IYAP1,JXA)	Ion contrib. to γ_{ij} (Ref., App. B)
DELTAP (IYAP1,JXA)	Ion contrib. to δ_{ij} (Ref., App. B)
SHAPE (IYAP1,JXA,NSOR)	$S_i(x,\theta)$ [Eq. (34)]
DENSE	n_e
ENERGE	E_e
EMOIM	m_e/m_i
PMASS	m_i/AMU
EMASS	m_e/AMU
EES	Electron source energy (keV)
EIONS	$\tilde{E}$ (keV)
TNORM	$\tilde{t}$
VNORM	$\tilde{v}$
DENSNO	$\tilde{n}$
GNORM	n_i
ENERGY	E_i
PNUMB	Z_i
KMAXE	Flag for zero n_e
KGNANDE	Flag for zero n_i

Table 2. MAIN program and common block variables. (continued)

Variable	Function
KAMBI	Flag for unsatisfied charge neutrality condition
NRSTRT	Cycle no. at which job should terminate for later restart
FEMIN	Initial value of $F_e(x_{max})$
JXEMI	JXE-1
JXEM2	JXE-2
GAMII	$\ln \Lambda_{ii}$
GAMIE	$\ln \Lambda_{ie}$
GAMEI	$\ln \Lambda_{ei}$
GAMEE	$\ln \Lambda_{ee}$
XSOE	Controls width of electron source S_e
KNPROBS	Index for problem number
FE (JXA)	$\gamma_m F_e \ln \Lambda_{ie} / Z_i n_i$
CEX (JXA)	$\frac{\partial C}{\partial x} / n_i$
GEX (JXA)	$\frac{\partial G_e}{\partial x} / n_i$
GEXX (JXA)	$\frac{\partial^2 G_e}{\partial x^2} / n_i$
TEMPG(IYJXA)	Temporary storage array
EP(IYJXA)	E_{ij} (Ref. App. B)
FP(IYJXA)	F_{ij} (Ref. App. B)
G(IYAP1,JXA)	$F_i(x, \theta)$
GCO(IYAP1,JXA)	$\ln \Delta_{ii} G_i / n_i$
FEI(JXA)	F_e interpolated on ion mesh
EMOF(JXA)	$M_0(F_e)$ (see <i>Computation of Coefficients of Ion and Electron Equations</i>)
ENOF(JXA)	$N_0(F_e)$ (see <i>Computation of Coefficients of Ion and Electron Equations</i>)
EEOF(JXA)	$E_0(F_e)$ (see <i>Computation of Coefficients of Ion and Electron Equations</i>)
CURCON	1.602×10^{-19}
CURRE	J_e (4.12)
CURR	J_i (4.4)
DCOE	$\bar{\alpha} C_e(s)$
COE	$A_e(s)$

Table 2. MAIN program and common block variables. (continued)

Variable	Function
SOURCE (IYAP1,JXA)	$s_i(x, \theta)$
TOTP	Amount of current injected
POTENT	ϕ
DENST	$\Delta\phi$
IO(JXA)	Nearest θ meshpoint to loss boundary
JO(IYAP1)	Nearest X meshpoint to loss boundary
BRATIO	R
BRATIOI	1/R
MIMAX	Number of allowable retries to satisfy charge neut. requirements
TINJSTOP	Time at which injection should stop
PCURROE	$A_e^{(s)}$ [Eq. (45)]
DPCURROE	$C_e^{(s)}$ [Eq. (45)]
PCURROSE	$B_e^{(s)}$ [Eq. (45)]
DCESE	$\sqrt{\tilde{n}} B_e^{(s)}$
CURROI (NSOR)	$A_i^{(s)}$ (Amps)
DCURRI (NSOR)	$C_i^{(s)}$ (Amps)
PCURI (NSOR)	$A_i^{(s)}$ [Eq. (37)]
DPCURI (NSOR)	$C_i^{(s)}$ [Eq. (37)]
PCURROSI (NSOR)	$B_i^{(s)}$ [Eq. (37)]
RSI (NSOR)	R_{si} [Eq. (39)]
DCI (NSOR)	D_{ci} [Eq. (39)]
DCISEI (NSOR)	$D_{12} \sqrt{\tilde{n}}$ [Eq. (39)]
CII (NSOR)	$\tilde{S}_i$ [Eq. (35)]
ESORI (NSOR)	Energy at which ion source is centered
XSOI (NSOR)	Velocity spread of ion source
YSOI (NSOR)	Theta spread of ion source
USOI (NSOR)	μ ($\cos \theta$) at which ion source is centered
EOI (NSOR)	Ion source energy
SOI (NSOR)	s_0 [Eq. (39)]
GST(IYAP1,JXA)	Temporary storage array
IYP1	IYA + 1
IY	IYA
IYM1	IYA - 1
IYM2	IYA - 2

Table 2. MAIN program and common block variables. (continued)

Variable	Function
JX	JXA
JXM1	JXA - 1
JXM2	JXA - 2
X(JXA)	Ion velocity mesh (x)
DXP5(JXA)	Forward difference of X
DXM5(JXA)	Backward difference of X
DX(JXA)	Centered difference of X
DXI(JXA)	1/DX (JXA)
Y(IYAP1)	θ mesh
DYP5(IYA)	Forward difference of Y
DYM5(IYA)	Backward difference of Y
DY(IYA)	Centered difference of Y
DYI(IYA)	1./DY (IYA)
CYNT(IYAP1)	Y integration constants [Eq. (81)]
CINT(JXA)	X integration constants [Eq. (81)]
CTNY(IYAP1)	$\cot \theta$
CTNY2(IYAP1)	$\cot^2 \theta$
SPHERI (IYA,JXA)	Two dimensional integration constants
X2I(JXA)	1/X ²
X3I(JXA)	1/X ³
X4I(JXA)	1/X ⁴
XSQ(JXA)	X ²
VEL(JXA)	$\tilde{v}X$
DEG(IYAP1)	θ in degrees
SINN(IYA)	$\sin \theta$
XE (JXE)	Electron velocity mesh (w)
XE2 (JXE)	(XE) ²
DXE (JXE)	Centered difference of XE
DXEM(JXEPI)	Backward difference of XE
FEE(JXE)	F _e (X)
FIE(JXE)	F _i (X) interpolated onto w mesh
ALPE(JXE)	A (Ref. 1, App. B)
BETE(JXE)	B (Ref. 1, App. B)
GAME(JXE)	C (Ref. 1, App. B)

Table 2. MAIN program and common block variables. (continued)

Variable	Function
DELE (JXE)	D (Ref. 1, App. B)
AA(JXE)	α_j (Ref. 1, App. B)
BB(JXE)	β_j (Ref. 1, App. B)
CC(JXE)	γ_j (Ref. 1, App. B)
DD(JXE)	δ_j (Ref. 1, App. B)
EE(JXE)	Used for solving Eq. B4 (Ref. 2, App. B)
FF(JXE)	Used for solving Eq. B4 (Ref. 2, App. B)
FEES(JXE)	Temporary storage array for F_e .
DXEP(JXE)	Forward difference of XE
PLEG ((0,MMX),IYA)	Legendre polynomials- P_ℓ
DPLEG ((0,MMX),IYA)	First derivatives of P_ℓ
DDPLEG ((0,MMX),IYA)	Second derivatives of P_ℓ
UM(IYA)	$\mu = \cos \theta$
DMU(IYA)	Centered μ differences
USIN(IYA)	$\sin \theta$
EMQF(JXA,(0,MX))	M_j [Eq. (28)]
ENQF(JXA,(0,MX))	N_j [Eq. (29)]
ERQF(JXA,(0,MX))	R_j [Eq. (30)]
EEQF(JXA,(0,MX))	E_j [Eq. (31)]
COG ((0,MX),8)	Coefficients in Eq. (32)
GCOX(IYAP1,JXA)	$\frac{\partial G_i}{\partial X} \ell_n \Lambda_{ii}/n_i$
GCOY(IYAP1,JXA)	$\frac{\partial G_i}{\partial \theta} \ell_n \Lambda_{ii}/n_i$
GCOXX(IYAP1,JXA)	$\frac{\partial^2 G_i}{\partial X^2} \ell_n \Lambda_{ii}/n_i$
GCOXY(IYAP1,JXA)	$\frac{\partial^2 G_i}{\partial X \partial \theta} \ell_n \Lambda_{ii}/n_i$
GCOYY(IYAP1,JXA)	$\frac{\partial^2 G_i}{\partial \theta^2} \ell_n \Lambda_{ii}/n_i$
XM(JXA,(MXP3M,MXP4))	Powers of X
MMAX	M [Eq. (26)] see def. of MX.
KNSET	Number of problems
E2	e^2

Table 2. MAIN program and common block variables. (continued)

Variable	Function
E4	e^4
BX	Used for defining initial F_i
CX	Used for defining initial F_i
BY	Used for defining initial F_i
EELECT	Initial electron energy
REIONE	Initial ion energy
KSORC	Chooses $S_2(\theta)$ [Eq. (36)]
GAMO	Γ_{i0}
VC2	x_e^2/Z_i [Eq. (76)]
YFACN	$S_2(\theta)$ for θ -Gaussian
YFACP	$S_2(\theta)$ for μ -Gaussian
YFACNM	$S_2(\theta)$ for normal mode approx.
YFAC	$S_2(\theta)$ [Eq. (36)]

Subroutines

BOUNDARY. Subroutine BOUNDARY computes the ion loss-cone boundary. Variables are listed in Table 3.

Table 3. Variables in subroutine BOUNDARY

Variable	Function
LZ	No. of ψ values
PSI(LZ)	Array of ψ
PPOT(LZ)	$\phi^*(\psi)$
IO(JX)	Corresponds to $\theta(x)$
IMIN	Corresponds to inf. θ
TLC	Candidate for $\bar{\theta}(x)$
JO(IY)	Corresponds to $\bar{x}(\theta)$, where $\bar{x}$ and $\bar{\theta}$ are inverse functions.
JMIN	Corresponds to $x_{\min}$

POTSHAPE. Subroutine POTSHAPE computes $\phi^*(\psi)$ (see *Spatially Dependent Functions*).

DENSINPT. Subroutine DENSINPT computes the quantities $n_{2i}((\psi, \phi^*(\psi)), n_{2e}(\psi, \phi^*(\psi)), p^0$ and p^1 (see *Spatially Dependent Functions*).

GETGF. Subroutine GETGF computes $G_i(x, \theta)$ and its derivatives. The Legendre polynomials and their derivatives along with some constants are computed the first time the subroutine is called, and are stored for the remainder of the problem. Before exiting subroutine COEF is called.

TLOOP. Subroutine TLOOP controls the program after initialization (controlled from MAIN) is completed. Table 4 lists the variables.

Table 4. Variables in subroutine TLOOP.

Variable	Function
NCHEC	Controls how often time dependent data is gathered and stored.
NSET1, ..., 7	Control when Δt is changed
DT1, ..., 7	Control new values of Δt
TINSTOP	Beam turnoff in τ -units.
TCH1,2	Used in deciding when to recompute $G_i(X, \theta)$
REALT	t
DENSITY	$\bar{n}_i$
RDE	n_e
ENERGR	$\bar{E}_i$
REE	$\bar{E}_e$

WHY. Subroutine WHY, which is called during the initialization phase of MAIN, computes the various meshes along with integration constants and other variables directly associated with the mesh values.

OUTPUT. Subroutine OUTPUT controls code output. Printed quantities include $F_i(x, t)$, $F_e(x)$, n_i , n_e , E_i , E_e , ϕ and a perpendicular velocity ion distribution. Plotted output includes time dependent density and energy plots, two two-dimensional views of F_i , a plot of F_e , and two perpendicular velocity distribution plots.

AMBI. Subroutine AMBI controls the advancement of F_i and F_e , and then checks to see if the charge neutrality requirements are being satisfied. (See *Ambipolar Potential*.) Table 5 lists the variables.

Table 5. Variables in subroutine AMBI.

Variable	Function
OLDDEN	$n_e(t)$
OLDE	$E_e(t)$
OLDI	$n_i(t)$
GST(IYAPI, JXA)	$F_i(t)$
FEES(JXE)	$F_e(t)$
MI	Used to index number of attempts to satisfy charge neut. requirements
POLDI	$Z_i n_i(t)$
PGNORM	$Z_i n_i(t + \Delta t)$
DEII	d_i [Eq. (73)]
DEEE	d_e [Eq. (74)]

GERR. Subroutine GERR resets F_i to 0 if F_i becomes negative.

COEF. Subroutine COEF computes the ion contributions to the coefficients of Eq. (1).

GNADE. Subroutine GNADE computes n_i and E_i .

XSWEEP. Subroutine XSWEEP recomputes the ion loss boundary and advances F_i . Table 6 lists the variables.

Table 6. Variables in subroutine XSWEEP.

Variable	Function
RBCN	$2/n_i \Delta \tau$
JMIN	Inf. JO (1)
IMIN	Inf. IO (J)
ALP1	Electron contribution to α_{ij} from a_1^e (Ref. App. B)
BET1	Electron contribution to β_{ij} from a_1^e
ALBE2	Electron contribution to α_{ij} and β_{ij} from b_1^e
GAM1	Electron contribution to γ_{ij} from a_3^e
DEL1	Electron contribution to δ_{ij} from a_3^e
GADE2	Electron contribution to γ_{ij} and δ_{ij} from b_2^e
CC1	$2\pi \gamma_m F_e/n_i Z_i^2$
COMIXI	Implicit a_2^i contribution
COMIXE	Explicit a_2^i contribution
PSI	ψ_{ij} (Ref. App. B)
CHI	χ_{ij} (Ref. App. B)

PREEL. Subroutine PREEL initializes the electron mesh, $F_e(x)$, and sets other electron related parameters. It is called during the initialization phase of MAIN.

GAMMAIN. Subroutine GAMMAIN computes γ_{ab} for $a, b = i, e$ (all combinations). The time-independent contributions are computed just once.

GELECT. Subroutine GELECT computes the electron contributions to the coefficients of Eq. (1).

MOMENTS. Subroutines MOMENTS interpolates F_e onto the ion mesh, and then computes $M(F_e)$, $N(F_e)$, and $E(F_e)$, which are used in subroutine GELECT.

IONISOT. Subroutine IONISOT computes $\tilde{F}_i$, and interpolates it onto the electron mesh.

GETCURE. Subroutine GETCURE adjusts $C_e^{(s)}$ to take into consideration the ionization of neon by cold electrons. The electron current $\tilde{J}_e$ is then computed using Eq. (45).

MAXE. Subroutine MAXE advances F_e . First the coefficients to the normalized Eq. (9) are computed, and the differencing technique discussed earlier is applied. Finally new values of n_e and E_e are computed.

QHOLD2. Subroutine QHOLD2 computes $(\sigma v)_{DT}$ by Eq. (60).

QFUTCH2. Subroutine QFUTCH2 computes $(\sigma v)_{DT}$ by Eq. (71). Subroutines EVALS and GETFG are used.

EVALS. Subroutine EVALS computes the $\hat{s}$ integrand in Eq. (71).

GETFG. Subroutine GETFG computes the $\tilde{F}_1$ and $\tilde{F}_2$ contributions to Eq. (71).

FINDSIG. Subroutine FINDSIG computes the cross-sections defined by Eqs. (62) to (70).

RESTART. Subroutine RESTART controls the temporary terminating and future restarting of the computer code.

OFFMON. Subroutine OFFMON is a local subroutine which terminates the code at a convenient location for future restarting.

INTGRL. Subroutine INTGRL integrates the quantity "F" with respect to "EP", and returns the value in "ANS".

MAXMIN. Subroutine MAXMIN computes the infimum and supremum of a given array "X", and returns the values in "X0" and "X1", respectively.

Appendix A. Modification of Ion Loss Boundary

The ion loss boundary has been recently modified to consider Yushmanov trapping. Equation (77) is replaced by

$$\sin^2 \bar{\theta} = \inf_{1 \leq \psi \leq R} \left\{ \frac{1}{\psi} \left(1 + \frac{\phi^*(\psi)}{x^2} \right) \right\}, \quad (\text{A-1})$$

where the notation is as in *Spatially Dependent Functions*. The quantity $x_{\min}$, defined in Eq. (75), is redefined as

$$x_{\min} = \sup \{ x / \sin^2 \bar{\theta}(x) = 1 \} \quad (\text{A-2})$$

Appendix B. 2X Version of HYBRID-1

INITIALIZATION OF ION DISTRIBUTION

The ion distribution function is given as a function of energy. A mean energy is calculated, and the ions are converted to a velocity distribution function.

TIME CHANGING MAGNETIC FIELD

From the right hand side of Eq. (1) we subtract the quantity

$$\frac{1}{\Gamma_{io}} \frac{\dot{B}}{B} \cdot \left\{ \frac{1}{v^2} \frac{\partial}{\partial v} \left(\frac{v^3 \sin^2 \theta}{2} f_i \right) + \frac{1}{v^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\frac{v^2 \sin^2 \theta \cos \theta}{2} f_i \right) \right\} \quad (B-1)$$

The quantity B is defined as

$$B = \frac{1 + c \sin \alpha}{1 + c \sin \alpha_0} \quad (B-2)$$

$$c = B_{pk}/B_{dc} \quad (B-3)$$

$$\alpha = \frac{\pi}{2} (t + \tau_1 - 45) / B_{up} \quad (B-4)$$

and

$$\alpha_0 = \frac{\pi}{2} (\tau_1 - 45) / B_{up} \quad (B-5)$$

(t is in μ sec)

if $t + \tau_1 \leq B_{up} + 45$, and

$$B = \frac{1 + c \sin \alpha}{1 + c \sin \alpha_0} \quad (B-6)$$

and

$$\alpha = \frac{B_{up} + 45 - t - \tau_1}{B_{up}} \quad (B-7)$$

if $t + \tau_1 > B_{up} + 45$. The quantities τ_1 , B_{pk} , B_{dc} , and B_{up} are input parameters. When the ion equation is differenced, the $\dot{B}/B$ term is split from the rest of the equation, and it is solved by fully implicit splitting.

The electrons are treated by subtracting from the right hand side of Eq. (9) the term

$$\frac{\dot{B}}{B} \frac{1}{v^2} \frac{\partial}{\partial v} \left(\frac{v^3}{3} f_e \right) \quad (B-8)$$

MODIFICATIONS OF SOURCE TERMS

The current associated with each ion-source shape is modified in the following manner. The quantity $A_i^{(s)}$ is given by

$$A_i^{(s)} = n_0^{(s)} \int \sigma_{xy}(v, \theta) f_i(v, \theta) d\vec{v} \quad (B-9)$$

where $n_0^{(s)}$ is the particle density in the beam. The quantity $\sigma_{xy}(v, \theta)$ is defined as

$$\sigma_{xy}(v, \theta) = \frac{2}{\pi} \sqrt{v_0^2 + v^2 + 2vv_0 \sin \theta} \cdot \int_0^{\frac{\pi}{2}} \sigma_x(v_r) \sqrt{1 - \frac{4vv_0 \sin \theta}{v_0^2 + v^2 + 2vv_0 \sin \theta}} \sin^2 \phi d\phi \quad (B-10)$$

where v_0 corresponds to the source energy and

$$\sigma_x(v_r) = \frac{6.937 \times 10^{-15} (1 - 0.673 \ln E)^2}{1 + 1.112 \times 10^{-15} E^{3.3}} \quad (B-11)$$

$$E = \frac{1}{2} (1.67165 \times 10^{-24} |v_r|^2 / 1.602 \times 10^{-12}) \quad (B-12)$$

and

$$v_r = \sqrt{v_0^2 + v^2 + 2vv_0 \sin \theta} \cdot \sqrt{1 - \frac{4vv_0 \sin \theta}{v_0^2 + v^2 + 2vv_0 \sin \theta}} \sin^2 \phi \quad (B-13)$$

In conjunction with the above, the ion distribution is modified at the beginning of the time step as

$$f_i(v, \theta) \rightarrow f_i(v, \theta) \left(1 - n_0^{(s)} \Delta t \sigma_{xy}(v, \theta) \right) \quad (B-14)$$

The quantity $C_i^{(s)}$ is given by

$$C_i^{(s)} = n_0^{(s)} (<\sigma_i v_r> + <\sigma_e v_r>) \quad (B-15)$$

where $<\sigma_i v_r>$ and $<\sigma_e v_r>$ are the ion and electron impact ionization cross sections. The quantity $<\sigma_i v_r>$ is defined as

$$<\sigma_i v_r> = \frac{\int \sigma_{iv}(v, \theta) f_i(v, \theta) d\vec{v}}{\int f_i(v, \theta) d\vec{v}} \quad (B-16)$$

where σ_{iv} is defined as in Eq. (B-10), with σ_i replacing σ_x . The quantity $\sigma_i(v_r)$ is defined as

$$\sigma_i(v_r) = 10^8 \quad (B-17)$$

where

$$a = -0.8712b^2 + 8.156b - 34.833 \quad (B-18)$$

and

$$b = 0.434294 \ln E \quad (B-19)$$

The term $<\sigma_e v_r>$ is defined by

$$<\sigma_e v_r> = 2.152 \times 10^{-8} \cdot \bar{E}_e^{-0.2465} \quad (B-20)$$

The electron source parameter $C_e^{(s)}$ is set to

$$C_e^{(s)} = \sum_{\text{ion sources}} Z_i C_i^{(s)} \quad (B-21)$$

Extra Subroutines and Variables

Subroutine PREION1 reads in the initial ion energy distribution function, and converts it to a velocity distribution function.

Subroutines CHARGE, VREL, SIGX, SIGI, and CSIGVR deal with the ion source modifications discussed above.

Subroutines BCHNG and FIELD compute the time dependent magnetic field contributions.

The loss cone boundary is computed assuming no Yushmanov trapping, so that subroutines BOUNDARY, POTSHAPE, and DENSINPT are not in this version of the code.

Table B-1 lists extra variables.

Table B-1. Extra variables in the 2X version.

Variable	Function
AEP(100)	Array of energies for initial t_i
AGE(100)	Initial ion energy distribution
REBLT0	τ_i
BTAU	B_τ
BPK	B_{pk}
BDC	B_{dc}
BUP	B_{up}
DEN01(NSOR)	$n_0^{(s)}$
SIGVR2(NSOR)	$<\sigma_i v_r>$
SIGVR3	$<\sigma_e v_r>$
SIGXV array	$\sigma_{xv}(v, \theta)$
SIGIV array	$\sigma_{iv}(v, \theta)$
DB1P(IYAPI, JXA)	Corresponds to $v^3 \sin^2 \theta / 2$
DB2P(IYAPI, JXA)	Corresponds to $v^2 \sin^2 \theta \cos \theta / 2$

Appendix C. Listing of HYBRID-1

```

PROGRAM DUMFG(TAPE2,TAPE6,TAPE3=TAPE6,PRINT)
CLICHE COMM
OPTIMIZE
PARAMETER(IYA=24,JXA=45)
PARAMETER(JXE=501)
PARAMETER(NSOR=5)
PARAMETER(MX=4)
PARAMETER(MMX=2*MX,MXP3M=-MMX-3,MXP4=MMX+4)
PARAMETER(JXEP1=JXE+1)
PARAMETER(IYAP1={YA+1})
PARAMETER(IYJXA=IYA+JXA)
PARAMETER(NONCH=200)
PARAMETER(LZ=25,NB=25,NPTS=200)
COMMON/MLZU/PPOT(LZ),ONS1(NB),ONSE(NB),VINT(JXA),VENT(JXE),
C PSI(LZ),Z(LZ),UMMU(IYA),PHI(NB),DDIFF(NB),VINTPE(JXA),
C VINTPA(JXA),COSYN(NPTS),PS1(LZ),PS2(LZ),DNSIE(LZ),PPERP(LZ),
C PPAR(LZ),PSVAC(LZ)
COMMON/MPLGT/PTIME(NONCH),PDENS(NONCH,2),PENGY(NONCH,2)
C ,NCH
COMMON/MCON1/TIME,STOP,DT,AMU,ECON,RHOP
C ,PIQ2,TWOPI,FOURPI,CHARGE,KSET,KWAIT,K,KGCO,PI,XEMAX
C ,TWOPI,XMAX,KGETGFK,DTR
C ,CXRXB(4),CXLOSS,CXLOST,HCXR(JXA)
C ,BETA0
C ,IMIN,JMIN,IMP1,JMP1
C ,NCPOT,NUPT
COMMON/MCOEF/CPB2(IYAP1,JXA),A2P(IYAP1,JXA),ALPHAP(IYAP1,JXA)
C ,BETAP(IYAP1,JXA),GAMMAP(IYAP1,JXA),DELTAP(IYAP1,JXA)
C ,SHAPE(IYAP1,JXA,NSOR)
COMMON/MCON2/DENSE,ENERGE,EMOIM,PMASS,EMASS,EES,EIONS,TNORM
C ,VNORM,DENSNO,GNORM,ENERGY,PNUMB,KMAXE,KGNANDE,KAMBI,NRSTRT
C ,FEMIN,JXEM1,JXEM2,GAM11,GAM1E,GAME1,GAMEE,XSOE
C ,KNPROBS
COMMON/MOIST/FE(JXA),GEX(JXA),GEX(JXA),GEXX(JXA),TEMPG(IYJXA)
C ,EP(IYJXA),FP(IYJXA),G(IYAP1,JXA),GCO(IYAP1,JXA),FEI(JXA)
C ,EMOF(JXA),ENOF(JXA),EEOF(JXA)
COMMON/MCURAMB/CURCON,CURRE,CURR,DCOE,COE,SOURCE(IYAP1,JXA),
C TOTP,POTENT,DENST,IO(JXA),JO(IYAP1),BRATIO,BRATIO1,MINAX,
C TINJSTOP,PCURROE,DPCURROE,PCURROSE,DCESE,CURROI(NSOR),
C DCURRI(NSOR),PCURI(NSOR),DPCURI(NSOR),PCURROSI(NSOR),
C RSI(NSOR),DCI(NSOR),DCISEI(NSOR),CII(NSOR),ESORI(NSOR),
C XSOI(NSOR),YSOI(NSOR),USOI(NSOR),EOI(NSOR),SOI(NSOR)
C ,GST(IYAP1,JXA)
COMMON/MESH/IYPI,IY,IYI,IYI,IYI2,JX,JXMI,JXMI2,X(JXA),DXP5(JXA)
C ,DXM5(JXA),DX(JXA),DXI(JXA),{(IYAP1),DYP5(IYA),DY5(IYA),DY(IYA)
C ,DYI(IYA),CYNT(IYAP1),CINT(JXA),CTNY(IYAP1),CTNY2(IYAP1)
C ,SPHERI(IYA,JXA),X2I(JXA),X3I(JXA),X4I(JXA),XSQ(JXA),VEL(JXA)
C ,DEG(IYAP1),SINN(IYA)
COMMON/MJXE/XE(JXE),XE2(JXE),DXE(JXE),DXEM(JXEP1),FEE(JXE)
C ,FIE(JXE),ALPE(JXE),BETE(JXE),GAME(JXE),DELE(JXE)
C ,AA(JXE),BB(JXE),CC(JXE),DD(JXE),EE(JXE),FF(JXE),FEES(JXE)
DIMENSION DXEP(JXE)
EQUIVALENCE(DXEP(1),DXEM(2))
LCM (MGETG)
COMMON/MGETG/PLEG((0,MMX),IYA),DPLEG((0,MMX),IYA),
C DOPLEG((0,MMX),IYA),UM(IYA),DMU(IYA),USIN(IYA),
C ENQF(JXA,(0,MX)),ENQF(JXA,(0,MX)),ERQF(JXA,(0,MX)),
C EEQF(JXA,(0,MX)),COG((0,MX),B),GCDX(IYAP1,JXA),GCOY(IYAP1,JXA),
C GCOXX(IYAP1,JXA),GCOXY(IYAP1,JXA),GCOYY(IYAP1,JXA),
C XM(JXA,(MXP3M,MXP4))

```

```

C ,MMAX
ENDCLICHE
USE COMM
DIMENSION ND(5) $ ND(1)=0 $ ND(2)=1HF $ ND(3)=6 $ ND(4)=0
ND(5)=1 $ CALL HSP(ND)
CALL ASSIGN(2,0,4RDATA,0)
KNSET=1
DO 1000 KN=1,KNSET
KNPROBS=KN
WRITE (3,1001) KN
1001 FORMAT(1H1,10HTHIS IS RUN NUMBER,15)
KMAXE=0 $ KGNANDE=0 $ KAMBI=0
C TAPE 2 CONTAINS INPUT ON DATA CARDS.
C SET UP VARIOUS CONSTANTS.
IY=IYA $ JX=JXA $ IYM1=IY-1
JXM1=JX-1 $ JXM2=JX-2 $ IYPI=IY+1 $ IYM2=IY-2
JXM1=JXE-1 $ JXM2=JXE-2
PI = 3.1415926536
TWOPI = 2. *PI
FOURPI = 4. *PI
PI02=.5*PI $ TWOPI=2./PI
CHARGE=4.803E-10
E2=CHARGE*CHARGE
E4=E2*E2
ECON=1.602E-9
CURCON=1.E-10*ECON
AMU=1.67165E-24
EMASS=9.1066E-28/AMU
TOTP=0.
TIME=0.
CXLOSS=0. $ CXLOST=0.
KGETGFK=0
DO 215 I=1,IYPI
DO 215 J=1,JX
G(I,J)=0.
215 GCO(I,J)=0.
C READ IN DATA FROM DATA CARDS - THEN OUTPUT IT.
10 FORMAT(10I5)
20 FORMAT(5E16.8)
READ (2,20) XMAX,BX,CX,BY
READ (2,10) KWAIT,KSET,MIMAX
READ (2,20) STOP,DTR,TINJSTOP
READ (2,20) POTENT,DENST
READ (2,20) PMASS,PNUMB
READ (2,20) EES,DENSNO,EELECT
READ (2,20) REIONE
READ (2,20) PCURROE,DPCURROE,PCURROSE
READ (2,20) BRATIO
READ (2,20) (CXRXB(K),K=1,4)
READ (2,10) NRSTRT
READ (2,20) FEMIN,XSOE
DO 511 IC=1,NSOR
READ (2,20) ESOR1(IC),XSOR1(IC),YSOR1(IC),USOR1(IC)
READ (2,20) PCUR1(IC),DPCUR1(IC),PCURROS1(IC)
511 CONTINUE
READ (2,10) MMAX,KSORC
READ (2,20) BETA0
READ (2,10) NCPOT
WRITE (3,129) NCPOT
129 FORMAT(6HNCPOT=,15)

```

```

WRITE (3,219) MMAX,KSORC
219 FORMAT(5HMMAX=,I5,5X,6HKSORC=,I5)
C KSORC=0 FOR MU GAUSSIAN,1 FOR N MODE,2 FOR THETA GAUSSIAN
WRITE (3,33) XMAX,BX,CX,BY
33 FORMAT(5HXMAX=,E16.6,5X,3HXB=,E16.6,5X,3HCX=,E16.6,5X,3HBY=,E16.6)
WRITE (3,301) KWAIT,KSET,MIMAX,POTENT,DENST
301 FORMAT(6HKWAIT=,I5,5X,5HKSET=,I5,5X,6HMMAX=,I5,5X,10HAMB POT =,
C E16.6,5X,6HDENST=,E16.6)
WRITE (3,302) STOP,DTR,PMASS,BRATIO
302 FORMAT(12HTIME LIMIT =,E16.6,2X,7HSECONDS,5X,4HDTR=,E16.6,5X,
C 10HION MASS =,F10.3,5H AMU,5X,14HMIRROR RATIO =,F10.3)
WRITE (3,308) PNUMB
308 FORMAT(19HION ATOMIC NUMBER =,E16.6)
WRITE (3,303) EES,REIONE,DENSNO,EELECT
303 FORMAT(16HELEC SOURCE EN =,E16.6,5H KEV,5X,7HREIONE=,E16.6,
C 5H KEV,5X,7HDENSNO=,E16.6,7HPART/CC,5X,9HELEC EN =,E16.6,
C 5H KEV)
WRITE (3,351) TINJSTOP
351 FORMAT(9HTINJSTOP=,E16.6,8H SECONDS)
WRITE (3,357) (CXRXB(K),K=1,4)
357 FORMAT(6HCXRXB=,4E16.6)
WRITE (3,501) FEMIN,XSOE
501 FORMAT(6HFEMIN=,E16.6,5X,5HXSOE=,E16.6)
DO 512 IC=1,NSOR
WRITE (3,411) IC,ESORI(IC),XSOI(IC),YSOI(IC),USOI(IC),
C PCURI(IC),DPCURI(IC),PCURROSI(IC)
411 FORMAT(3HIC=,I5,5X,5HESOR=,E16.6,5X,4HXSO=,E16.6,5X,4HYSO=,
C E16.6,5X,4HUSO=,E16.6,/,5HPCUR=,E16.6,5X,6HPCUR=,E16.6,5X,
C 8HPCURROSI=,E16.6)
512 CONTINUE
CALL WHY
C SUBROUTINE WHY SETS UP MESH AND INTEGRATION CONSTANTS
DO 34 J=1,JX
G(IY,J)=EXP(-BX*(X(J)-CX)**2)
34 CONTINUE
FACTORA=1./COSF(Y(I))**2
DO 131 I=1,IYI
FACTORB=1.-FACTORA*COSF(Y(I))**2
IF (BY .LT. 0.) FACTORB=EXP(BY*COSF(Y(I))**2)
DO 132 J=1,JX
G(I,J)=G(IY,J)*FACTORB
132 CONTINUE
131 CONTINUE
DO 133 J=1,JX
133 G(IYI,J)=G(IYI,J)
CALL GNANDE
C SUBROUTINE GNANDE COMPUTES ION DENSITY AND ION ENERGY (NORMALIZED).
C RENORMALIZE ION DISTRIBUTION.
IF (KGNANDE .EQ. 1) GO TO 1002
DO 1400 I = 1, IYI
DO 1400 J = 1, JX
1400 G(I,J)=G(I,J)/GNORM
GNORM=1.
EIONS=REIONE/ENERGY
EMOIM = EMASS/PMASS
VNORM = SQRTF (2.*EIONS*ECON/(PMASS*AMU))
GAMO=FOURPI*E4*PNUMB**4/(PMASS*AMU)**2
DO 401 J = 1, JX
401 VEL(J) = X(J)*VNORM
WRITE (3,32)

```

```

32 FORMAT(/,3X,16HJ      NORMED VEL,8X,8HVELOCITY,/)
WRITE (3,31) (J,X(J),VEL(J),J=1,JX)
31 FORMAT(15,F10.5,E18.8)
C COMPUTE SHAPE FUNCTION FOR ION SOURCE.
DO 773 IC=1,NSOR
  XSOR=SQRTF(ESOR1(IC)/EIONS)
  XSO=XSO1(IC)
  DO 320 J=1,JX
    ARGX=XSO*(X(J)-XSOR)**2
    SHAPE(IY,J,IC)=EXP(-ARGX)
  DO 320 I=1,IY
    YSO=YSO1(IC) $ USO=USO1(IC)
    YFACN=EXP(-YSO*(Y(I)-PI02)**2)
    YFACP=EXP(-YSO*(COS(Y(I))-USO)**2)
    REFF=BRAT(0/(1.+PNUMB*POTENT/EIONS)
    UMR2=1.-1./REFF
    UM2=COS(Y(I))**2
    ENUM=UMR2-UM2*(3.*UMR2-1.)*LOGF((1.-UM2)/(1.-UMR2))
    DENOM=UMR2-(3.*UMR2-1.)*LOGF(1.-UMR2)
    YFACNM=ENUM/DENOM
    YFAC=YFACN
    YFAC=YFACNM
    YFAC=YFACP
    IF (KSORC.EQ. 1) YFAC=YFACNM
    IF (KSORC.EQ. 2) YFAC=YFACN
    SHAPE(I,J,IC)=SHAPE(IY,J,IC)*YFAC
    TEMPG(I)=YFAC
320 CONTINUE
  WRITE (3,323)
323 FORMAT(/,5HPOINT,5X,6HCOSINE,9X,12HSOURCE SHAPE)
  DO 321 I=1,IY
    S=COS(Y(I)) $ T=TEMPG(I)
    WRITE (3,322) I,S,T
322 FORMAT(15,5X,F10.3,5X,E16.6)
321 CONTINUE
C COMPUTE ENERGY OF ION SOURCE.
EO = 0.
CI = 0.
DO 653 I = 1, IY
  DO 653 J = 1, JX
    EO=EO+SHAPE(I,J,IC)*XSQ(J)*SPHERI(I,J)
653 CI=CI+SHAPE(I,J,IC)*SPHERI(I,J)
  EO = EO*EIONS/CI
  EO1(IC)=EO $ CII(IC)=CI
773 CONTINUE
  TNORM = VNORM**3/(DENSNO*GAMO)
  DT=DTR/TNORM
  RHOP = 2./DT
  VC2=POTENT/EIONS
  VC=SQRTF(VC2)
  DO 774 IC=1,NSOR
    CURRO1(IC)=PCURI(IC)*CURCON $ DCURRI(IC)=DPCURI(IC)*CURCON
    RSI(IC)=(PCURI(IC)*TNORM)/(CII(IC)*DENSNO)
    DCI(IC)=DCURRI(IC)*DENSNO/CURRO1(IC)
    DCISE1(IC)=TNORM*PCURROSI(IC)/(CII(IC)*SQRTF(DENSNO))
774 CONTINUE
  ENERGE=EELECT/EIONS
  DENSE=GNORM*PNUMB
  CALL PREEL
C SUBROUTINE PREEL SETS UP CONSTANTS INVOLVING ELECTRON EQUATION.

```

```

CALL POTSHAPE
CALL BOUNDARY
GNST=GNORM $ CALL GNADE $ IF (KGNANDE .EQ. 1) GO TO 1002
FACT=GNST/GNORM $ DO 76 I=1,IYPI $ DO 76 J=1,JX
76 G(I,J)=G(I,J)*FACT
GNORM=GNST
CALL GAMMAIN(0)
DO 775 I=1,IY $ DO 775 J=1,JX
775 SOURCE(I,J)=0.
CURR=0.
DO 776 IC=1,NSOR
SOI(IC)=RSI(IC)*(1.+DCI(IC)*GNORM)+DCISEI(IC)*SQRTF(GNORM)
DO 113 I=1,IY
DO 113 J=1,JX
113 SOURCE(I,J)=SOURCE(I,J)+SOI(IC)*SHAPE(I,J,IC)
CURR=CURR+SOI(IC)*PCURI(IC)/RSI(IC)
776 CONTINUE
WRITE (3,324)
324 FORMAT(//,6HSOURCE,/)
DO 325 J=1,JX
WRITE (3,326) (SOURCE(I,J),I=1,IY)
326 FORMAT(6E16.6)
325 CONTINUE
DO 777 IC=1,NSOR
WRITE (3,305) IC,PCURI(IC),DPCURI(IC),PCURROSI(IC),RSI(IC),
C DCI(IC),DCISEI(IC),SOI(IC),EOI(IC)
305 FORMAT(3HIC=,I5,5X,5HPCUR=,E16.6,5X,6HDPCUR=,E16.6,5X,8HPCURROS=,
C E16.6,/,3HRS=,E16.6,5X,3HDC=,E16.6,5X,6HDCISE=,E16.6,/,3HSD=,
C E16.6,5X,3HEO=,E16.6)
777 CONTINUE
WRITE (3,306) VC,ENERGY,GNORM,VNORM,TNORM
306 FORMAT(3HVC=,E16.6,5X,17HNORM ION ENERGY =,E16.6,5X,
C 18HNORM ION DENSITY =,E16.6,/,6HVNORM=,E16.6,5X,6HTNORM=,E16.6)
WRITE (3,307) PCURROE,DPCURROE,PCURROSE
307 FORMAT(18HELEC PART SOURCE =,3E16.6)
CALL CRTID(10HAM 2D IONS,1,1)
CALL FRAME
CALL OUTPUT
K = 0
KGCO=0
CALL TLOOP
C SUBROUTINE TLOOP TAKES CARE OF THE REGULAR CYCLING OF THE PROGRAM.
CALL OUTPUT
C SUBROUTINE OUTPUT PERFORMS STANDARD OUTPUT.
IF (KGNANDE .EQ. 1 .OR. KMAXE .EQ. 1 .OR. KAMBI .EQ. 1) GO TO 1002
CALL QHOLD2
CALL QFUTCH2
CALL DENSINPT
1002 CONTINUE
1000 CONTINUE
CALL QUIT(1)
END
SUBROUTINE BOUNDARY
USE COMM
DO 1 J=2,JX
IO(J)=IY
DO 2 L=1,LZ
XC2=PNUMB*PPOT(L)/EIONS
SIN2T=(1.+XC2/XSQ(J))/PSI(L)
IST=IO(J)

```

```

IF (SIN2T .GE. 1.-1.E-13) GO TO 2
TLC=ASINF(SQRTF(SIN2T))
10(J)=1 $ IF (TLC .LE. Y(1)+.5*DYP5(1)) GO TO 3
10(J)=1Y
DO 4 I=2,IYM1
IF (TLC .LE. Y(1)+.5*DYP5(1)) GO TO 5
4 CONTINUE
I=1Y
5 10(J)=1
IF (I .GT. 1ST) 10(J)=1ST
2 CONTINUE
3 CONTINUE
1 CONTINUE
10(1)=1Y
DO 6 J=1,JX
IF (10(J) .LT. 1Y) GO TO 7
6 CONTINUE
WRITE (3,8) ((L,PSI(L),PPOT(L)),L=1,LZ)
8 FORMAT(/,16HBOUNDARY FAILURE,/, (2HL=,15,5X,4HP51=.E16.6,5X,4HPOT=,
C E16.6))
DO 9 I=1,IYP1 $ DO 9 J=1,JX
9 G(I,J)=0.
RETURN
7 CONTINUE
JMIN=J-1
JO(IY)=JMIN
IMIN=10(JX)
JO(IMIN)=JX
IMP1=IMIN+1 $ JMP1=JMIN+1
DO 10 I=IMP1,IYM1
JO(I)=0 $ DO 12 J=JMP1,JXM1
IF (10(J) .EQ. 1) JO(I)='
12 CONTINUE
10 CONTINUE
DO 13 I=1,IMIN-1
13 JO(I)=JX
DO 14 I=IMP1,IYM1
I=IMP1+IYM1-I
IF (JO(I) .EQ. 0) JO(I)=JO(I+1)
14 CONTINUE
DO 15 I=1,IY
DO 15 J=1,JO(I)
15 G(I,J)=0.
IF (TIME .GT. 0.) RETURN
WRITE (3,107) JMIN,((1,JO(1)),I=1,IY)
107 FORMAT(/,5HJMIN=,15,/, (2H1=,15,2X,3HJO=,15))
WRITE (3,108) IMIN,((J,10(J)),J=1,JX)
108 FORMAT(/,5HIMIN=,15,/, (2HJ=,15,2X,3H10=,15))
RETURN
END
SUBROUTINE POTSHAPE
USE COMM
IF (TIME .GT. 0.) GO TO 96
NBMI=NB-1 $ EMZ=EMOIM*PNUMB
BETN=PI/FLOATF(NPTS) $ BETN2=.5*BETN
LZM1=LZ-1
DO 97 IN=1,NPTS
ARGN=FLOATF(2*IN-1)*BETN2
COSYN(IN)=COSF(ARGN)
97 CONTINUE

```

```

DO 1 I=2,IY
1 UMMU(I)=COSF(Y(IYI-I))
  UMMU(I)=0.
95 CONTINUE
  PPOT(I)=0. $ PPOT(LZ)=POTENT
  VC2=PNUMB*POTENT/EIONS
  CUT=SQRTF(VC2/EMZ)
  PHI(I)=0. $ HP=VC2/FLOATF(NBM1)
  DO 2 K=2,NB
2 PHI(K)=PHI(K-1)+HP
  KLIM=1
  S=0.
  XCN2=POTENT/(EMOIM*EIONS)
  XCN=SQRTF(XCN2)
  DO 32 J=1,JXE
  FACT=1. $ IF (XE(J) .LE. XCN) GO TO 322
  FACT=SQRTF(1.-BRATIO*(1.-XCN2/XE2(J)))
322 S=S+FACT*FEE(J)*XE2(J)*DXE(J)
321 CONTINUE
  DENSEU=FOURPI*S
  DO 73 L=2,LZM1
  DO 3 K=KLIM,NB
  DNSI(K)=0. $ DNSE(K)=0.
  PH=PHI(K) $ PS=PSI(L)
  DO 4 J=1,JX
  VINT(J)=0. $ IF (J .EQ. 1) GO TO 4
  S=0. $ DO 5 IN=1,NPTS
  SJN=COSYN(IN)
  RATSW=X(J)/SQRTF(XSQ(J)+PH)
  ULLIM=RATSW*PSI(L) $ ULLI=RATSW*PS2(L)
  IF (SJN .GT. ULLIM .OR. SJN .LT. ULLIM) GO TO 5
  WJN=SJN/RATSW $ UMU2=1.-WJN**2/PS
  UMU=SQRTF(UMU2+1.E-13) $ IO 7 II=1,IYMI
  IF (UMU .LT. UMMU(II) .OR. JMU .GE. UMMU(II+1)) GO TO 7
  IUSE=II $ GO TO 8
7 CONTINUE
  IUSE=IYMI
8 WI=(UMU-UMMU(IUSE))/(UMMU(IUSE+1)-UMMU(IUSE))
  W0=1.-WI
  IVSE=IYPI-IUSE
  FIONS=W0*G(IVSE,J)+WI*G(IVSE-1,J)
  S=S+FIONS*SJN
5 CONTINUE
  VINT(J)=BETN*S
4 CONTINUE
  S=0. $ DO 10 J=1,JX
  T=X(J)*SQRTF(XSQ(J)+PH)
10 S=S+T*VINT(J)*CINT(J)
  DNSI(K)=S
  DO 21 JJ=1,JXEM1
  IF (CUT .LT. XE(JJ) .OR. CUT .GE. XE(JJ+1)) GO TO 21
  JP=JJ $ GO TO 22
21 CONTINUE
  CUT=XEMAX $ JP=JXE
22 JPI=JP+1
  CUTL=SQRTF(PH/EMZ)
  DO 23 JJ=1,JXEM1
  IF (CUTL .LT. XE(JJ) .OR. CUTL .GE. XE(JJ+1)) GO TO 23
  JQ=JJ $ GO TO 24
23 CONTINUE

```



```

      CUTL=XEMAX $ JQ=JXE
24  JQ1=JQ+1
      DO 27 J=1,JQ
27  VENT(J)=0.
      DO 25 J=JQ1,JP
25  VENT(J)=SQRTF(1.-PH/(EMZ*XE2(J)))
      DO 26 J=JP1,JXE
26  VENT(J)=SQRTF(1.-PS/BRATIO+(PS/BRATIO*VC2-PH)/(XE2(J)*EMZ))
      S=0. $ DO 13 J=1,JXE
      T=XE2(J)*FEE(J)*VENT(J)
13  S=S+T*DXE(J)
      DNSE(K)=S*GNORM/DENSEU
      3  CONTINUE
      IF (TIME .GT. 0.) GO TO 173
      WRITE (3,174) L,((K,DNSI(K),DNSE(K)),K=KLIM,NB)
174  FORMAT(/,2HL=,15,/,12HK=,15,2X,5HDNSI=,E16.6,2X,5HDNSE=,E16.6)
173  CONTINUE
      DO 31 K=KLIM,NB
31  DDIFF(K)=DNSI(K)-DNSE(K)
      DO 33 K=KLIM,NB
      PROD=DDIFF(K)*DDIFF(K+1)
      IF (PROD .LE. 0.) GO TO 34
33  CONTINUE
      S=ABSF(DDIFF(KLIM)) $ KHU=KLIM
      DO 133 K=KLIM+1,NB
      T=ABSF(DDIFF(K)) $ IF (: .LE. T) GO TO 133
      S=T $ KHU=K
133  CONTINUE
      PPOT(L)=PHI(KHU)*EIONS/F.UMB
      KLIM=KHU
      GO TO 73
34  CONTINUE
      DENOM=DDIFF(K)-DDIFF(K+1)
      ENUM=PHI(K+1)-PHI(K)
      IF (ABSF(DENOM) .LT. 1.E-90) DENOM=1.E+90
      PPOT(L)=(PHI(K)+DDIFF(K)*ENUM/DENOM)*EIONS/PNUMB
      KLIM=K
73  CONTINUE
      IF (TIME .GT. 0.) RETURN
      WRITE (3,175) ((L,PSI(L),PPOT(L)),L=1,LZ)
175  FORMAT(2HL=,15,2X,4HPSI=,E16.6,2X,5HPPOT=,E16.6)
      RETURN
      END
      SUBROUTINE DENSINPT
      USE COMM
      EMZ=EMOIM*PNUMB
      BETN=PI/FLOAT(NPTS) $ BETN2=.5*BETN
      LZM1=LZ-1
      VC2=PNUMB*POTEN1/EIONS
      CUT=SQRTF(VC2/EMZ)
      DO 73 L=1,LZM1
      PH=PPOT(L)*PNUMB/EIONS
      PS=PSI(L)
      DO 4 J=1,JX
      VINT(J)=0. $ VINTPE(J)=0. $ VINTPA(J)=0.
      IF (J .EQ. 1) GO TO 4
      S=0. $ SE=0. $ SA=0.
      DO 5 IN=1,NPTS
      SJN=COSYN(IN) $ RATSW=X(J)/SQRTF(XSQ(J)+PH)
      UULIM=RATSW*PSI(L) $ ULLIM=RATSW*PS2(L)

```

```

IF (SJN .GT. UULIM .OR. SJN .LT. ULLIM) GO TO 5
WJN=SJN/RATSW $ UMU2=1.-WJN**2/PS
UMU=SQRTF(UMU2+1.E-13) $ DO 7 I1=1,IYM1
IF (UMU .LT. UMMU(I1) .OR. UMU .GE. UMMU(I1+1)) GO TO 7
IUSE=I1 $ GO TO 8
7 CONTINUE
IUSE=IYM1
8 W1=(UMU-UMMU(IUSE))/(UMMU(IUSE+1)-UMMU(IUSE))
W0=1.-W1
IVSE=IYPI-IUSE
FIONS=W0*G(IVSE,J)+W1*G(IVSE-1,J)
T=FIONS*SJN
TE=T*EIONS*ECON*PS*XSQ(J)*(1.-UMU**2)
TA=2.*T*EIONS*ECON*(XSQ(J)*(1.-PS*(1.-UMU**2))+PH)
S=S+T $ SE=SE+TE $ SA=SA+TA
5 CONTINUE
VINT(J)=BETN*S $ VINTPE(J)=BETN*SE $ VINTPA(J)=BETN*SA
4 CONTINUE
S=0. $ SE=0. $ SA=0.
DO 10 J=1,JX
T0=X(J)*SQRTF(XSQ(J)+PH)
T=T0*VINT(J) $ TE=T0*VINTPE(J) $ TA=T0*VINTPA(J)
S=S+T*CINT(J) $ SE=SE+TE*CINT(J) $ SA=SA+TA*CINT(J)
10 CONTINUE
DNSIE(L)=S*FOURPI*DENSNO $ PPERP(L)=SE*FOURPI*DENSNO
PPAR(L)=SA*FOURPI*DENSNO
DO 21 JJ=1,JXEM1
IF (CUT .LT. XE(JJ) .OR. CUT .GE. XE(JJ+1)) GO TO 21
JP=JJ $ GO TO 22
21 CONTINUE
CUT=XEMAX $ JP=JXE
22 JPI=JP+1
CUTL=SQRTF(PH/EMZ) $ DO 23 JJ=1,JXEM1
IF (CUTL .LT. XE(JJ) .OR. CUTL .GE. XE(JJ+1)) GO TO 23
JQ=JJ $ GO TO 24
23 CONTINUE
CUTL=XEMAX $ JQ=JXE
24 JQ1=JQ+1
DO 27 J=1,JQ
27 VENT(J)=0.
DO 25 J=JQ1,JP
VENT(J)=SQRTF(1.-PH/(EMZ*XE2(J)))
FAC1=EMASS*AMU*VNORM**2/3.*(XE2(J)-PH/EMZ)
VENT(J)=FAC1*VENT(J)
25 CONTINUE
DO 26 J=JPI,JXE
26 VENT(J)=0.
S=0. $ DO 13 J=1,JXE
T=XE2(J)*FEE(J)*VENT(J)
13 S=S+T*DXE(J)
PPERP(L)=PPERP(L)+S*FOURPI*DENSNO
PPAR(L)=PPAR(L)+S*FOURPI*DENSNO
73 CONTINUE
DNSIE(LZ)=0. $ PPERP(LZ)=0. $ PPAR(LZ)=0.
WRITE (3,41) ((L,PSI(L),PPOT(L),DNSIE(L),PPERP(L),PPAR(L)),L=1,LZ)
41 FORMAT(/,(2HL=,15,2X,4HPSI=,E16.6,2X,4HPOT=,E16.6,2X,5HDENS=,
C E16.6,/,10X,6HPPERP=,E16.6,2X,5HPPAR=,E16.6))
REALT=TNORM*TIME
CALL FRAME
CALL CARTMM(LZ,DMIN,DMAX,PPOT,1)

```

```

CALL CARTMM(LZ,EMIN,EMAX,DNSIE,1)
CALL MAPS(1.,BRATIO,DMIN,DMAX,.2,.999,.15,.5)
CALL TRACE(PSI,PPOT,LZ)
CALL MAPS(1.,BRATIO,EMIN,EMAX,.2,.999,.65,.999)
CALL TRACE(PSI,DNSIE,LZ)
CALL SETCH(17.,1.,1,0,1,0)
WRITE (100,81) REALT
81 FORMAT(5HTIME=,E16.6,9H SECONDS)
CALL SETCH(1.,17.,1,0,1,1)
WRITE (100,82)
82 FORMAT(14HN OF PSI - TOP)
CALL SETCH(2.,17.,1,0,1,1)
WRITE (100,83)
83 FORMAT(19HPHI OF PSI - BOTTOM)
CALL FRAME
CALL CARTMM(LZ,DMIN,DMAX,PPERP,1)
CALL CARTMM(LZ,EMIN,EMAX,PPAR,1)
CALL MAPS(1.,BRATIO,DMIN,DMAX,.2,.999,.15,.5)
CALL TRACE(PSI,PPERP,LZ)
CALL MAPS(1.,BRATIO,EMIN,EMAX,.2,.999,.65,.999)
CALL TRACE(PSI,PPAR,LZ)
CALL SETCH(17.,1.,1,0,1,0)
WRITE (100,81) REALT
CALL SETCH(1.,17.,1,0,1,1)
WRITE (100,84)
84 FORMAT(17HPPAR OF PSI - TOP)
CALL SETCH(2.,17.,1,0,1,1)
WRITE (100,85)
85 FORMAT(21HPPERP OF PSI - BOTTOM)
CALL FRAME
EKM=1.E+90 $ DO 201 L=1,LZM1
DENOM=PPERP(L)-PPERP(L+)
IF (DENOM .LE. 0.) GO TO 201
EK=(PSI(L+1)**2-PSI(L)**2)/DENOM
IF (EKM .GT. EK) EKM=EK
201 CONTINUE
ELM=1.E+90 $ DO 209 L=1,LZM1
DENOM=PPAR(L+1)-PPAR(L) $ IF (DENOM .LE. 0.) GO TO 209
EK=(PSI(L+1)**2-PSI(L)**2)/DENOM
IF (ELM .GT. EK) ELM=EK
209 CONTINUE
BETMAX=EKM*PPERP(1)/(1.+EKM*PPERP(1))
BETO=BETA0 $ IF (BETO .GT. BETMAX) BETO=BETMAX
DO 202 L=1,LZ
202 PSVAC(L)=SQRTF(PSI(L)**2+BETO*PPERP(L)/(PPERP(1)*(1.-BETO)))
RVAC=BRATIO*SQRTF(1.-BETO)
EKTP=PPERP(1)/DNSIE(1) $ EKTPK=EKTP/ECON
CALL CARTMM(LZ,DMIN,DMAX,PSVAC,1)
CALL MAPS(1.,BRATIO,DMIN,DMAX,.2,.999,.5,.999)
CALL TRACE(PSI,PSVAC,LZ)
CALL SETCH(1.,17.,1,0,1,1)
WRITE (100,203)
203 FORMAT(13HPSIVAC OF PSI)
CALL SETCH(17.,7.,1,0,1,0)
WRITE (100,204) EKM,ELM,RVAC,BETMAX,BETA0,EKTP,EKTPK
204 FORMAT(4HEKM=,E16.6,2X,4HELM=,E16.6,/,5HRVAC=,E16.6,/,7HBETMAX=,
C E16.6,2X,6HBETA0=,E16.6,/,7HKTPERP=,E16.6,9H ERGS = ,E16.6,
C 5H KEV)
CALL SETCH(17.,1.,1,0,1,0)
WRITE (100,81) REALT

```

```

CALL FRAME
WRITE (3,205) EKM,ELM,RVAC,BETMAX,BETA0,EKTP,EKTPK.
C ((L,PSI(L),PSVAC(L)),L=1,LZ)
205 FORMAT(/,4HEKM=,E16.6,2X,4HELM=,E16.6,2X,5HRVAC=,E16.6,2X,
C 7HSETMAX=,E16.6,/,6HBETA0=,E16.6,5X,7HKTPERP=,E16.6,9H ERGS = ,
C E16.6,5H KEV,/,7HPSIVAC=,/, (15,2E16.6))
RETURN
END
SUBROUTINE GETGF
USE COMM
IF (KGETGFK .EQ. 1) GO TO 10
DO 20 M=MXP3M,MXP4
DO 1 J=2,JX
1 XM(J,M)=X(J)**M
20 XM(1,M)=0.
DO 3 I=1,IY
3 UM(I)=COSF(Y(I))
DO 4 I=1,IY
PLEG(0,I)=1. $ DPLEG(0,I)=0.
IF (MX .EQ. 0) GO TO 4
PLEG(1,I)=UM(I) $ DPLEG(1,I)=1.
DO 5 M=2,MMX
PLEG(M,I)=(2*M-1)*UM(I)*PLEG(M-1,I)-(M-1)*PLEG(M-2,I)/FLOATF(M)
DPLEG(M,I)=(2*M-1)*PLEG(M-1,I)+DPLEG(M-2,I)
5 CONTINUE
4 CONTINUE
DO 6 I=1,IY
DO 6 M=0,MMX
DDPLEG(M,I)=(2.*UM(I)*DPLEG(M,I)-FLOATF(M*(M+1))*PLEG(M,I))/
C (1.-UM(I)**2)
6 CONTINUE
DO 7 MM=0,MX
M=2*MM
FAC=FOURPI/(2*M+1)
COG(MM,1)=FAC/FLOATF(2*M+3)
COG(MM,2)=FAC/FLOATF(2*M-1)
COG(MM,3)=FLOATF(M+2)*COG(MM,1)
COG(MM,4)=FLOATF(M+1)*COG(MM,1)
COG(MM,5)=FLOATF(M)*COG(MM,2)
COG(MM,6)=FLOATF(M-1)*COG(MM,2)
COG(MM,7)=FLOATF(M+1)*COG(MM,3)
COG(MM,8)=FLOATF(M-1)*COG(MM,5)
7 CONTINUE
DO 9 I=2,IYMI
9 DMU(I)=.5*(UM(I+1)-UM(I-1))
DMU(1)=.5*(UM(2)-UM(1)) $ DMU(IY)=.5*(UM(IY)-UM(IYMI))
DO 10 I=1,IY
10 USIN(I)=SINF(Y(I))
KGETGFK=1
10 CONTINUE
DO 14 MM=0,MX
M=2*MM
DO 12 J=1,JX
S=0. $ DO 11 I=1,IY
11 S=S+G(I,J)*PLEG(M,I)*DMU(I)
12 TEMPG(J)=-FLOATF(2*M+1)*S
IX1=1-M $ IX2=2-M $ IX3=3-M $ IX4=4-M
IX5=3-IX1 $ IX6=3-IX2 $ IX7=3-IX3 $ IX8=3-IX4
SM=0. $ SN=0. $ SR=0. $ SE=0.
EMQF(JX,MM)=SM $ ENQF(1,MM)=SN $ ERQF(JX,MM)=SR $ EEQF(1,MM)=SE

```

```

DO 13 J=2,JX
JJ=JX+1-J
TM=.5*DXP5(JJ)*(XM(JJ,1X1)*TEMPG(JJ)+XM(JJ+1,1X1)*TEMPG(JJ+1))
TN=.5*DXM5(JJ)*(XM(J-1,1X2)*TEMPG(J-1)+XM(J,1X2)*TEMPG(J))
TR=.5*DXP5(JJ)*(XM(JJ,1X3)*TEMPG(JJ)+XM(JJ+1,1X3)*TEMPG(JJ+1))
TE=.5*DXM5(JJ)*(XM(J-1,1X4)*TEMPG(J-1)+XM(J,1X4)*TEMPG(J))
SM=SM+TM $ SN=SN+TN $ SR=SR+TR $ SE=SE+TE
EMQF(JJ,MM)=SM $ ENQF(J,MM)=SN $ ERQF(JJ,MM)=SR $ EEQF(J,MM)=SE
13 CONTINUE
DO 15 J=1,JX
EMQF(J,MM)=EMQF(J,MM)*XM(J,1X5)
ENQF(J,MM)=ENQF(J,MM)*XM(J,1X6)
ERQF(J,MM)=ERQF(J,MM)*XM(J,1X7)
EEQF(J,MM)=EEQF(J,MM)*XM(J,1X8)
15 CONTINUE
14 CONTINUE
DO 16 J=1,JX $ DO 16 I=1,IY
S=0. $ SX=0. $ SY=0. $ SXX=0. $ SXY=0. $ SYY=0.
DO 17 M=0,MX
IF (M.GT. MMAX) GO TO 17
MM=2*M
T=COG(M,1)*(EEQF(J,M)+EMQF(J,M))-COG(M,2)*(ENQF(J,M)+ERQF(J,M))
TX=X21(J)*X(J)*(COG(M,3)*EMQF(J,M)-COG(M,4)*EEQF(J,M)-
C COG(M,5)*ERQF(J,M)+COG(M,6)*ENQF(J,M))
TX=X21(J)*(COG(M,7)*(ENQF(J,M)+EMQF(J,M))
C COG(M,8)*(ENQF(J,M)+ERQF(J,M)))
S=S+T*PLEG(MM,1) $ SX=SX+TX*PLEG(MM,1)
SXX=SXX+TX*TX*PLEG(MM,1) $ SY=SY+T*DPLEG(MM,1)
SXY=SXY+TX*DPLEG(MM,1) $ SYY=SYY+T*DDPLEG(MM,1)
17 CONTINUE
GCO(I,J)=S
GCOX(I,J)=SX $ GCOY(I,J)=-USIN(I)*SY
GCOXX(I,J)=SXX $ GCOXY(I,J)=-USIN(I)*SXY
GCOYY(I,J)=-UM(I)*SY+USIN(I)**2*SYY
16 CONTINUE
DO 19 J=1,JX
GCO(IYP1,J)=GCO(IYM1,J)
GCOX(IYP1,J)=GCOX(IYM1,J) $ GCOY(IYP1,J)=-GCOY(IYM1,J)
GCOXX(IYP1,J)=GCOXX(IYM1,J) $ GCOXY(IYP1,J)=-GCOXY(IYM1,J)
GCOYY(IYP1,J)=GCOYY(IYM1,J)
19 CONTINUE
WRITE (3,2)
2 FORMAT(14HGCO RECOMPUTED)
GN=GAM11/GNORM
DO 21 I=1,IYP1 $ DO 21 J=1,JX
GCO(I,J)=GN*GCO(I,J)
GCOX(I,J)=GN*GCOX(I,J) $ GCOY(I,J)=GN*GCOY(I,J)
GCOXX(I,J)=GN*GCOXX(I,J) $ GCOXY(I,J)=GN*GCOXY(I,J)
GCOYY(I,J)=GN*GCOYY(I,J)
21 CONTINUE
KGCO=0 $ CALL COEF
RETURN
END
SUBROUTINE TLOOP
USE COMM
TINSTOP=TINJSTOP/TNORM
N=0
NCH=0 $ NCHC=1
READ (2,11) NSET1,NSET2,NSET3,NSET4,NSET5,NSET6,NSET7
11 FORMAT(7I5)

```

```

      READ (2,12) DT1,DT2,DT3,DT4,DT5,DT6,DT7
      12 FORMAT(4E16.6,/,3E16.6)
204  CONTINUE
      N=N+1
      IF (N.EQ. NRSTRT) GO TO 998
      IF (SENSE SWITCH 1) 998,999
998  CALL RESTART
999  CONTINUE
C USE THIS AREA TO ALTER DT IN FUTURE
      IF (N.EQ. NSET1) DTR=DT1
      IF (N.EQ. NSET2) DTR=DT2
      IF (N.EQ. NSET3) DTR=DT3
      IF (N.EQ. NSET4) DTR=DT4
      IF (N.EQ. NSET5) DTR=DT5
      IF (N.EQ. NSET6) DTR=DT6
      IF (N.EQ. NSET7) DTR=DT7
      DT=DTR/TNORM
      TCH1=10.01*DT $ TCH2=50.01*DT
      IF (N.NE. NSET1 .AND. N.NE. NSET2 .AND. N.NE. NSET3 .AND.
C N.NE. NSET4 .AND. N.NE. NSET5 .AND. N.NE. NSET6 .AND.
C N.NE. NSET7) GO TO 601
      TCH1=TIME+TCH1 $ TCH2=TIME+TCH2
601  CONTINUE
      RHOP=2./DT
C RESET KWAIT AND KSET AS TIME GOES ON.
      KWAIT=20 $ IF (TIME.LT. TCH2) KWAIT=5
      IF (TIME.LT. TCH1) KWAIT=-1
      K=K+1
      KGCO=KGCO+1
C RECOMPUTE GCO ARRAY IF KGCO EQUALS KWAIT.
      IF (KGCO.LT. KWAIT) GO TO 572
      CALL GETGF
      GO TO 571
572  CALL COEF
571  CONTINUE
      REAL=TIME*TNORM $ IF (REAL.GT. STOP) GO TO 1000
      WRITE (3,2) TIME,REAL
      2  FORMAT(5HTIME=,E12.3,5X,10HREAL TIME=,E12.3)
      IF (ABSF(TINSTOP-TIME).GT. .75*DT) GO TO 1001
      DO 779 IC=1,NSOR
      RSI(IC)=1.E-90 $ DCI(IC)=0. $ DCISEI(IC)=0.
      PCURI(IC)=1.E-90 $ DPCURI(IC)=0. $ PCURROSI(IC)=0.
      CURROI(IC)=1.E-90 $ DCURRI(IC)=0.
779  CONTINUE
      COE=0. $ DCESE=0.
      PCURROSE=0. $ PCURROE=0.
      CXRXB(4)=0.
      TCH1=TIME+DT+TCH1 $ TCH2=TIME+DT+TCH2
      CALL DENSINPT
1001 CONTINUE
      DENSITY=GNORM*DENSNO $ ENERGR=ENERGY*EIONS
      RDE=DEIISE*DENSNO $ REE=ENERGE*EIONS
      IF (NCH.EQ. NONCH) NCH=0
      IF (N-NCHEC*(N/NCHEC).GT. 0) GO TO 602
      NCH=NCH+1
      PTIME(NCH)=REAL $ PDENS(NCH,1)=DENSITY $ PDENS(NCH,2)=RDE
      PENGY(NCH,1)=ENERGR $ PENGY(NCH,2)=REE
602  CONTINUE
      WRITE (3,3) DENSITY,RDE,ENERGR,REE,POTENT
      3  FORMAT(5X,8H1ON DEN=,E16.6,5X,9HELEC DEN=,E16.6,/,5X,7H1ON EN=,

```

```

C E16.6,5X,8HELEC EN=,E16.6,5X,7HPOTENT=,E16.6)
TIME=TIME+DT
DO 780 I=1,IY $ DO 780 J=1,JX
780 SOURCE(I,J)=0.
CURR=0.
DO 781 IC=1,NSOR
SOI(IC)=RSI(IC)*(1.+DCI(IC)*GNORM)+DCISEI(IC)*SQRTF(GNORM)
DO 700 I=2,IY
DO 700 J=1,JX
700 SOURCE(I,J)=SOURCE(I,J)+SOI(IC)*SHAPE(I,J,IC)
TOTP=TOTP+DT*SOI(IC)*CII(IC)
CURR=CURR+SOI(IC)*PCURI(IC)/RSI(IC)
781 CONTINUE
NUPOT=XMODF(N,NCPOT)
CALL AMBI
C SUBROUTINE AMBI COMPUTES AND ADJUSTS AMBIPOLAR POTENTIAL, AND CALLS SUBROU-
C TIME MAXE, WHICH ADVANCES ELECTRON EQUATION.
IF (KAMBI .EQ. 1 .OR. KGNANDE .EQ. 1 .OR. KMAXE .EQ. 1) RETURN
CXLOSS=0. $ DO 41 I=1,IY $ DO 41 J=1,JX
CXLOSS=CXLOSS+HCXR(J)*G(I,J)*SPHERI(I,J)
41 CONTINUE
CXLOSS=DENSNO*CXLOSS
IF (TIME .LE. TINSTOP) GO TO 42
CXLOST=CXLOST+CXLOSS*DT*T NORM
42 CONTINUE
IF (K .LT. KSET) GO TO 204
C OUTPUT IF K EQUALS KSET.
CALL OUTPUT $ K=0
GO TO 204
1000 RETURN
END
SUBROUTINE WHY
USE COMM
DEGTOR=180./PI
BRATIO1=1./BRATIO
THETAL=ACOSF(SQRTF(1.-BRATIO1))
DEGREES=THETAL*DEGTOR
WRITE (3,60) BRATIO,THETAL,DEGREES
60 FORMAT(14HMIRROR RATIO =,E12.3,/,16HLOSSCONE ANGLE =,F11.5,
C 8H RAD!ANS,3H = ,F11.5,8H DEGREES,/,1H1,10X,7HRADIANS,7X,
C 7HDEGREES)
C COMPUTE Y MESH.
Y(1) = THETAL
DELY=(PI02-Y(1))/FLOATF(IYM1)
DO 301 I = 2, IY
301 Y(I) = Y(I-1) + DELY
Y(IYP1) = Y(IY) + Y(IY) - Y(IYM1)
C COMPUTE Y-MESH QUANTITIES.
DO 400 I = 1, IYP1
400 DEG(I)=Y(I)*DEGTOR
WRITE (3,20) (I,Y(I),DEG(I),I=1,IYP1)
20 FORMAT (15,2F14.9)
DO 201 I=2,IY
CTNY(I)=1./TANF(Y(I))
201 CTNY2(I)=CTNY(I)*CTNY(I)
DO 203 I = 2, IY
DYP5(I)=Y(I+1)-Y(I)
DYM5(I)=Y(I)-Y(I-1)
DY(I)=.5*(Y(I+1)-Y(I-1))
DYI(I)=1./DY(I)

```

```

      SINN(1)=SINF(Y(1))
203 CONTINUE
      DYP5(1)=DYM5(2)
      SINN(1)=SINF(Y(1))
C COMPUTE X MESH
      X(1)=0. $ H=XMAX/FLOATF(JXM1)
      X(2)=H $ RAM=1.
      DO 3 J=3,JX
        3 X(J)=X(J-1)+RAM*(X(J-1)-X(J-2))
C COMPUTE X-MESH QUANTITIES.
      DO 202 J=2,JXM1
        DXPS(J)=X(J+1)-X(J)
        DXMS(J)=X(J)-X(J-1)
      202 DX(J)=.5*(X(J+1)-X(J-1))
        DX(1)=.5*(X(2)-X(1)) $ DX(JX)=.5*(X(JX)-X(JXM1))
        DXPS(1)=DXMS(2) $ DXMS(JX)=DXPS(JXM1)
        DO 200 J=2,JXM1
          DX1(J)=1./DX(J)
          X21(J)=1./(X(J)*X(J))
          X31(J)=X21(J)/X(J)
      200 X41(J)=X31(J)/X(J)
      DO 102 J=1,JX
      102 XSQ(J)=X(J)*X(J)
C SET UP X-INTEGRATION CONSTANTS.
      F2 = DXPS(2)
      B2=DXMS(2)
      CINT(1)=(2.*B2**3-F2**3+3.*B2**2*F2)/(6.*B2*(F2+B2))
      DO 104 J=2,JXM1,2
        F0=DXPS(J)
        B0=DXMS(J)
      104 CINT(J)=(F0**3+B0**3+3.*F0*B0*(F0+B0))/(6.*F0*B0)
        DO 105 J=3,JXM2,2
          F1=DXPS(J-1)
          B1=DXMS(J-1)
          F2=DXPS(J+1)
          B2=DXMS(J+1)
      105 CINT(J)=(2.*F1**3-B1**3+3.*F1**2*B1)/(6.*F1*(F1+B1))
          1 + (2.*B2**3-F2**3+3.*B2**2*F2)/(6.*B2*(F2+B2))
          F1=DXPS(JXM1)
          B1=DXMS(JXM1)
          CINT(JX)=(2.*F1**3-B1**3+3.*F1**2*B1)/(6.*F1*(F1+B1))
C SET UP Y-INTEGRATION CONSTANTS.
      F2 = DYP5(2)
      B2 = DYM5(2)
      CYNT(1)=(2.*B2**3-F2**3+3.*B2**2*F2)/(6.*B2*(F2+B2))
      DO 204 I = 2, IY ,2
        F0 = DYP5(1)
        B0 = DYM5(1)
      204 CYNT(I)=(F0**3+B0**3+3.*F0*B0*(F0+B0))/(6.*F0*B0)
        DO 205 I = 3, IYM1, 2
          F1 = DYP5(I-1)
          B1 = DYM5(I-1)
          F2 = DYP5(I+1)
          B2 = DYM5(I+1)
      205 CYNT(I)=(2.*F1**3-B1**3+3.*F1**2*B1)/(6.*F1*(F1+B1))
          1 + (2.*B2**3-F2**3+3.*B2**2*F2)/(6.*B2*(F2+B2))
        DO 302 I = 1, IYM1
      302 CYNT(I) = 2. * CYNT(I)
C SET UP TWO-DIMENSIONAL INTEGRATION CONSTANTS.
      DO 103 I = 1, IY

```



```

DO 103 J=1,JX
103 SPHER(1,J) = CYNT(1)*SINF(Y(1)) *CINT(J)*XSQ(J)*TWOPI
DZ=1./FLOATF(LZ-1) $ Z(1)=0.
DO 71 L=2,LZ
71 Z(L)=Z(L-1)+DZ
HP=(BRATIO-1.)/FLOATF(LZ-1) $ PSI(1)=1. $ DO 72 L=2,LZ
72 PSI(L)=PSI(L-1)+HP
DO 73 L=1,LZ
PSI(L)=SQRTF(PSI(L)) $ PS2(L)=SQRTF(PSI(L)/BRATIO)
73 CONTINUE
RETURN
END
SUBROUTINE OUTPUT
USE COMM
REALT=TIME*TNORM
WRITE (3,24) TIME,REALT
24 FORMAT(1H1,5HTIME=,E16.6,5X,11HREAL TIME =,E16.6,9H SECONDS)
WRITE (3,21) DENSNO,VNORM
21 FORMAT(/,52HNORMED DIST. FN. IS GIVEN. IT MUST BE MULTIPLIED BY ,
1/,E16.9,33H/CC., THE DENSITY NORM TIMES (1./,E16.9,10HCM/SEC)**3,
2/,45H THE CUBE OF THE INVERSE OF THE VELOCITY NORM )
WRITE (3,22) (Y(I),I=1,IY)
22 FORMAT(/,45H THETA,/,
1(12F9.4))
WRITE (3,1)
1 FORMAT(1H1,10H J VEL ,10X,12HDISTRIBUTION ,/)
DO 230 J=1,JX
WRITE (3,23) (J,X(J),(G(I,J),I=1,IY))
23 FORMAT(/,13,F7.3,/,16E17.3))
230 CONTINUE
WRITE (3,231) ((J,XE(J),EE(J)),J=1,JXE)
231 FORMAT(1H1,18HELEC DIST FUNCTION,/, (2HJ=,15,5X,3HXE=,E16.6,5X,
C 4HLEE=,E16.6))
DENSITY=GNORM*DENSNO $ ENERGR=ENERGY*EIONS
REPCUR=CURR $ RECURR=REPCUR*CURCON
RDE=DENSE*DENSNO $ REE=ENERGE*EIONS
RETOTP=TOTP*DENSNO
WRITE (3,31) DENSITY,ENERGR,ROE,REE
31 FORMAT(10H ION DEN =,E16.6,5X,12HION ENERGY =,E16.6,5H KEV,
C /,10HELEC DEN =,E16.6,4X,13HELEC ENERGY =,E16.6,5H KfV)
WRITE (3,32) RECURR,REPCUR,RETOTP,POTENT
32 FORMAT(1,7HION SOURCE CURR =,E16.6,12H AMP/CC/SEC,5H = ,
C E16.5,13H PART/CC/SEC,/,22HTOTAL NO OF PART INJ =,E16.6,5X,
C 9HAMB POT =,E16.6)
CALL GETCURE
CURRER=CURRE*CURCON
WRITE (3,33) CURRER,CURRE
33 FORMAT(18HELEC SOURCE CURR =,E16.6,12H AMP/CC/SEC,
C 5H = ,E16.6,13H PART/CC/SEC)
WRITE (3,34) CXLOSS,CXLOST
34 FORMAT(15HCX LOSS TERMS =,2E16.6)
WRITE (3,771) ((L,PSI(L),PPOT(L)),L=1,LZ)
771 FORMAT(/, (2HL=,15,5X,4HPSI=,E16.6,5X,4HPOT=,E16.6))
C SET UP PLOTTING OUTPUT.
CALL SETCH(1.,32.,0,0,2,0,0)
WRITE (100,5) REALT,DENSITY,ENERGR
5 FORMAT(5HTIME=,E10.3,5H SEC.,3X,8HDENSITY=,E10.3,4H/CC.,6HION E=,
1F7.3,3HKEV )
CALL SETCH(1.,31.,0,0,2,0,0)
WRITE (100,7) ROE,REE

```

```

7 FORMAT(23X,BHDENSITY=,E10.3,4H/CC.,6HELE E=,F7.3,3HKEV)
  CALL SETCH(1.,30.,0,0,2,0,0)
  WRITE (100,8) POTENT
8 FORMAT(45X,6HAM PT=,F7.3,3HKEV)
  CALL SETCH(17.,1.,1,0,2 0)
  WRITE (100,6)
6 FORMAT(16HION DISTRIBUTION)
  DO 611 KK=1,2
    PLTANG=45. $ IF (KK .EQ. 2) PLTANG=225.
    PTANG=PLTANG
    DO 41 J=1,JX
      TEMPG(J)=X(J) $ DO 41 I=1,IY
41 GST(I,J)=G(I,J)
    IF (PLTANG .LE. 180.) GO TO 42
    PTANG=PLTANG-180.
    DO 43 J=1,JX
      JJ=JX+1-J $ TEMPG(J)=X(JJ) $ DO 43 I=1,IY
43 GST(I,J)=G(I,JJ)
42 CONTINUE
  CALL PLOT3D(PTANG,TEMPG,Y,GST,1.,1.,1.,JX,IY,IYP1)
611 CONTINUE
  CALL CARTMM(JXE,DMIN,DMAX,FEE,1)
  CALL MAPS(0.,XEMAX,0.,DMAX,.2..999,.2..999)
  CALL SETCH(17.,3.,1,0,1,0)
  WRITE (100,61)
61 FORMAT(30HELECTRON DISTRIBUTION FUNCTION)
  CALL SETCH(17.,1.,1,0,1,0)
  WRITE (100,62) REALT
62 FORMAT(5HTIME=,E16.6,2X,7HSECONDS)
  CALL TRACE(XE,FEE,JXE)
  CALL FRAME
  CALL MAPS(1.,BRATIO,0.,PJTENT,.2..999,.65..999)
  CALL SETCH(1.,40.,1,0,1,1)
  WRITE (100,91)
91 FORMAT(10HPI OF PSI)
  CALL SETCH(17.,1.,1,0,1,0)
  WRITE (100,53) REALT
  CALL TRACE(PSI,PPOT,LZ)
  CALL MAPS(0.,XMAX,Y(1),PI02,.2..999,.15..5)
  DIMENSION THLOS(JXA)
  DO 92 J=1,JX
92 THLOS(J)=Y(10(J))
  CALL TRACE(X,THLOS,JX)
  CALL LINE(0.,Y(1),0.,PI02)
  CALL LINE(0.,PI02,XMAX,PI02)
  CALL LINE(XMAX,PI02,XMAX,Y(1))
  CALL LINE(XMAX,Y(1),0.,Y(1))
  CALL SETCH(1.,10.,1,0,1,1)
  WRITE (100,93)
93 FORMAT(13HLOSS BOUNDARY)
  CALL FRAME
  IF (TIME .EQ. 0.) RETURN
  CALL CARTMM(NCH,DMIN,DMAX,POENS(1,1),1)
  DMAX=1.1*DMAX
  XMINP=PTIME(1) $ XMAXP=PTIME(NCH)
  CALL MAPS(XMINP,XMAXP,0.,DMAX,.2..999,.65..999)
  CALL SETCH(1.,45.,1,0,1,1)
  WRITE (100,51)
51 FORMAT(11HION DENSITY)
  CALL TRACE(PTIME,POENS(1,1),NCH)

```

```

CALL CARTMM(NCH,DMIN,DMAX,PENGY(1,1),1)
DMAX=1.1*DMAX
CALL MAPS(XMINP,XMAXP,DMIN,DMAX,.2,.999,.15,.5)
CALL SETCH(1.,5.,1,0,1,1)
WRITE (100,52)
52 FORMAT(10HION ENERGY)
CALL TRACE(PTIME,PENGY(1,1),NCH)
CALL SETCH(17.,1.,1,0,1,0)
WRITE (100,53) REALT
53 FORMAT(11HREAL TIME =,E16.6,8H SECONDS)
CALL FRAME
CALL CARTMM(NCH,DMIN,DMAX,PDENS(1,2),1)
DMAX=1.1*DMAX
CALL MAPS(XMINP,XMAXP,0.,DMAX,.2,.999,.65,.999)
CALL SETCH(1.,45.,1,0,1,1)
WRITE (100,54)
54 FORMAT(12HELEC DENSITY)
CALL TRACE(PTIME,PDENS(1,2),NCH)
CALL CARTMM(NCH,DMIN,DMAX,PENGY(1,2),1)
DMAX=1.1*DMAX
CALL MAPS(XMINP,XMAXP,0.,DMAX,.2,.999,.15,.5)
CALL SETCH(1.,5.,1,0,1,1)
WRITE (100,55)
55 FORMAT(11HELEC ENERGY)
CALL TRACE(PTIME,PENGY(1,2),NCH)
CALL SETCH(17.,1.,1,0,1,0)
WRITE (100,53) REALT
CALL FRAME
KSTORE=K
DIMENSION CV(IYAP1),CH(IYAP1),PERP(JXA),VPAR(IYAP1)
DO 81 KK=1,2
KP=2*(KK-1)
DO 80 J=2,JXM1
PERP(J)=0.
DO 68 I=1,IY
CV(I)=X(J)/SINF(Y(I)) $ VPAR(I)=CV(I)*COSF(Y(I))
CH(I)=0.
K=J-1
69 K=K+1
IF (K .GT. JXM1) GO TO 68
IF (CV(I) .LT. X(K) .OR. CV(I) .GE. X(K+1)) GO TO 69
CH(I)=G(I,K)+(G(I,K+1)-G(I,K))*(CV(I)-X(K))/(X(K+1)-X(K))
CH(I)=CH(I)*VPAR(I)**KP
68 CONTINUE
CALL INTGRL(IY,CH,VPAR,ANSI)
PERP(J)=-ANSI
80 CONTINUE
IF (KK .EQ. 1) GO TO 83
WRITE (3,82)
82 FORMAT(//,31HTHIS PERP DIST CONTAINS VPAR**2,/)
83 CONTINUE
WRITE (3,75) ((X(J),PERP(J)),J=1,JX)
75 FORMAT(//,35HION VELOCITY PERP. DISTRIBUTION,/,2E16.6)
CALL MAXMIN(PERP,JX,PMAX,PMIN)
CALL SETCH(10.,32.,0,0,2,0)
WRITE (100,76) REALT
76 FORMAT(26HPERP. VELOCITY DISTR. AT ,E10.3,4HSEC.)
CALL MAPG(X(1),X(JX),0.,PMAX,.15,.999,.15,.9)
CALL SETCRT(X(1),PERP(1))
CALL TRACE(X(1),PERP(1),JX)

```

```

      IF (KK .EQ. 1) GO TO 95
      CALL SETCH(10.,1.,1.0,1.0)
      WRITE (100,84)
84  FORMAT(31HTHIS PERP DIST CONTAINS VPAR**2)
85  CONTINUE
      CALL FRAME
81  CONTINUE
      K=KSTORE
      RETURN
      END
      SUBROUTINE AMB1
      USE COMM
      MI = 0
      OLDDEN=DENSE
      OLDE = ENERGE
      OLDI=GNORM
      DO 3 I=1,IYP1 $ DO 3 J=1,JX
3   GST(I,J)=G(I,J)
      DO 13 J=1,JXE
13  FEES(J)=FEE(J)
10  MI=MI+1
      CALL GAMMAIN(1)
      CALL IONISOT
      CALL GELECT
      CALL XSWEPT
      CALL GNADE
      IF (KGNANDE .EQ. 1) RETURN
      POLDI=PNUMB*OLDI $ PGNORM=PNUMB*GNORM $ DEII=PGNORM-POLDI
      CALL MAXE
      IF (KMAXE .EQ. 1) RETURN
      DEEE = DENSE - OLDDEN
      WRITE (3,465)POLDI,PGNORM,DEII,OLDDEN,DENSE,DEEE
465 FORMAT(10X,7HOLD NI=,E16.6,5X,7HNEW NI=,E16.6,5X,5HDIFF=,E16.6,
C 7,10X,7HOLD NE=,E16.6,5X,7HNEW NE=,E16.6,5X,5HDIFF=,E16.6)
      IF (OLDDEN .EQ. POLDI) GO TO 160
      IF (TIME .LT. 1.000001*DT) GO TO 160
      IF (MIMAX .EQ. 0) GO TO 160
      IF (MI .GE. MIMAX) GO TO 460
      IF (DENST .LT. 0.) GO TO 904
      IF (PGNORM .GT. DENSE .AND. DEII .GT. DEEE) 161,160
904 IF (PGNORM .LT. DENSE .AND. DEII .LT. DEEE) 161,160
161 POTENT=POTENT+DENST*POTENT
      DO 16 L=1,LZ
16  PPOT(L)=PPOT(L)*(1.+DENST)
      DENSE = OLDDEN
      ENERGE = OLDE
      GNORM=OLDI
      DO 4 I=1,IYP1 $ DO 4 J=1,JX
4   G(I,J)=GST(I,J)
      DO 14 J=1,JXE
14  FEE(J)=FEES(J)
      GO TO 10
160 CONTINUE
      IF (NUPOT .EQ. 0) CALL POTSHAPE
      CALL BOUNDARY
      CALL GNADE
      PGNORM=PNUMB*GNORM
      IF (PGNORM .LT. DENSE .AND. DENST .GT. 0.) DENST=-DENST
      IF (PGNORM .GT. DENSE .AND. DENST .LT. 0.) DENST=-DENST
      RETURN

```

```

460 WRITE (3,461) POTENT
461 FORMAT(/,36HAMB POT REQ NOT SATISFIED - POTENT =,E16.6)
KAMB1=1 $ RETURN
END
SUBROUTINE GERR
USE COMM
DO 10 J=1,JX
DO 10 I=1,IYP1
IF (G(I,J) .GE. 0.) GO TO 10
G(I,J)=0.
10 CONTINUE
RETURN
END
SUBROUTINE COEF
USE COMM
C THIS SUBROUTINE COMPUTES SOME OF THE COEFFICIENTS FOR THE ION EQUATION.
GNI=1./GNORM
FACTOR=.5*TNORM*GNI
CHBET=3.491E-31 $ CHA=1.111E+15 $ CHB=1.351E+14
CHC=5.099E-2 $ CHD=2.092E-53 $ CHV1=1.000E+8
DO 51 J=2,JX
ENUM=1.-EXP(-CHBET*(X(J)*VNORM)**4)
DENOM=CHA+CHB*ABSF(LOGF(VNORM*X(J)/CHV1))+CHC*(VNORM*X(J)-CHV1)**2
C +CHD*(VNORM*X(J))**8
HCXR(J)=CXRXB(1)*VNORM*X(J)*ENUM/DENOM
HCXR(J)=HCXR(J)+CXRXB(4)
51 CONTINUE
PGNI=TWOPI*GNI*GAM11
DO 100 I=2,IY
DO 101 J=2,JXM1
A1P=.5*GCOXX(1,J) $ DGX=GCOXY(1,J) $ DGY=GCOYY(1,J)
DGX=GCOX(1,J) $ DGY=GCOY(1,J)
A2P(1,J)=X21(J)*DGX-X31(J)*DGY
A3P=.5*(X41(J)*DGY+X31(J)*DGX)
B1P=.5*X31(J)*(DGY+CTNY(1)*DGY)+X21(J)*DGX
B2P=X41(J)*DGY-X31(J)*DGX+.5*X31(J)*CTNY(1)*DGX+.5*
1X41(J)*CTNY2(1)*DGY
ALPHAP(1,J)=(A1P/DXP5(J)+B1P*.5)*DX1(J)
BETAP(1,J)=(A1P/DXM5(J)-B1P*.5)*DX1(J)
GAMMAP(1,J)=(A3P/DYP5(1)+B2P*.5)*DY1(1)
DELTAP(1,J)=(A3P/DYM5(1)-B2P*.5)*DY1(1)
CPB2(1,J)=PGNI*G(1,J)
CPB2(1,J)=CPB2(1,J)-HCXR(J)*FACTOR
101 CONTINUE
100 CONTINUE
RETURN
END
SUBROUTINE GNANDE
USE COMM
C THIS SUBROUTINE COMPUTES THE NORMALIZED ION DENSITY AND ENERGY.
EN=0.
DO 100 J = 1,JX
DO 100 I = 1, IY
100 EN=EN+G(1,J)*XSQ(J)*SPHERI(1,J)
EEN=EN
GN=0.
DO 101 J=1,JX
DO 101 I=1,IY
GN=GN+G(1,J)*SPHERI(1,J)
101 CONTINUE

```

```

      GGN=GN
      IF (GGN.EQ. 0.) GO TO 1
      ENERGY=EEN/GGN
      GNORM=GN
      RETURN
1  KGNANDE=1
   WRITE (3,2)
2  FORMAT(/,12HZERO DENSITY)
   RETURN
   END
   SUBROUTINE XSWEEP
   USE COMM
   DO 3333 I=1,IYJXA
3333  TEMPG(I)=0.
      GNELOG=GNORM
      GNELOG1=1./GNELOG
      RBGN = RHOP/GNELOG
      CALL BOUNDARY
C  FIRST COMPUTE ALONG I=CONST ROWS.
C  STORE ROW IMIN OF ARRAY G.
      DO 111 J=JMP1,JX
111  TEMPG(J)=0.
      DO 100 I=1MP1,IY
      JS=JO(I)
      JSP=JS+1
      JXMJP=JX-JSP
C  COMPUTE E AND F (EP AND FP) FOR ADI SCHEME.
      EP(JS)=0.
      FP(JS)=0.
      DMODP=DYM5(I)/DYP5(I)
      DO 101 J=JSP,JXMI
      ALP1=GEXX(J)/(2.*DXP5(J)*DX(J))
      BET1=ALP1*DXP5(J)/DXM5(J)
      ALBE2=(CEX(J)+GEX(J)*X21(J))/(2.*DX(J))
      GAMI=GEX(J)*X31(J)/(2.*DYP5(I)*DY(I))
      DEL1=GAMI*DYP5(I)/DYM5(I)
      GADE2=GEX(J)*CTNY(I)*X31(J)/(4.*DY(I))
      CCI=THOP1*FE(J)/PNUMB
      COMIX1 = A2P(I,J)/(4.*DX(J)*DYM5(I))
      COMIXE=COMIX1*DMODP
      BMIX= BETAP(I,J)+BET1-ALBE2-COMIX1
      BP= ALPHAP(I,J)+BETAP(I,J)+ALP1+BET1-CPB2(I,J)-CCI +RBGN
      PT1=(GAMMAP(I,J)+GAMI+GADE2)*(G(I+1,J)-G(I,J))+ (DELTA P(I,J)+DEL1-
C  GADE2)*(G(I-1,J)-G(I,J))+ (CPB2(I,J)+CCI)*G(I,J)+
C  COMIX1*(TEMPG(J-1)-TEMPG(J+1))+COMIXE*(G(I+1,J+1)-G(I+1,J-1)-
C  G(I,J+1)+G(I,J-1))+RBGN*G(I,J)+SOURCE(I,J)*GNELOG1
      DEN=BP-BMIX*EP(J-1)
      EP(J)=(ALPHAP(I,J)+ALP1+ALBE2+COMIX1)/DEN
101  FP(J)=(PT1+BMIX*FP(J-1))/DEN
      TEMPG(JX)= 0.
C  TEMPG NOW CONTAINS NEWLY COMPUTED ROW I-1.  SET G ON ROW I-1 EQUAL TO TEMPG
C  AND COMPUTE G ON ROW I AND STORE IN TEMPG.
      DO 120 J=JMP1,JXMI
120  G(I-1, J) = TEMPG(J)
      DO 102 J=1,JXMJP
102  TEMPG(JX-J)=EP(JX-J)*TEMPG(JX-J+1)+FP(JX-J)
      DO 226 J=1,JS
226  TEMPG(J)=0.
100  CONTINUE
      DO 110 J=JMP1,JXMI

```

```

      G(IY, J) = TEMPG(J)
      G(IYI, J) = G(IYI, J)
110 CONTINUE
C   NOW COMPUTE ALONG J=CONST ROWS.
C   STORE ROW JMIN OF ARRAY G.
      DO 112 I=JMPI, IYPI
112 TEMPG(I)=0.
C   COMPUTE E AND F (EP AND FP) FOR ADI SCHEME.
      DO 105 J=JMPI, JXMI
        EP(I0(J))=0.
        FP(I0(J))=0.
        ISM=I0(J)-1
        IST=I0(J)+1
        ALP1=GEXX(J)/(2.*DXP5(J)*DX(J))
        BET1=ALP1*DXP5(J)/DXM5(J)
        ALBE2=(CEX(J)+GEX(J)*X2I(J))/(2.*DX(J))
        CC1=THOP1*FE(J)/PNUMB
        DO 106 I=IST, IY
          GAMI=GEX(J)*X3I(J)/(2.*DYP5(I)*DY(I))
          DEL1=GAMI*DYP5(I)/DYM5(I)
          GADE2=GEX(J)*CTNY(I)*X3I(J)/(4.*DY(I))
          COMIX1 = A2P(I, J)/(4.*DY(I)*DXM5(J))
          COMIXE = COMIX1*DXM5(J)/DXP5(J)
          DMIX= DELTAP(I, J)+DEL1-GADE2-COMIX1
          BP= GAMMAP(I, J)+DELTAP(I, J)+GAMI+DEL1-CPB2(I, J)-CC1 + RBGN
          CH1= (ALPHAP(I, J)+ALP1+ALBE2)*(G(I, J+1)-G(I, J))+(BETAP(I, J)+BET1-
            1ALBE2)*(-G(I, J)+G(I, J-1))+(CPB2(I, J)+CC1)*G(I, J)+COMIX1*(-TEMPG
            2(I+1)+TEMPG(I-1))+COMIXE*(G(I+1, J+1)-G(I-1, J+1)-G(I+1, J)+G(I-1, J))
          3 +RBGN*G(I, J) + SOURCE(I, J)/GNELOG
          DEN=BP-DMIX*EP(I-1)
          EP(I)=(GAMMAP(I, J)+GAMI+GADE2+COMIX1)/DEN
106 FP(I) = (CH1 + DMIX * FP(I-1))/DEN
C   TEMPG NOW CONTAINS NEWLY COMPUTED ROW J-1.  SET G ON ROW J-1 EQUAL TO TEMPG
C   AND COMPUTE G ON ROW J AND STORE IN TEMPG.
      DO 121 I=JMPI, IYPI
121 G(I, J-1) = TEMPG(I)
      TEMPG(IYPI) = (EP(IYI)*FP(IY) + FP(IYI))/(1.- EP(IYI)*EP(IY))
      IYM=IYPI-I0(J)
      DO 107 I=1, IYM
107 TEMPG(IYPI-I) = EP(IYPI-I)*TEMPG(IYPI-I+1) + FP(IYPI-I)
      DO 228 I=1, ISM
228 TEMPG(I) = 0.
105 CONTINUE
      DO 113 I=JMPI, IYPI
        G(I, JXMI) = TEMPG(I)
113 CONTINUE
      CALL GERR
      RETURN
      END
      SUBROUTINE PREEL
      USE COMM
      BEE=.75*EMASS*AMU*VNORM**2/(ECON*EIONS*ENERGE)
      XEMAX=SQRTF(-LOGF(FEMINI/BEE))
      IF (XEMAX.LT. XMAX) XEMAX=XMAX
      H=XEMAX/FLOATF(JXEMI)
      XE(1)=0. $ XE(JXE)=XEMAX
      XE(2)=H $ RAM=1.
      DO 1 J=3, JXEMI
1  XE(J)=XE(J-1)+RAM*(XE(J-1)-XE(J-2))
      WRITE (3,10) ((J,XE(J)),J=1,JXE)

```

```

10 FORMAT(//,7HXE MESH,//,(15,E16.6))
DO 2 J=2,JXE
2 DXEM(J)=XE(J)-XE(J-1)
DXEM(1)=0. $ DXEM(JXE+1)=0.
DO 3 J=1,JXE
XE2(J)=XE(J)**2
3 DXE(J)=.5*(DXEM(J)+DXEM(J+1))
DO 4 J=1,JXE
4 FEE(J)=EXP(-BEE*XE2(J))
S=0. $ DO 5 J=1,JXE
5 S=S+FEE(J)*XE2(J)*DXE(J)
S=FOURPI*S
FACT=DENSE/S
DO 6 J=1,JXE
6 FEE(J)=FACT*FEE(J)
XOS2=EES/(EMOIM*EIONS)
XOS=SQRTF(XOS2)
DO 7 J=1,JXE
7 DELE(J)=EXP(-XSOE*(XE(J)-XOS)**2)
S=0. $ DO 8 J=1,JXE
8 S=S+DELE(J)*XE2(J)*DXE(J)
FACT=TNORM/(FOURPI*DENSNO*S)
DO 9 J=1,JXE
9 DELE(J)=FACT*DELE(J)
COE=PCURROE $ DCOE=DPCURROE*DENSNO
DCESE=PCURROE*SQRTF(DENSNO)
RETURN
END
SUBROUTINE GAMMAIN(INDEX)
USE COMM
IF (INDEX.EQ.1) GO TO 1
ALP=1./137. $ R0=2.82E-13 $ CLIGHT=3.E+10
GACON1=2.*ALP/(SQRTF(3.*PI)*R0*CLIGHT)
GACON2=GACON1/(3.7943E-4*EMASS*AMU)
GAM1IS=.5*PMASS*AMU*GACON2
GAMIES=PMASS*EMASS*AMU*GACON2/(PMASS*EMASS)
GAMEIS=GAMIES
GAMEES=.5*EMASS*AMU*GACON2
1 CONTINUE
DEBY=SQRTF(ENERGE*EIONS*.1E+3/(DENSE*DENSNO))
SI=ENERGY/PMASS $ SE=ENERGE/EMASS
SE1=SI $ IF (SE.GT. SI) SE1=SE
EEA=EIONS*ECON/AMU
SI=SI*EEA $ SE=SE*EEA $ SE1=SE1*EEA
SIS=SQRTF(SI) $ SES=SQRTF(SE) $ SEIS=SQRTF(SE1)
GAM3=GAM1IS*DEBY*SIS $ GAM1=LOGF(GAM3)-.5
GAM3=GAMIES*DEBY*SEIS $ GAMIE=LOGF(GAM3)-.5
GAME1=GAMIE
GAM3=GAMEES*DEBY*SES $ GAME2=LOGF(GAM3)-.5
RETURN
END
SUBROUTINE GELECT
USE COMM
CALL MOMENTS
EMCON=1.-1./EMOIM
DO 1 J=2,JXM1
EFE=FE1(J)
FE(J)=EFE*GAMIE/(GNORM*PNUMB*EMOIM)
EMOFE=EMOF(J) $ ENOFE=EMOF(J) $ EEOF=EEOF(J)
CEX(J)=-GAMIE*FOURPI*ENOFE*EMCON/(GNORM*XSG(J)*PNUMB**2)

```



```

GEX(J)=GAMIE*FOURP1*((2.*X(J)*EMOFE-X2I(J)*EEOFE)/3.+EMOFE)/
C (GNORM*PNUMB**2)
GEXX(J)=GAMIE*FOURP1*2.*(EEOFE*X3I(J)+EMOFE)/(3.*GNORM*PNUMB**2)
1 CONTINUE
RETURN
END
SUBROUTINE MOMENTS
USE COMM
DO 1 J=1,JX
DO 2 JJ=1,JXEM1
IF (X(J) .LT. XE(JJ) .OR. X(J) .GE. XE(JJ+1)) GO TO 2
JP=JJ $ GO TO 3
2 CONTINUE
JP=JXEM1
3 CONTINUE
W1=(X(J)-XE(JP))/DXEP(JP) $ W0=1.-W1
FE1(J)=W0*FEE(JP)+W1*FEE(JP+1)
1 CONTINUE
S=0. $ DO 19 JJ=JP,JXEM1
J=JP+JXEM1-JJ
SM=.5*DXEP(J)*(XE(J)*FEE(J)+XE(J+1)*FEE(J+1))
S=S+SM
19 CONTINUE
EM1=S-W1*SM
EMOF(JX)=EM1 $ ENOF(1)=0. $ EEOF(1)=0.
DO 4 J=2,JX
JJ=JX+1-J
T1=.5*DXP5(JJ)*(X(JJ)*FE1(JJ)+X(JJ+1)*FE1(JJ+1))
T2=.5*DXM5(J)*(XSQ(J-1)*FE1(J-1)+XSQ(J)*FE1(J))
T4=.5*DXM5(J)*(XSQ(J-1)**2*FE1(J-1)+XSQ(J)**2*FE1(J))
EMOF(JJ)=EMOF(JJ+1)+T1 $ ENOF(J)=ENOF(J-1)+T2
EEOF(J)=EEOF(J-1)+T4
4 CONTINUE
RETURN
END
SUBROUTINE IONISOT
USE COMM
DO 1 J=1,JX
S=0. $ DO 2 I=1,IY
2 S=S+CYNT(I)*SINN(I)*G(I,J)
1 TEMPG(J)=.5*S
DO 5 J=1,JXE
DO 3 JJ=1,JXM1
IF (XE(J) .LT. X(JJ) .OR. XE(J) .GE. X(JJ+1)) GO TO 3
JP=JJ $ GO TO 4
3 CONTINUE
FIE(J)=0. $ GO TO 5
4 CONTINUE
W1=(XE(J)-X(JP))/DXP5(JP) $ W0=1.-W1
FIE(J)=W0*TEMPG(JP)+W1*TEMPG(JP+1)
5 CONTINUE
RETURN
END
SUBROUTINE GETCURE
USE COMM
CURRE=COE+DCOE*DENSE+DCESE*SQRTE(DENSE)
REE=EIONS*ENERGE $ RDE=DENSNO*DENSE $ DNNE=CXRXB(2)
REE2=REE**2
SIGA=9.109E+4 $ SIGB=6.535E+5 $ SIGC=1.330E+7
SIGVNN=REE2/(SIGA+SIGB*REE+SIGC*REE2)

```

```

CURRE=CURRE+SIGVNNE*DNNE*RDE
RETURN
END
SUBROUTINE MAXE
USE COMM
TECON=FOURP1/(PNUMB**2*EMOIM)**2
TIE=TECON*PNUMB**2*GAME1 $ TEE=TECON*GAMEE
VC2=POTENT/EIONS $ XCN2=VC2/EMOIM $ XCN=SQRTF(XCN2)
ONODT=1./DT
CALL GETCURE
ENFI=0. $ EEFI=0. $ ENFE=0. $ EEFE=0.
ALPE(1)=0. $ BETE(1)=0. $ GAME(1)=0.
DO 1 J=2,JXE
ENFI=ENFI+.5*DXEM(J)*(XE2(J-1)*FIE(J-1)+XE2(J)*FIE(J))
ENFE=ENFE+.5*DXEM(J)*(XE2(J-1)*FEE(J-1)+XE2(J)*FEE(J))
EEFI=EEFI+.5*DXEM(J)*(XE2(J-1)**2*FIE(J-1)+XE2(J)**2*FIE(J))
EEFE=EEFE+.5*DXEM(J)*(XE2(J-1)**2*FEE(J-1)+XE2(J)**2*FEE(J))
ALPE(J)=TIE*ENFI*EMOIM+TEE*ENFE
BETE(J)=(TIE*EEFI+TEE*EEFE)/(3.*XE(J))
GAME(J)=TIE*(ENFI-EEFI/(3.*XE2(J)))+TEE*(ENFE-EEFE/(3.*XE2(J)))
1 CONTINUE
EMFI=0. $ EMFE=0.
DO 2 JJ=2,JXE
J=JXEP1-JJ+1
EMFI=EMFI+.5*DXEP(J)*(XE(J)*FIE(J)+XE(J+1)*FIE(J+1))
EMFE=EMFE+.5*DXEP(J)*(XE(J)*FEE(J)+XE(J+1)*FEE(J+1))
BETE(J)=BETE(J)+XE2(J)*(TIE*EMFI+TEE*EMFE)/3.
PROB=0. $ IF (XE(J) .LE. XCN) GO TO 3
REFF=BRATIO/(1.-XCN2/XE2(J))
PROB=2.3025851/LOGF(REFF)
3 CONTINUE
GAME(J)=GAME(J)+2.*XE(J)*(TIE*EMFI+TEE*EMFE)/3.
GAME(J)=.5*GAME(J)*PROB/XE(J)
2 CONTINUE
DO 4 J=2,JXEM1
ANP2=.25*(ALPE(J)+ALPE(J+1)) $ ANM2=.25*(ALPE(J-1)+ALPE(J))
BNP2=.5*(BETE(J)+BETE(J+1))/DXEP(J)
BNM2=.5*(BETE(J-1)+BETE(J))/DXEM(J)
AA(J)=(ANP2+BNP2)/(XE2(J)*DXE(J))
BB(J)=ONODT-((ANP2-ANM2-BNP2-BNM2)/DXE(J)-GAME(J))/XE2(J)
CC(J)=(-ANM2+BNM2)/(XE2(J)*DXE(J))
DD(J)=ONODT*FEE(J)+CURRE*DELE(J)
4 CONTINUE
ANP2=.5*ALPE(2) $ BNP2=BETE(2)/DXEM(2)
EE(1)=(BNP2+ANP2)/(BNP2-ANP2)
FF(1)=0.
DO 11 J=2,JXEM1
DEN=BB(J)-CC(J)*EE(J-1)
EE(J)=AA(J)/DEN
FF(J)=(DD(J)+CC(J)*FF(J-1))/DEN
11 CONTINUE
ANP2=.5*(ALPE(JXEM1)+ALPE(JXE))
BNP2=(BETE(JXEM1)+BETE(JXE))/DXEP(JXEM1)
ABN2=(BNP2+ANP2)/(BNP2-ANP2)
FEE(JXE)=FF(JXEM1)/(ABN2-EE(JXEM1))
DO 12 JJ=1,JXEM1
J=JXE-JJ
FEE(J)=EE(J)*FEE(J+1)+FF(J)
12 CONTINUE
DO 14 J=1,JXE

```

```

      IF (FEE(J) .LT. 0.) FEE(J)=0.
14  CONTINUE
      S2=0. $ S4=0.
      DO 13 J=1,JXE
      S2=S2+DXE(J)*XE2(J)*FEE(J)
      S4=S4+DXE(J)*XE2(J)**2*FEE(J)
13  CONTINUE
      IF (S2 .EQ. 0) GO TO 15
      DENSE=S2*FOURPI
      ENERGE=EMOIM*S4/S2
      RETURN
15  KMAXE=1
      RETURN
      END
      SUBROUTINE QHOLD2
      USE COMM
      DIMENSION SIGMA(JXA)
      DO 1 J=1,JX
      S=0.
      DO 2 I=1,IY
2  S=S+CYNT(I)*SINF(Y(I))*G(I,J)
1  TEMPG(J)=S
      WRITE (3,16) ((J,TEMPG(J)),J=1,JX)
16  FORMAT(//,10HVELOC DIST./,(15,E16.6))
      UM2=5./3.
      UM=SQRTF(UM2)
      S=0.
      DO 3 J=1,JX
3  S=S+CINT(J)*XSQ(J)*TEMPG(J)
      DENS=S
      FACTOR=UM*VNORM/DENS
      UMU=UM2*EIONS
      CALL FINDSIG(UMU,SIGMA)
      DO 4 J=1,JX
      ENGY=UMU*XSQ(J)
      WRITE (3,5) ENGY,SIGMA(J)
5  FORMAT(7HENERGY=,E16.6,5X,4HSIG=,E16.6)
4  CONTINUE
      S=0.
      DO 6 J=1,JX
      S=S+CINT(J)*SIGMA(J)*TEMPG(J)*X(J)*XSQ(J)
6  CONTINUE
      SIGV=FACTOR*S
      WRITE (3,7) SIGV
7  FORMAT(/,17HHOLDREN SIGMAVEE=,E16.6)
      RETURN
      END
      SUBROUTINE QFUTCH2
      USE COMM
      COMMON/MQCAL/LT,TEMPGT(JXA)
      DIMENSION SIGMA(JXA)
      DIMENSION ANSWER(2)
      DO 1 J=1,JX
      S=0.
      DO 2 I=1,IY
2  S=S+CYNT(I)*SINF(Y(I))*G(I,J)
1  TEMPG(J)=S
      SQIP5=SQRTF(1.5)
      TEMPGT(I)=TEMPG(I)
      DO 162 J=2,JX

```

```

      QJX3=SQIP5*X(J)
      DO 163 JJ=1,JXM1
      IF (X(JJ) .LE. QJX3 .AND. QJX3 .LT. X(JJ+1)) GO TO 164
163  CONTINUE
      TEMPGT(J)=0. $ GO TO 162
164  RATIO=(QJX3-X(JJ))/DXP5(JJ)
      TEMPGT(J)=(1.-RATIO)*TEMPG(JJ)+RATIO*TEMPG(JJ+1)
162  CONTINUE
      WRITE (3,631) ((JL,TEMPG(JL),TEMPGT(JL)),JL=1,JX)
631  FORMAT(15,2E16.6)
      UMU=EIONS
      DO 30 LT=1,2
      IF (LT .EQ. 1) GO TO 31
      CALL FINDSIG(UMU,SIGMA)
      DO 34 J=1,JX
      ENGY=UMU*XSQ(J)
      WRITE (3,35) ENGY,SIGMA(J)
35  FORMAT(7HENERGY=,E16.6,5X,4HSIG=,E16.6)
34  CONTINUE
      GO TO 32
31  CONTINUE
      DO 33 J=1,JX
33  SIGMA(J)=1.
32  CONTINUE
      QANS=0.
      DO 20 J=1,JX
20  QANS=QANS+CINT(J)*EVALS(J,SIGMA)
      ANSWER(LT)=QANS
30  CONTINUE
      SIGV=VNORM*ANSWER(2)/ANSWER(1)
      WRITE (3,7) SIGV
7  FORMAT(/,15HFUTCH SIGMAVEE=,E16.6)
      RETURN
      END
      FUNCTION EVALS(J,SIGMA)
      USE COMM
      COMMON/MQCAL/LT,TEMPGT(JXA)
      DIMENSION SIGMA(1)
      DIMENSION STFG(JXA),UPART(JXA)
      DO 1 KK=1,J
1  STFG(KK)=(XSQ(J)-XSQ(KK))*GETFG(J,KK)
      UPART(1)=0.
      DO 2 KK=2,J
      ADDIT=.5*DXM5(KK)*(STFG(KK-1)+STFG(KK))
      UPART(KK)=UPART(KK-1)+ADDIT
2  CONTINUE
      SUM=0.
      DO 4 KK=2,J
      ARGM=X(KK-1)**LT1*SIGMA(KK-1)*UPART(KK-1)
      ARGP=X(KK)**LT*SIGMA(KK)*UPART(KK)
      SUM=SUM+.5*DXM5(KK)*(ARGM+ARGP)
4  CONTINUE
      EVALS=SUM
      RETURN
      END
      FUNCTION GETFG(J,KK)
      USE COMM
      COMMON/MQCAL/LT,TEMPGT(JXA)
      ARG1=.5*(X(J)+X(KK))
      ARG2=.5*(X(J)-X(KK))

```

```

J1=0 $ J2=0
DO 1 JM=1,JXM1
IF (X(JM) .LE. ARG1 .AND. ARG1 .LT. X(JM+1)) J1=JM
IF (X(JM) .LE. ARG2 .AND. ARG2 .LT. X(JM+1)) J2=JM
1 CONTINUE
IF (ARG1 .GE. X(JX)) J1=JXM1
IF (ARG2 .GE. X(JX)) J2=JXM1
FRAC1=(ARG1-X(J1))/DXP5(J1)
FRAC2=(ARG2-X(J2))/DXP5(J2)
ANS1=(1.-FRAC1)*TEMPG(J1)+FRAC1*TEMPG(J1+1)
ANS2=(1.-FRAC2)*TEMPGT(J2)+FRAC2*TEMPGT(J2+1)
ANS3=(1.-FRAC2)*TEMPG(J2)+FRAC2*TEMPG(J2+1)
ANS4=(1.-FRAC1)*TEMPGT(J1)+FRAC1*TEMPGT(J1+1)
GETFG=ANS1*ANS2+ANS3*ANS4
RETURN
END
SUBROUTINE FINDSIG(UMU,SIGMA)
USE COMM
DIMENSION SIGMA(1)
SIGMA(1)=0.
DO 1 J=2,JX
EPS=UMU*X(J)**2
IF (EPS .GE. 20.) GO TO 41
TEMPQ=2.25E+4*EXP(-44.4/SQRTF(EPS))/EPS $ GO TO 40
41 IF (EPS .GE. 30.) GO TO 42
TEMPQ=4.62E+4*EPS*EPS-0.137 $ GO TO 40
42 IF (EPS .GE. 50.) GO TO 43
TEMPQ=6.70E+4*EPS*EPS-0.320 $ GO TO 40
43 IF (EPS .GE. 87.) GO TO 44
TEMPQ=8.55E+02*EPS-2.93 $ GO TO 40
44 IF (EPS .GE. 107.) GO TO 45
TEMPQ=1.98E+4/((EPS-107.)*2+4000.) $ GO TO 40
45 IF (EPS .GE. 137.) GO TO 46
TEMPQ=3.56E+4/((EPS-107.)*2+7200.) $ GO TO 40
46 IF (EPS .GE. 185.) GO TO 47
TEMPQ=8.70-0.0314*EPS $ GO TO 40
47 IF (EPS .GE. 400.) GO TO 48
TEMPQ=7.050E+3/(EPS*SQRTF(EPS)) $ GO TO 40
48 TEMPQ=255./(EPS-110.)
49 CONTINUE
SIGMA(J)=1.E-24*TEMPQ
1 CONTINUE
RETURN
END
SUBROUTINE RESTART
USE COMM
CALL QHOLD2
CALL QFUTCH2
CALL PLOTE
CALL OFFMON
DIMENSION ND(5) $ ND(1)=0 $ ND(2)=1HF $ ND(3)=6 $ ND(4)=0
ND(5)=1 $ CALL HSP(ND)
CALL CRTID(10HAM RESTART,1,1)
RETURN
END
SUBROUTINE OFFMON
COMMON/GOBCOM/IX(4)
IX(24)=IX(24).INT. 77777777777777777777
IADD=.LOC. RSTRT
CALL QUIT(IADD)

```

```

RSTRT RETURN
END
SUBROUTINE INTGRL(N,F,EP,ANS)
DIMENSION EP(N),F(N),CONST(100)
HPLUS=EP(3)-EP(2)
HMINUS=EP(2)-EP(1)
CONST(1)=(2.*HMINUS**3-HPLUS**3+3.*HMINUS**2*HPLUS)/
C (6.*HMINUS*(HPLUS+HMINUS))
NN1=N-1
NN2=N-2
DO 11 I=3,NN2,2
HPLUS=EP(I)-EP(I-1)
HMINUS=EP(I-1)-EP(I-2)
HPLUSP=EP(I+2)-EP(I+1)
HMINUSP=EP(I+1)-EP(I)
CONST(I)=(2.*HPLUS**3-HMINUS**3+3.*HPLUS**2*HMINUS)/
C (6.*HPLUS*(HPLUS+HMINUS)) +
C (2.*HMINUSP**3-HPLUSP**3+3.*HMINUSP**2*HPLUSP) /
C (6.*HMINUSP*(HPLUSP+HMINUSP))
11 CONTINUE
A=.5*N
B=N/2
L=0
IF ((A-B).GT. 1.E-3) L=1
C IF L=1 N IS ODD.
I=N-1
IF (L.EQ. 1) I=N
HPLUS=EP(I)-EP(I-1)
HMINUS=EP(I-1)-EP(I-2)
CONST(1)=(2.*HPLUS**3-HMINUS**3+3.*HPLUS**2*HMINUS) /
C (6.*HPLUS*(HPLUS+HMINUS))
IF (L.EQ. 0) CONST(1)=CONST(1)+(EP(N)-EP(N-1))/3.
IF (L.EQ. 0) CONST(N)=L.
DO 12 I=2,NN1,2
HPLUS=EP(I+1)-EP(I)
HMINUS=EP(I)-EP(I-1)
CONST(I)=(HPLUS**3+HMINUS**3+3.*HPLUS*HMINUS*(HPLUS+HMINUS)) /
C (6.*HPLUS*HMINUS)
12 CONTINUE
C CONSTANTS NOW COMPLETE--I=1,N.
SUM=0.
DO 14 I=1,N
SUM=SUM+CONST(I)*F(I)
14 CONTINUE
ANS=SUM
RETURN
END
SUBROUTINE MAXMIN(X,N,X1,X0)
C
C C TO FIND MAXIMUM AND MINIMUM VALUES OF ARRAY X
C
P DIMENSION X(8192)
BIG=X(1)
DO 1 I=2,N
1 BIG=MAX1F(BIG,X(I))
X1=BIG
SML=X(1)
DO 2 I=2,N
2 SML=MIN1F(SML,X(I))
X0=SML

```

Appendix D. Listing of HYBRID-1-2X

```

PROGRAM DUMFG(TAPE2,TAPE6,TAPE3=TAPE6,PRINT)
CLICHE COMM
OPTIMIZE
PARAMETER(JXA=45, IYA=22)
PARAMETER(JXE=251)
PARAMETER(NSOR=4)
PARAMETER(MX=4)
PARAMETER(MMX=2*MX, MXP3M=-MMX-3, MXP4=MMX+4)
PARAMETER(JXEP1=JXE+1)
PARAMETER(IYA2=2*IYA-1, IYA02=IYA/2, IYAM1=IYA-1, IYAP1=IYA+1)
PARAMETER(IYJXA=IYA+JXA)
PARAMETER(NONCH=200)
COMMON/MPL0T/PTIME(NONCH), PDENS(NONCH,2), PENGY(NONCH,2)
C ,NCH
COMMON/MCON1/TIME,STOP,DT,AMU,ECON,RHOP
C ,PI02,TWOPI,FOURPI,CHARGE,KSET,KWAIT,K,KGCO,PI,XEMAX
C ,TWOPI,XMAX,KGETGFK,DTR
C ,CXRXB(4),CXLOSS,CXLOST,HCXR(JXA)
C ,PLTANG
COMMON/MCOEF/CPB2(IYAP1,JXA),A2P(IYAP1,JXA),ALPHAP(IYAP1,JXA)
C ,BETAP(IYAP1,JXA),GAMMAP(IYAP1,JXA),DELTAP(IYAP1,JXA)
C ,SHAPE(IYAP1,JXA,NSOR)
COMMON/MCON2/DENSE,ENERGE,EMOIM,PMASS,EMASS,EES,EIONS,TNORM
C ,VNORM,DENSNO,GNORM,ENERGY,PNUMB,KMAXE,KGNANDE,KAMBI,NRSTR
C ,FEMIN,JXEM1,JXEM2,GAM11,GAM1E,GAME1,GAMEE,XSOE
C ,KNPROBS
COMMON/MD1ST/FE(JXA),CEX(JXA),GEX(JXA),GEXX(JXA),TEMPG(IYJXA)
C ,EP(IYJXA),FP(IYJXA),G(IYAP1,JXA),GCO(IYAP1,JXA),FE1(JXA)
C ,EMOF(JXA),ENOF(JXA),EEOF(JXA)
COMMON/MCURAMB/CURCON,CURRE,CURR,DCOE,COE,SOURCE(IYAP1,JXA),
C ,TOTP,POTENT,DENST,IO(JXA),JO(IYAP1),BRATIO,BRATIOI,MIMAX,
C ,TINJSTOP,PCURROE,DPCURROE,PCURROSE,DCESE,CURRO1(NSOR),
C ,DCURRI(NSOR),PCURRI(NSOR),DPCURRI(NSOR),PCURROSI(NSOR),
C ,RS1(NSOR),DC1(NSOR),DCISE1(NSOR),CI(NSOR),ESOR1(NSOR),
C ,XS01(NSOR),YS01(NSOR),US01(NSOR),EO1(NSOR),SO1(NSOR)
C ,GST(IYAP1,JXA)
COMMON/MESH/IYP1,IY,IYM1,IYM2,JX,JXM1,JXM2,X(JXA),DXP5(JXA)
C ,DXM5(JXA),DX(JXA),DXI(JXA),Y(IYAP1),DYP5(IYA),DYM5(IYA),DY(IYA)
C ,DYI(IYA),CYNT(IYAP1),CINI(JXA),CTNY(IYAP1),CTNY2(IYAP1)
C ,SPHERI(IYA,JXA),X2I(JXA),X3I(JXA),X4I(JXA),XSQ(JXA),VEL(JXA)
C ,DEG(IYAP1),SINN(IYA)
COMMON/MJXE/XE(JXE),XE2(JXE),DXE(JXE),DXEM(JXE),FEE(JXE)
C ,FIE(JXE),ALPE(JXE),BETE(JXE),GAME(JXE),DELE(JXE)
C ,AA(JXE),BB(JXE),CC(JXE),DD(JXE),EE(JXE),FF(JXE),FEES(JXE)
DIMENSION DXEP(JXE)
EQUIVALENCE(DXEP(1),DXEM(2))
LCM (MGETG)
COMMON/MGETG/PLEG((0,MMX),IYA),DPLEG((0,MMX),IYA),
C ,DDPLEG((0,MMX),IYA),UM(IYA),DMU(IYA),USIN(IYA),
C ,EMQF(JXA,(0,MX)),ENQF(JXA,(0,MX)),ERQF(JXA,(0,MX)),
C ,EQGF(JXA,(0,MX)),COG((0,MX),B),GLOX(IYA,IYA+1,JXA),GCOY(IYAP1,JXA),
C ,GCOXX(IYAP1,JXA),GCOXY(IYAP1,JXA),GCOYY(IYAP1,JXA),
C ,XM(JXA,(MXP3M,MXP4))
C ,MMAX
COMMON/MPREION/AXQ(100),AEP(100),AGE(100),AF(100)
COMMON/MCHARGEX/SIGVR2(NSOR),SIGVR3,CXIOND(NSOR),
C ,GST2(IYAP1,JXA),CURROIS(NSOR),RSIS(NSOR),DEN01(NSOR)
C ,SIGXV(IYA,JXA,NSOR),SIGIV(IYA,JXA,NSOR)
COMMON/MTDMAGF/BTAU,BE,DBDT,DCP,Z1(IYA),Z2(IYA),
C ,DB1P(IYAP1,JXA),DB2P(IYAP1,JXA)

```

```

C ,REBLT0,BDC,BPK,BUP
ENDCLICHE
USE COMM
DIMENSION ND(5) $ ND(1)=0 $ ND(2)=1HF $ ND(3)=6 $ ND(4)=0
ND(5)=1 $ CALL HSP(ND)
CALL ASSIGN(2,0,4RDATA,0)
KNSET=1
DO 1000 KN=1,KNSET
KNPROBS=KN
WRITE (3,1001) KN
1001 FORMAT(1H1,18HTHIS IS RUN NUMBER,15)
KMAXE=0 $ KGNANDE=0 $ KAMBI=0
C TAPE 2 CONTAINS INPUT ON DATA CARDS.
C SET UP VARIOUS CONSTANTS.
IY=IYA $ JX=JXA $ IYM1=IY-1
JXM1=JX-1 $ JXM2=JX-2 $ IYP1=IY+1 $ IYM2=IY-2
JXEM1=JXE-1 $ JXEM2=JXE-2
PI = 3.1415926536
TWOPI = 2. *PI
FOURPI = 4. *PI
PI02=.5*PI $ TWOOP1=2./PI
CHARGE=4.803E-10
E2=CHARGE*CHARGE
E4=E2*E2
ECON=1.502E-9
CURCON=1.E-10*ECON
AMU=1.67165E-24
EMASS=9.1066E-28/AMU
TOTP=0.
TIME=0.
CXLOSS=0. $ CXLOST=0.
KGETGFK=0
DO 215 I=1,IYP1
DO 215 J=1,JX
G(I,J)=0.
215 GCO(I,J)=0.
CALL PREIONI
C READ IN DATA FROM DATA CARDS - THEN OUTPUT IT.
10 FORMAT(1015)
20 FORMAT(5E16.8)
READ (2,20) XMAX
READ (2,10) KWAIT,KSET,MIMAX
READ (2,20) STOP,DTR,TINJSTOP
READ (2,20) POTENT,DENST
READ (2,20) PMASS,PNUMB
READ (2,20) EES,DENSNO,EELECT
READ (2,20) PCURROE,DPCURROE,PCURROSE
READ (2,20) BRATIO
READ (2,20) (CXRXB(K),K=1,4)
READ (2,10) NRSTRT
READ (2,20) FEMIN,XSOE
DO 511 IC=1,NSOR
READ (2,20) ESOR1(IC),XSOR1(IC),YSOR1(IC),USOR1(IC)
READ (2,20) PCUR1(IC),DPCUR1(IC),PCURROS1(IC)
READ (2,20) DENO1(IC)
511 CONTINUE
READ (2,10) MMAX,KSORC
READ (2,20) PLTANG,BTAU
READ (2,20) REBLT0,BDC,BPK,BUP
WRITE (3,219) MMAX,KSORC

```



```

219 FORMAT(5HMAX=,15,5X,6HKSORC=,15)
C   KSORC=0 FOR MU GAUSSIAN,1 FOR N MODE,2 FOR THETA GAUSSIAN
   WRITE (3,33) XMAX
33  FORMAT(5HXMAX=,E16.6)
   WRITE (3,301) KWAIT,KSET,MIMAX,POTENT,DENST
301  FORMAT(6HKAIT=,15,5X,5HKSET=,15,5X,6HMIMAX=,15,5X,10HAMB1 POT =,
C   E16.6,5X,6HDENST=,E16.6)
   WRITE (3,302) STOP,DTR,PMASS,BRATIO
302  FORMAT(12HTIME LIMIT =,E16.6,2X,7HSECONDS,5X,4HDTR=,E16.6,5X,
C   10HION MASS =,F10.3,5H AMU,5X,14HMIRROR RATIO =,F10.3)
   WRITE (3,308) PNUMB
308  FORMAT(19HION ATOMIC NUMBER =,E16.6)
   WRITE (3,303) EES,DENSNO,EELECT
303  FORMAT(16HELEC SOURCE EN =,E16.6,5H KEV,5X,7HDENSNO=,E16.6,
C   7HPART/CC,5X,9HELEC EN =,E16.6,5H KEV)
   WRITE (3,351) TINJSTOP
351  FORMAT(9HTINJSTOP=,E16.6,8H SECONDS)
   WRITE (3,357) (CXRXB(K),K=1,4)
357  FORMAT(6HCXRXB=,4E16.6)
   WRITE (3,501) FEMIN,XSOE
501  FORMAT(6HFEMIN=,E16.6,5X,5HXSOE=,E16.6)
   WRITE (3,721) PLTANG,BTAU
721  FORMAT(7HPLTANG=,E16.6,5X,5HBTAU=,E16.6)
   DO 512 IC=1,NSOR
   WRITE (3,411) IC,ESOR(IC),XSO(IC),YSO(IC),USO(IC),
C   PCUR(IC),DPCUR(IC),PCURROS(IC),DENO(IC)
411  FORMAT(3HIC=,15,5X,5HESOR=,E16.6,5X,4HXSO=,E16.6,5X,4HYSO=,
C   E16.6,5X,4HUSO=,E16.6,/,5HPCUR=,E16.6,5X,6HDPCUR=,E16.6,5X,
C   8HPCURROS=,E16.6,5X,5HDENO=,E16.6)
512  CONTINUE
   WRITE (3,517) REBLT0,BDC,BPK,BUP
517  FORMAT(7HREBLT0=,E16.6,5X,4HBDC=,E16.6,5X,4HBPK=,E16.6,5X,
C   4HBUP=,E16.6)
   CALL WHY
C   SUBROUTINE WHY SETS UP MESH AND INTEGRATION CONSTANTS
   CALL PREION2
   FACTORA=1./COSF(Y(1))*2
   DO 131 I=1,IYMI
   FACTORB=1.-FACTORA*COSF(Y(1))*2
   DO 132 J=1,JX
   G(I,J)=G(IY,J)*FACTORB
132  CONTINUE
131  CONTINUE
   DO 133 J=1,JX
133  G(IY1,J)=G(IYMI,J)
   CALL GNAUDE
C   SUBROUTINE GNAUDE COMPUTES ION DENSITY AND ION ENERGY (NORMALIZED).
C   RENORMALIZE ION DISTRIBUTION.
   IF (KGNAUDE.EQ.1) GO TO 1002
   DO 1400 I=1,IY1
   DO 1400 J=1,JX
1400  G(I,J)=G(I,J)/GNORM
   GNORM=1.
   REIONE=EIONS*ENERGY
   WRITE (3,34) REIONE
34  FORMAT(12HION ENERGY =,E16.6,5H KEV)
   EMOIM = EMASS/PMASS
   VNORM = SQRTF(2.*EIONS*ECON/(PMASS*AMU))
   GAMO=FOURPI*E4*PNUMB**4/(PMASS*AMU)**2
   DO 401 J=1,JX

```

```

401 VEL(J) = X(J)*VNORM
WRITE (3,32)
32 FORMAT(/,3X,16HJ      NORMED VEL,8X,8HVELOCITY,/)
WRITE (3,31) (J,X(J),VEL(J),J=1,JX)
31 FORMAT(15,F10.5,E18.8)
C COMPUTE SHAPE FUNCTION FOR ION SOURCE.
DO 773 IC=1,NSOR
XSQR=SQRTF(ESORJ(IC)/EIONS)
XSQ=XSQ1(IC)
DO 320 J=1,JX
ARGX=XSQ*(X(J)-XSQR)**2
SHAPE(IY,J,IC)=EXPF(-ARGX)
DO 320 I=1,IY
YSQ=YSQ1(IC) $ USQ=USQ1(IC)
YFACN=EXPF(-YSQ*(Y(I)-P102)**2)
YFACP=EXPF(-YSQ*(COSF(Y(I))-USQ)**2)
REFF=BRATIO/(1.+PNUMB*POTENT/EIONS)
UMR2=1.-1./REFF
UM2=COSF(Y(I))**2
ENUM=UMR2-UM2+(3.*UMR2-1.)*LOGF((1.-UM2)/(1.-UMR2))
DENOM=UMR2-(3.*UMR2-1.)*LOGF(1.-UMR2)
YFACNM=ENUM/DENOM
YFAC=YFACN
YFAC=YFACNM
YFAC=YFACP
IF (KSORC.EQ. 1) YFAC=YFACNM
IF (KSORC.EQ. 2) YFAC=YFACN
SHAPE(I,J,IC)=SHAPE(IY,J,IC)*YFAC
TEMPG(I)=YFAC
320 CONTINUE
WRITE (3,323)
323 FORMAT(/,5HPOINT,5X,6HCOSSINE,9X,12HSOURCE SHAPE)
DO 321 I=1,IY
S=COSF(Y(I)) $ T=TEMPG(I)
WRITE (3,322) I,S,T
322 FORMAT(15,5X,F10.3,5X,E16.6)
321 CONTINUE
C COMPUTE ENERGY OF ION SOURCE.
EO = 0.
CI = 0.
DO 653 I = 1, IY
DO 653 J = 1, JX
EO=EO+SHAPE(I,J,IC)*XSQ(J)*SPHERI(I,J)
653 CI=CI+SHAPE(I,J,IC)*SPHERI(I,J)
EO = EO*EIONS/CI
EO1(IC)=EO $ CI1(IC)=CI
773 CONTINUE
TNORM = VNORM**3/(DENSNO*GAMO)
DT=DTR/TNORM
RHOP = 2./DT
VC2=POTENT/EIONS
VC=SQRTF(VC2)
DO 774 IC=1,NSOR
CURRO1(IC)=PCURI(IC)*CURCON $ DCURRI(IC)=DPCUR1(IC)*CURCON
RS1(IC)=(PCURI(IC)*TNORM)/(CI1(IC)*DENSNO)
DC1(IC)=DCURRI(IC)*DENSNO/CURRO1(IC)
DCISEI(IC)=TNORM*PCURROSI(IC)/(CI1(IC)*SQRTF(DENSNO))
CURROIS(IC)=CURRO1(IC) $ RSIS(IC)=RS1(IC)
774 CONTINUE
CALL VREL

```

```

ENERGE=E/ELECT/EIONS
DENSE=GNORM*PNUMB
CALL PREEL
C SUBROUTINE PREEL SETS UP CONSTANTS INVOLVING ELECTRON EQUATION.
CALL GAMMAIN(0)
CALL IONISOT
CALL GELECT
C SUBROUTINE GELECT SETS UP ELECTRON QUANTITIES USED IN ION EQUATION.
DO 775 I=1,IY $ DO 775 J=1,JX
775 SOURCE(I,J)=0.
CURR=0.
DO 776 IC=1,NSOR
SOI(IC)=RSI(IC)*(1.+DCI(IC)*GNORM)+DCISEI(IC)*SQRTF(GNORM)
DO 113 I = 1, IY
DO 113 J = 1,JX
113 SOURCE(I,J)=SOURCE(I,J)+SOI(IC)*SHAPE(I,J,IC)
CURR=CURR+SOI(IC)*PCURI(IC)/RSI(IC)
776 CONTINUE
WRITE (3,324)
324 FORMAT(/,6HSOURCE,/)
DO 325 J=1,JX
WRITE (3,326) (SOURCE(I,J),I=1,IY)
326 FORMAT(6E16.6)
325 CONTINUE
DO 777 IC=1,NSOR
WRITE (3,305) IC,PCURI(IC),DPCURI(IC),PCURROSI(IC),RSI(IC),
C DCI(IC),DCISEI(IC),SOI(IC),EOI(IC)
305 FORMAT(3HIC=,I5,5X,5HPCUR=,E16.6,5X,6HDPCUR=,E16.6,5X,8HPCURRO=,
C E16.6,/,3HRS=,E16.6,5X,3HDC=,E16.6,5X,6HDCISE=,E16.6,/,3HSO=,
C E16.6,5X,3HEO=,E16.6)
777 CONTINUE
WRITE (3,306) VC,ENERGY,GNORM,VNORM,TNORM
306 FORMAT(3HVC=,E16.6,5X,17HNORM ION ENERGY =,E16.6,5X,
C 18HNORM ION DENSITY =,E16.6,/,6HVNORM=,E16.6,5X,6HTNORM=,E16.6)
WRITE (3,307) PCURROE,DPCURROE,PCURROSE
307 FORMAT(18HELEC PART SOURCE =,3E16.6)
BTAU=BTAU/TNORM $ BE=1. $ DBDT=0.
DO 91 I=1,IY
Z1(I)=SINF(Y(I))*2 $ Z2(I)=.5*SINF(2.*Y(I))
91 CONTINUE
CALL CRTID(10HAM 2D IONS,1,1)
CALL FRAME
CALL OUTPUT
CALL GETGF
C SUBROUTINE GETGF COMPUTES THE GCO ARRAY.
DO 135 J=1,JX
WRITE (3,136)((Y(I),X(J),GCO(I,J)),I=1,IY)
136 FORMAT(6(2F5.2,E10.2),/)
135 CONTINUE
K = 0
KGCO=0
CALL TLOOP
C SUBROUTINE TLOOP TAKES CARE OF THE REGULAR CYCLING OF THE PROGRAM.
CALL OUTPUT
C SUBROUTINE OUTPUT PERFORMS STANDARD OUTPUT.
IF (KGNANDE .EQ. 1 .OR. KMAXE .EQ. 1 .OR. KAMBI .EQ. 1) GO TO 1002
CALL QHOLD2
CALL QFUTCH2
1002 CONTINUE
1000 CONTINUE

```

```

CALL QUIT(1)
END
SUBROUTINE GETGF
USE COMM
IF (KGETGFK .EQ. 1) GO TO 10
DO 20 M=MXP3M,MXP4
DO 1 J=2,JX
1 XM(J,M)=X(J)**M
20 XM(1,M)=0.
DO 3 I=1,IY
3 UM(I)=COSF(Y(I))
DO 4 I=1,IY
PLEG(0,I)=1. $ DPLEG(0,I)=0.
IF (MX .EQ. 0) GO TO 4
PLEG(I,I)=UM(I) $ DPLEG(I,I)=1.
DO 5 M=2,MMX
PLEG(M,I)={(2*M-1)*UM(I)*PLEG(M-1,I)-(M-1)*PLEG(M-2,I)}/FLOATF(M)
DPLEG(M,I)=(2*M-1)*PLEG(M-1,I)+DPLEG(M-2,I)
5 CONTINUE
4 CONTINUE
DO 6 I=1,IY
DO 6 M=0,MMX
DDPLEG(M,I)=(2.*UM(I)*DPLEG(M,I)-FLOATF(M*(M+1))*PLEG(M,I))/
C (1.-UM(I)**2)
6 CONTINUE
DO 7 MM=0,MX
M=2*MM
FAC=FOURPI/(2*M+1)
COG(MM,1)=FAC/FLOATF(2*M+3)
COG(MM,2)=FAC/FLOATF(2*M-1)
COG(MM,3)=FLOATF(M+2)*COG(MM,1)
COG(MM,4)=FLOATF(M+1)*COG(MM,1)
COG(MM,5)=FLOATF(M)*COG(MM,2)
COG(MM,6)=FLOATF(M-1)*COG(MM,2)
COG(MM,7)=FLOATF(M+1)*COG(MM,3)
COG(MM,8)=FLOATF(M-1)*COG(MM,5)
7 CONTINUE
DO 9 I=2,IYM1
9 DMU(I)=.5*(UM(I+1)-UM(I-1))
DMU(1)=.5*(UM(2)-UM(1)) $ DMU(IY)=.5*(UM(IY)-UM(IYM1))
DO 18 I=1,IY
18 USIN(I)=SINF(Y(I))
KGETGFK=1
10 CONTINUE
DO 14 MM=0,MX
M=2*MM
DO 12 J=1,JX
S=0. $ DO 11 I=1,IY
11 S=S+G(I,J)*PLEG(M,I)*DMU(I)
12 TEMPG(J)=-FLOATF(2*M+1)*S
IX1=1-M $ IX2=2+M $ IX3=3-M $ IX4=4+M
IX5=3-IX1 $ IX6=3-IX2 $ IX7=3-IX3 $ IX8=3-IX4
SM=0. $ SN=0. $ SR=0. $ SE=0.
EMQF(JX,MM)=SM $ ENQF(1,MM)=SN $ ERQF(JX,MM)=SR $ EEQF(1,MM)=SE
DO 13 J=2,JX
JJ=JX+1-J
TM=.5*DXP5(JJ)*(XM(JJ,IX1)*TEMPG(JJ)+XM(JJ+1,IX1)*TEMPG(JJ+1))
TN=.5*DXM5(J)*(XM(J-1,IX2)*TEMPG(J-1)+XM(J,IX2)*TEMPG(J))
TR=.5*DXP5(JJ)*(XM(JJ,IX3)*TEMPG(JJ)+XM(JJ+1,IX3)*TEMPG(JJ+1))
TE=.5*DXM5(J)*(XM(J-1,IX4)*TEMPG(J-1)+XM(J,IX4)*TEMPG(J))

```

```

SM=SM+TM $ SN=SN+TN $ SR=SR+TR $ SE=SE+TE
EMQF(JJ,MM)=SM $ ENQF(J,MM)=SN $ ERQF(JJ,MM)=SR $ EEQF(J,MM)=SE
13 CONTINUE
DO 15 J=1,JX
  EMQF(J,MM)=EMQF(J,MM)*XM(J,IX5)
  ENQF(J,MM)=ENQF(J,MM)*XM(J,IX6)
  ERQF(J,MM)=ERQF(J,MM)*XM(J,IX7)
  EEQF(J,MM)=EEQF(J,MM)*XM(J,IX8)
15 CONTINUE
14 CONTINUE
DO 16 J=1,JX $ DO 16 I=1,IY
  S=0. $ SX=0. $ SY=0. $ SXX=0. $ SXY=0. $ SYY=0.
  DO 17 M=0,MX
    IF (M.GT. MMAX) GO TO 17
    MM=2*M
    T=COG(M,1)*(EEQF(J,M)+EMQF(J,M))-COG(M,2)*(ENQF(J,M)+ERQF(J,M))
    TX=X21(J)*X(J)*(COG(M,3)*EMQF(J,M)-COG(M,4)*EEQF(J,M)-
    C COG(M,5)*ERQF(J,M)+COG(M,6)*ENQF(J,M))
    TXX=X21(J)*(COG(M,7)*(EEQF(J,M)+EMQF(J,M))-
    C COG(M,8)*(ENQF(J,M)+ERQF(J,M)))
    S=S+T*PLEG(MM,1) $ SX=SX+TX*PLEG(MM,1)
    SXX=SXX+TXX*PLEG(MM,1) $ SY=SY+T*DPLEG(MM,1)
    SXY=SXY+TX*DPLEG(MM,1) $ SYY=SYY+T*DDPLEG(MM,1)
17 CONTINUE
  GCO(1,J)=S
  GCOX(1,J)=SX $ GCOY(1,J)=-USIN(1)*SY
  GCOXX(1,J)=SXX $ GCOXY(1,J)=-USIN(1)*SXY
  GCOYY(1,J)=-UM(1)*SY+USIN(1)*2*SYY
16 CONTINUE
DO 19 J=1,JX
  GCO(IYP1,J)=GCO(IYM1,J)
  GCOX(IYP1,J)=GCOX(IYM1,J) $ GCOY(IYP1,J)=-GCOY(IYM1,J)
  GCOXX(IYP1,J)=GCOXX(IYM1,J) $ GCOXY(IYP1,J)=-GCOXY(IYM1,J)
  GCOYY(IYP1,J)=GCOYY(IYM1,J)
19 CONTINUE
WRITE (3,2)
2 FORMAT(14HGCCO RECOMPUTED)
GN=GAM11/GNORM
DO 21 I=1,IYP1 $ DO 21 J=1,JX
  GCO(I,J)=GN*GCO(I,J)
  GCOX(I,J)=GN*GCOX(I,J) $ GCOY(I,J)=GN*GCOY(I,J)
  GCOXX(I,J)=GN*GCOXX(I,J) $ GCOXY(I,J)=GN*GCOXY(I,J)
  GCOYY(I,J)=GN*GCOYY(I,J)
21 CONTINUE
KGCO=0 $ CALL COEF
RETURN
END
SUBROUTINE TLOOP
USE COMM
TINSTOP=TINJSTOP/TNORM
N=0
NCH=0 $ NCH=1
NCH=0 $ NCH=1
READ (2,1) NSE(1,NSET2,NSET3,NSET4,NSET5,NSET6,NSET7)
11 FORMAT(7I5)
READ (2,12) DT1,DT2,DT3,DT4,DT5,DT6,DT7
12 FORMAT(4E16.6,/,3E16.6)
204 CONTINUE
N=N+1
IF (N.EQ. NRSTRT) GO TO 999

```

```

      IF (SENSE SWITCH 1) 998,999
998 CALL RESTART
999 CONTINUE
C USE THIS AREA TO ALTER DT IN FUTURE
      IF (N .EQ. NSET1) DTR=DT1
      IF (N .EQ. NSET2) DTR=DT2
      IF (N .EQ. NSET3) DTR=DT3
      IF (N .EQ. NSET4) DTR=DT4
      IF (N .EQ. NSET5) DTR=DT5
      IF (N .EQ. NSET6) DTR=DT6
      IF (N .EQ. NSET7) DTR=DT7
      DT=DTR/TNORM
      TCH1=10.01*DT $ TCH2=50.01*DT
      IF (N .NE. NSET1 .AND. N .NE. NSET2 .AND. N .NE. NSET3 .AND.
C N .NE. NSET4 .AND. N .NE. NSET5 .AND. N .NE. NSET6 .AND.
C N .NE. NSET7) GO TO 601
      TCH1=TIME+TCH1 $ TCH2=TIME+TCH2
601 CONTINUE
      RHOP=2./DT
C RESET KWAIT AND KSET AS TIME GOES ON.
      KWAIT=20 $ IF (TIME .LT. TCH2) KWAIT=5
      IF (TIME .LT. TCH1) KWAIT=1
      K=K+1
      KGCO=KGCO+1
C RECOMPUTE GCO ARRAY IF KGCO EQUALS KWAIT.
      IF (KGCO .LT. KWAIT) GO TO 572
      CALL GETGF
      CALL CSIGVR
      GO TO 571
572 CALL COEF
571 CONTINUE
      REALT=TIME*TNORM $ IF (REALT .GT. STOP) GO TO 1000
      WRITE (3,2) TIME,REALT
      2 FORMAT(5H TIME=,E12.3,5X,10H REAL TIME=,E12.3)
      IF (ABSF(TINSTOP-TIME) .GT. .75*DT) GO TO 1001
      DO 779 IC=1,NSOR
      RSI(IC)=1.E-90 $ DCI(IC)=0. $ DCISE1(IC)=0.
      PCURI(IC)=1.E-90 $ DPCURI(IC)=0. $ PCURROSI(IC)=0.
      CURROI(IC)=1.E-90 $ DCURRI(IC)=0.
      CURROIS(IC)=CURROI(IC) $ RSIS(IC)=RSI(IC) $ DENDI(IC)=0.
779 CONTINUE
      COE=0. $ DCESE=0.
      PCURROSE=0. $ PCURROE=0.
      CXRXB(4)=0.
      TCH1=TIME+DT+TCH1 $ TCH2=TIME+DT+TCH2
1001 CONTINUE
      DENSITY=GNORM*DENSNO $ ENERGR=ENERGY*E10NS
      RDE=DENSE*DENSNO $ REE=ENERGE*E10NS
      IF (NCH .EQ. NONCH) NCH=0
      IF (N-NCHEC*(N/NCHEC) .GT. 0) GO TO 602
      NCH=NCH+1
      PTIME(NCH)=REALT $ PDENS(NCH,1)=DENSITY $ PDENS(NCH,2)=RDE
      PENGY(NCH,1)=ENERGR $ PENGY(NCH,2)=REE
602 CONTINUE
      WRITE (3,3) DENSITY,RDE,ENERGR,REE,POTENT
      3 FORMAT(5X,8H1ON DEN=,E16.6,5X,9H1ELEC DEN=,E16.6,/,5X,7H1ON EN=,
C E16.6,5X,8H1ELEC EN=,E16.6,5X,7HPOTENT=,E16.6)
      TIME=TIME+DT
      CALL CHARGE
      SIGVR3=2.152E-8*(E10NS*ENERGE)**(-.2465)

```

```

DO 780 I=1,IY $ DO 780 J=1,JX
780 SOURCE(I,J)=0.
CURR=0.
ST=0.
DO 781 IC=1,NSOR
DCURRI(IC)=CURCON*DEN01(IC)*(VNORM*SIGVR2(IC)+SIGVR3)
CURROI(IC)=CURROIS(IC)-CURCON*CXIOND(IC)*DENSNO/DTR
PCURI(IC)=CURROI(IC)/CURCON $ DPCURI(IC)=DCURRI(IC)/CURCON
WRITE (3,713) IC,CURROI(IC),DCURRI(IC)
713 FORMAT(3HIC=,I5,5X,6HCURRO=,E16.6,5X,6HDCURR=,E16.6)
RSI(IC)=RSIS(IC)*CURROI(IC)/CURROIS(IC)
DCI(IC)=DCURRI(IC)*DENSNO/CURROI(IC)
SOI(IC)=RSI(IC)*(1.+DCI(IC)*GNORM)+DCISEI(IC)*SQRTF(GNORM)
DO 700 I=2,IY
DO 700 J=1,JX
700 SOURCE(I,J)=SOURCE(I,J)+SOI(IC)*SHAPE(I,J,IC)
TOTP=TOTP+DT*SOI(IC)*CII(IC)
CURR=CURR+SOI(IC)*PCURI(IC)/RSI(IC)
ST=ST+DPCURI(IC)*PNUMB
781 CONTINUE
DPCURROE=ST $ DCOE=DPCURROE*DENSNO
CALL AMBI
C SUBROUTINE AMBI COMPUTES AND ADJUSTS AMBIPOLAR POTENTIAL, AND CALLS SUBROUTINE
C TIME MAXE, WHICH ADVANCES ELECTRON EQUATION.
IF (KAMBI.EQ.1.OR.KGNANDE.EQ.1.OR.KMAXE.EQ.1) RETURN
CXLOSS=0. $ DO 41 I=1,IY $ DO 41 J=1,JX
CXLOSS=CXLOSS+HCXR(J)*G(I,J)*SPHERI(I,J)
41 CONTINUE
CXLOSS=DENSNO*CXLOSS
IF (TIME.LE.TINSTOP) GO TO 42
CXLOST=CXLOST+CXLOSS*DT*TNORM
42 CONTINUE
IF (K.LT.KSET) GO TO 204
C OUTPUT IF K EQUALS KSET.
CALL OUTPUT $ K=0
GO TO 204
1000 RETURN
END
SUBROUTINE WHY
USE COMM
DEGTOR=180./PI
BRATIOI=1./BRATIO
THETAL=ACOSF(SQRTF(1.-BRATIOI))
DEGREES=THETAL*DEGTOR
WRITE (3,60) BRATIO,THETAL,DEGREES
60 FORMAT(14HMIRROR RATIO =,E12.3,/,16HLOSSCONE ANGLE =,F11.5,
C 8H RADIANS,3H =,F11.5,8H DEGREES,/,1H1,10X,7HRADIANS,7X,
C 7HDEGREES)
C COMPUTE Y MESH.
Y(1)=THETAL
DELY=(PI02-Y(1))/FLOATF(IYMI)
DO 301 I=2,IY
301 Y(I)=Y(I-1)+DELY
Y(IYI)=Y(IY)+Y(IY)-Y(IYMI)
C COMPUTE Y-MESH QUANTITIES.
DO 400 I=1,IYI
400 DEG(I)=Y(I)*DEGTOR
WRITE (3,20) (I,Y(I),DEG(I),I=1,IYI)
20 FORMAT (15,2F14.9)
DO 201 I=2,IY

```

```

      CTNY(1)=1./TANF(Y(1))
201 CTNY2(1)=CTNY(1)*CTNY(1)
      DO 203 I = 2, IY
        DYP5(1)=Y(I+1)-Y(1)
        DYM5(1)=Y(1)-Y(I-1)
        DY(1)=.5*(Y(I+1)-Y(I-1))
        DYI(1)=1./DY(1)
        SINN(1)=SINF(Y(1))
203 CONTINUE
      DYP5(1)=DYM5(2)
      SINN(1)=SINF(Y(1))
C COMPUTE X MESH
      X(1)=0. $ H=XMAX/FLOAT(JXM1)
      X(2)=H $ RAM=1.
      DO 3 J=3,JX
        3 X(J)=X(J-1)+RAM*(X(J-1)-X(J-2))
C COMPUTE X-MESH QUANTITIES.
      DO 202 J=2,JXM1
        DXP5(J)=X(J+1)-X(J)
        DXM5(J)=X(J)-X(J-1)
202 DX(J)=.5*(X(J+1)-X(J-1))
        DX(1)=.5*(X(2)-X(1)) $ DX(JX)=.5*(X(JX)-X(JXM1))
        DXP5(1)=DXM5(2) $ DXM5(JX)=DXP5(JXM1)
        DO 200 J=2,JXM1
          DXI(J)=1./DX(J)
          X2I(J)=1./(X(J)*X(J))
          X3I(J)=X2I(J)/X(J)
200 X4I(J)=X3I(J)/X(J)
        DO 102 J=1,JX
          102 XSQ(J)=X(J)*X(J)
C SET UP X-INTEGRATION CONSTANTS.
      F2 = DXP5(2)
      B2=DXM5(2)
      CINT(1)=(2.*B2**3-F2**3+3.*B2**2*F2)/(6.*B2*(F2+B2))
      DO 104 J=2,JXM1,2
        F0=DXP5(J)
        B0=DXM5(J)
104 CINT(J)=(F0**3+B0**3+3.*F0*B0*(F0+B0))/(6.*F0*B0)
        DO 105 J=3,JXM2,2
          F1=DXP5(J-1)
          B1=DXM5(J-1)
          F2=DXP5(J+1)
          B2=DXM5(J+1)
105 CINT(J)=(2.*F1**3-B1**3+3.*F1**2*B1)/(6.*F1*(F1+B1))
          1 + (2.*B2**3-F2**3+3.*B2**2*F2)/(6.*B2*(F2+B2))
          F1=DXP5(JXM1)
          B1=DXM5(JXM1)
          CINT(JX)=(2.*F1**3-B1**3+3.*F1**2*B1)/(6.*F1*(F1+B1))
C SET UP Y-INTEGRATION CONSTANTS.
      F2 = DYP5(2)
      B2 = DYM5(2)
      CYNT(1)=(2.*B2**3-F2**3+3.*B2**2*F2)/(6.*B2*(F2+B2))
      DO 204 I = 2, IY, 2
        F0 = DYP5(1)
        B0 = DYM5(1)
204 CYNT(I)=(F0**3+B0**3+3.*F0*B0*(F0+B0))/(6.*F0*B0)
        DO 205 I = 3,IYM1, 2
          F1 = DYP5(I-1)
          B1 = DYM5(I-1)
          F2 = DYP5(I+1)

```



```

      B2 = DYM5(I+1)
205 CYNT(1)=(2.*F1**3-B1**3+3.*F1**2*B1)/(6.*F1*(F1+B1))
      1+(2.*B2**3-F2**3+3.*B2**2*F2)/(6.*B2*(F2+B2))
      DO 302 I = 1, IYM1
302 CYNT(1) = 2. * CYNT(1)
C SET UP TWO-DIMENSIONAL INTEGRATION CONSTANTS.
      DO 103 I = 1, IY
      DO 103 J=1,JX
103 SPHER1(I,J) = CYNT(1)*SINF(Y(1)) *CINT(J)*XSQ(J)*TWOPI
      RETURN
      END
      SUBROUTINE CHARGEX
      USE COMM
      GNST=GNORM
      DO 102 I=1,IYP1 $ DO 102 J=1,JX
      GST(I,J)=G(I,J)
102 GST2(I,J)=G(I,J)
      DO 101 IC=1,NSOR
      CON=DENO1(IC)*VNORM*TNORM
      AA=4.152E-16 $ BB=9.309E-9
      DO 10 J=1,JX
      DO 11 I=1,IY
      G(I,J)=GST2(I,J)
      G(I,J)=G(I,J)*(1.-CON*DT*SIGXV(I,J,IC))
11 CONTINUE
      G(IYP1,J)=G(IYM1,J)
10 CONTINUE
      CALL GNADE
      CXIOND(IC)=GNORM-GNST
      DO 103 I=1,IYP1 $ DO 103 J=1,JX
      GST(I,J)=GST(I,J)+G(I,J)-GST2(I,J)
103 CONTINUE
101 CONTINUE
      DO 1 I=1,IYP1 $ DO 1 J=1,JX
      1 G(I,J)=GST(I,J)
      GNORM=GNST
      RETURN
      END
      SUBROUTINE VREL
      USE COMM
      PARAMETER(NPH1=31)
      DIMENSION PHI(NPH1)
      PHI(1)=0. $ DPH1=PI02/FLOATF(NPH1-1)
      DO 1 L=2,NPH1
      1 PHI(L)=PHI(L-1)+DPHI
      DO 101 IC=1,NSOR
      WRITE (3,13) IC
13 FORMAT(/,20HREL VELOC ARRAYS - SOURCE NO,15,/)
      V0N2=ESOR1(IC)/EIONS $ V0N=SQRTF(V0N2)
      DO 11 J=1,JX $ DO 10 I=1,IY
      CON1=2.*V0N*X(J)*SINF(Y(1))
      CON=V0N2+XSQ(J)+CON1
      EK2=2.*CON1/CON
      CONRT=SQRTF(CON)
      S=0. $ T=0.
      DO 12 L=1,NPH1
      CON2=1.-EK2*(SINF(PHI(L)))**2
      CON2RT=SQRTF(CON2)
      VR=CONRT*CON2RT
      VRNP=.5*(VR*VNORM)**2

```

```

SIGS=SIGX(VRNP) $ SIGT=SIGI(VRNP)
GRANDS=SIGS*CON2RT $ GRANDT=SIGT*CON2RT
FACT=DPHI $ IF (L.EQ.1.OR.L.EQ.NPHI) FACT=.5*DPHI
S=S+FACT*GRANDS $ T=T+FACT*GRANDT
12 CONTINUE
SIGXV(I,J,IC)=CONRT*S/PI02
SIGIV(I,J,IC)=CONRT*T/PI02
10 CONTINUE
WRITE (3,14) J,((I,SIGXV(I,J,IC),SIGIV(I,J,IC)),I=1,IY)
14 FORMAT(/,2HJ=,15,/, (2H1=,15,5X,6HSIGXV=,E16.6,5X,6HSIGIV=,E16.6))
11 CONTINUE
101 CONTINUE
RETURN
END
FUNCTION SIGX(V)
E=1.043948E-12*V $ TLOG=.434294*LOGF(E)
PWR=E**3.3
SIGX=(6.937E-15*(1-.155*TLOG)**2)/(1.+1.112E-15*PWR)
RETURN
END
FUNCTION SIGI(V)
E=1.043948E-12*V $ PAREN=.434294*LOGF(E)
TLS=-.8712*PAREN**2+8.156*PAREN-34.833
SIGI=EXP(2.30259*TLS)
RETURN
END
SUBROUTINE CSIGVR
USE COMM
DO 3 IC=1,NSOR
S1=0. $ S2=0. $ DO 1 I=1,IY $ DO 1 J=1,JX
S3=SIGIV(I,J,IC)
S1=S1+S3*G(I,J)*SPHERI(I,J) $ S2=S2*G(I,J)*SPHERI(I,J)
1 CONTINUE
SIGVR2(IC)=S1/S2
3 CONTINUE
RETURN
END
SUBROUTINE PREION1
USE COMM
READ (2,22) JPM,(AEP(I),AGE(I),I=1,JPM)
22 FORMAT(12,/, (F6.3,F7.3))
WRITE (3,27) JPM,(AEP(I),AGE(I),I=1,JPM)
27 FORMAT(1X,12,/, (2F7.3))
DO 10 I=1,JPM
10 AF(I)=AGE(I)*AEP(I)
CALL INTGRL(JPM,AF,AEP,ANS1) $ CALL INTGRL(JPM,AGE,AEP,ANS2)
EIONS=ANS1/ANS2
DO 15 I=1,JPM
15 AXQ(I)=SQRTF(AEP(I)/EIONS)
AXQ(JPM+1)=2.*AXQ(JPM)-AXQ(JPM-1)
AGE(JPM+1)=0. $ DO 14 I=2,JPM+1
14 AGE(I)=AGE(I)/AXQ(I)
RETURN
ENTRY PREION2
G(IY,1)=0.
DO 120 J=2,JXM1
DO 121 JP=1,JPM
IF (X(J).LT.AXQ(JP).OR.X(J).GE.AXQ(JP+1)) GO TO 121
JLO=JP $ JHI=JP+1
GO TO 120

```

```

121 CONTINUE
120 G(IY,J)=AGE(JLO)+(AGE(JHI)-AGE(JLO))*(X(J)-AXQ(JLO))/
  C (AXQ(JHI)-AXQ(JLO))
  G(IY,JX)=0.
  RETURN
  END
  SUBROUTINE BCHNG
  USE COMM
  CALL FIELD
  DCP=DBDT/BE
  DO 11 J=1,JX $ DO 1 I=1,IY
  DB1P(I,J)=-.5*X(J)*XSQ(J)*Z1(I)*DCP
  DB2P(I,J)=-.5*XSQ(J)*Z2(I)*SINN(I)*DCP
1 CONTINUE
  DB1P(IYP1,J)=DB1P(IYM1,J) $ DB2P(IYP1,J)=-DB2P(IYM1,J)
11 CONTINUE
  WRITE (3,2) BE,DBDT,DCP
2 FORMAT(3HBE=,E16.6,5X,5HDBDT=,E16.6,5X,4HDCP=,E16.6)
  RETURN
  END
  SUBROUTINE FIELD
  USE COMM
  IF (BTAU.EQ. 0.) GO TO 1
  BE=EXP(-1.E+6*TIME/BTAU)
  DBDT=-1.E+6*BE/BTAU
  CR=BPK/BDC
  ARG0=PI02*(REBL0-45.)/BUP
  REALT=TNORM*TIME
  REBL=REALT*1.E+6
  REBLT=REBL+REBL0
  BREF=1./(1.+CR*SINF(ARG0))
  IF (REBL.GT. BUP+45.) GO TO 2
  ARG0=PI02*(REBLT-45.)/BJP
  BE=BREF*(1.+CR*SINF(ARG0))
  DBDT=1.E+6*BREF*TNORM*CR*PI02*COSF(ARG0)/BUP
  GO TO 3
2 CONTINUE
  ARG0=(BUP+45.-REBLT)/(TNORM*BTAU)
  BE=BREF*(1.+CR*EXP(ARG0))
  DBDT=-1.E+6*BREF*CR*EXP(ARG0)/BTAU
3 CONTINUE
  RETURN
1 BE=1. $ DBDT=0. $ RETURN
  END
  SUBROUTINE OUTPUT
  USE COMM
  REALT=TIME*TNORM
  WRITE (3,24) TIME,REALT
24 FORMAT(1H),5HTIME=,E16.6,5X,11HREAL TIME =,E16.6,9H SECONDS)
  WRITE (3,21) DENSNO,VNORM
21 FORMAT(/,52HNORMED DIST. FN. IS GIVEN. IT MUST BE MULTIPLIED BY ,
  1/,E16.9,33H/CC., THE DENSITY NORM TIMES (1./,E16.9,10HCM/SEC)**3,
  2/,45H THE CUBE OF THE INVERSE OF THE VELOCITY NORM )
  WRITE (3,22) (Y(I),I=1,IY)
22 FORMAT(/,45H THETA,/,
  1(12F9.4))
  WRITE (3,1)
1 FORMAT(1H),10H J VEL ,10X,12HDISTRIBUTION ,/)
  DO 230 J=1,JX
  WRITE (3,23) (J,X(J),(G(I,J),I=1,IY))

```

```

23 FORMAT(/,13,F7.3,/,,(6E12.3))
230 CONTINUE
WRITE (3,231) ((J,XE(J),FEE(J)),J=1,JXE)
231 FORMAT(1H1,18HELEC DIST FUNCTION,/,,(2HJ=,15,5X,3HXE=,E16.6,5X,
C 4HFEE=,E16.6))
DENSITY=GNORM*DENSNO $ ENERGR=ENERGY*EIONS
REPCUR=CURR $ RECURR=REPCUR*CURCON
RDE=DENSE*DENSNO $ REE=ENERGE*EIONS
RETOTP=TOTP*DENSNO
WRITE (3,31) DENSITY,ENERGR,RDE,REE
31 FORMAT(10H 1ON DEN =,E16.6,5X,12H1ON ENERGY =,E16.6,5H KEV,
C /,10HELEC DEN =,E16.6,4X,13HELEC ENERGY =,E16.6,5H KEV)
WRITE (3,32) RECURR,REPCUR,RETOTP,POTENT
32 FORMAT(17H1ON SOURCE CURR =,E16.6,12H AMP/CC/SEC,5H = ,
C E16.6,13H PART/CC/SEC,/,22HTOTAL NO OF PART INJ =,E16.6,5X,
C 9HAMB POT =,E16.6)
CALL GETCURE
CURRER=CURRE*CURCON
WRITE (3,33) CURRER,CURRE
33 FORMAT(18HELEC SOURCE CURR =,E16.6,12H AMP/CC/SEC,
C 5H = ,E16.6,13H PART/CC/SEC)
WRITE (3,34) CXLOSS,CXLOST
34 FORMAT(15HCX LOSS TERMS =,2E16.6)
C SET UP PLOTTING OUTPUT.
CALL SETCH(1.,32.,0,0,2,0,0)
WRITE (100,5) REALT,DENSITY,ENERGR
5 FORMAT(5HTIME=,E10.3,5H SEC.,3X,8HDENSITY=,E10.3,4H/CC.,6H1ON E=,
IF7.3,3HKEV )
CALL SETCH(1.,31.,0,0,2,0,0)
WRITE (100,7) RDE,REE
7 FORMAT(23X,8HDENSITY=,E10.3,4H/CC.,6HELE E=,F7.3,3HKEV)
CALL SETCH(1.,30.,0,0,2,0,0)
WRITE (100,8) POTENT
8 FORMAT(45X,6HAM PT=,F7.3,3HKEV)
CALL SETCH(17.,1.,1,0,2,0)
WRITE (100,6)
6 FORMAT(16H1ON DISTRIBUTION)
PTANG=PLTANG
DO 41 J=1,JX
TEMPG(J)=X(J) $ DO 41 I=1,IY
41 GST(I,J)=G(I,J)
IF (PLTANG .LE. 180.) GO TO 42
PTANG=PLTANG-180.
DO 43 J=1,JX
JJ=JX+1-J $ TEMPG(J)=X(JJ) $ DO 43 I=1,IY
43 GST(I,J)=G(I,JJ)
42 CONTINUE
CALL PLOT3D(PTANG,TEMPG,Y,GST,1.,1.,1.,JX,IY,IYP1)
CALL CARTMM(JXE,DMIN,DMAX,FEE,1)
CALL MAPS(0.,XEMAX,0.,DMAX,.2,.999,.2,.999)
CALL SETCH(17.,3.,1,0,1,0)
WRITE (100,61)
61 FORMAT(30HELECTRON DISTRIBUTION FUNCTION)
CALL SETCH(17.,1.,1,0,1,0)
WRITE (100,62) REALT
62 FORMAT(5HTIME=,E16.6,2X,7HSECONDS)
CALL TRACE(XE,FEE,JXE)
CALL FRAME
IF (TIME .EQ. 0.) RETURN
CALL CARTMM(NCH,DMIN,DMAX,PDENS(1,1),1)

```

```

DMAX=1.1*DMAX
XMINP=PTIME(1) $ XMAXP=PTIME(NCH)
CALL MAPSSL(XMINP,XMAXP,DMIN,DMAX,.2,.999,.65,.999)
CALL SETCH(1.,45.,1,0,1,1)
WRITE (100,51)
51 FORMAT(11HION DENSITY)
CALL TRACE(PTIME,PDENS(1,1),NCH)
CALL CARTMM(NCH,DMIN,DMAX,PENGY(1,1),1)
DMAX=1.1*DMAX
CALL MAPS(XMINP,XMAXP,DMIN,DMAX,.2,.999,.15,.5)
CALL SETCH(1.,5.,1,0,1,1)
WRITE (100,52)
52 FORMAT(10HION ENERGY)
CALL TRACE(PTIME,PENGY(1,1),NCH)
CALL SETCH(17.,1.,1,0,1,0)
WRITE (100,53) REALT
53 FORMAT(11HREAL TIME =,E16.6,8H SECONDS)
CALL FRAME
CALL CARTMM(NCH,DMIN,DMAX,PDENS(1,2),1)
DMAX=1.1*DMAX
CALL MAPSSL(XMINP,XMAXP,DMIN,DMAX,.2,.999,.65,.999)
CALL SETCH(1.,45.,1,0,1,1)
WRITE (100,54)
54 FORMAT(12HELEC DENSITY)
CALL TRACE(PTIME,PDENS(1,2),NCH)
CALL CARTMM(NCH,DMIN,DMAX,PENGY(1,2),1)
DMAX=1.1*DMAX
CALL MAPS(XMINP,XMAXP,0.,DMAX,.2,.999,.15,.5)
CALL SETCH(1.,5.,1,0,1,1)
WRITE (100,55)
55 FORMAT(11HELEC ENERGY)
CALL TRACE(PTIME,PENGY(1,2),NCH)
CALL SETCH(17.,1.,1,0,1,0)
WRITE (100,53) REALT
CALL FRAME
KSTORE=K
DIMENSION CV(IYAP),CH(IYAP),PERP(JXA),VPAR(IYAP)
DO 81 KK=1,2
  IF (KK.EQ. 2) GO TO 81
  KP=2*(KK-1)
  DO 80 J=2,JXM1
    PERP(J)=0.
    DO 68 I=1,IY
      CV(I)=X(J)/SINF(Y(I)) $ VPAR(I)=CV(I)*COSF(Y(I))
      CH(I)=0.
      K=J-1
69 K=K+1
      IF (K.GT. JXM1) GO TO 68
      IF (CV(I).LT. X(K).OR. CV(I).GE. X(K+1)) GO TO 69
      CH(I)=G(I,K)+(G(I,K+1)-G(I,K))*(CV(I)-X(K))/(X(K+1)-X(K))
      CH(I)=CH(I)*VPAR(I)*KP
68 CONTINUE
      CALL INTGRL(IY,CH,VPAR,ANS1)
      PERP(J)=-ANS1
80 CONTINUE
      IF (KK.EQ. 1) GO TO 83
      WRITE (3,82)
82 FORMAT(//,31HTHIS PERP DIST CONTAINS VPAR**2,/)
83 CONTINUE
      WRITE (3,75) ((X(J),PERP(J)),J=1,JX)

```

```

75 FORMAT(/,35HION VELOCITY      PERP. DISTRIBUTION,/, (2E16.6))
   CALL MAXMIN(PERP,JX,PMAX,PMIN)
   CALL SETCH(10.,32.,0,0,2,0)
   WRITE (100,76) REALT
76 FORMAT(26HPERP. VELOCITY DISTR. AT ,E10.3,4HSEC.)
   CALL MAPG(X(1),X(JX),0.,PMAX,.15,.999,.15,.9)
   CALL SETCRT(X(1),PERP(1))
   CALL TRACE(X(1),PERP(1),JX)
   IF (KK .EQ. 1) GO TO 85
   CALL SETCH(10.,1.,1,0,1,0)
   WRITE (100,84)
84 FORMAT(31HTHIS PERP DIST CONTAINS VPAR**2)
85 CONTINUE
   CALL FRAME
81 CONTINUE
   K=KSTORE
   RETURN
END
SUBROUTINE AMB1
  USE COMM
  MI = 0
  OLDDEN=DENSE
  OLDE = ENERG
  OLDI=GNORM
  DO 3 I=1,IYP1 $ DO 3 J=1,JX
3  GST(I,J)=G(I,J)
  DO 13 J=1,JXE
13 FEES(J)=FEE(J)
10 MI=MI+1
   CALL GAMMAIN(1)
   CALL IONISOT
   CALL GELECT
   CALL XSWEPT
   CALL GNaNDE
   IF (KGNANDE .EQ. 1) RETURN
   POLDI=PNUMB*OLDI $ PGNORM=PNUMB*GNORM $ DEII=PGNORM-POLDI
   CALL MAXE
   IF (KMAXE .EQ. 1) RETURN
   DEEE = DENSE - OLDDEN
   WRITE (3,465)POLDI,PGNORM,DEII,OLDDEN,DENSE,DEEE
465 FORMAT(10X,7HOLD NI=,E16.6,5X,7HNEW NI=,E16.6,5X,5HDIFF=,E16.6,
  C /,10X,7HOLD NE=,E16.6,5X,7HNEW NE=,E16.6,5X,5HDIFF=,E16.6)
   IF (OLDDEN .EQ. POLDI) GO TO 160
   IF (TIME .LT. 1.000001*DT) GO TO 160
   IF (MIMAX .EQ. 0) GO TO 160
   IF (MI .GE. MIMAX) GO TO 460
   IF (DENST .LT. 0.) GO TO 904
   IF (PGNORM .GT. DENSE .AND. DEII .GT. DEEE) 161,160
904 IF (PGNORM .LT. DENSE .AND. DEII .LT. DEEE) 161,160
161 POTENT=POTENT+DENST*POTENT
   VC2=PNUMB*POTENT/EIONS
   DENSE = OLDDEN
   ENERG = OLDE
   GNORM=OLDI
   DO 4 I=1,IYP1 $ DO 4 J=1,JX
4  G(I,J)=GST(I,J)
   DO 14 J=1,JXE
14 FEE(J)=FEES(J)
   GO TO 10
160 CONTINUE

```

```

      IF (PGNORM .LT. DENSE .AND. DENST .GT. 0.) DENST=-DENST
      IF (PGNORM .GT. DENSE .AND. DENST .LT. 0.) DENST=-DENST
      RETURN
460 WRITE (3,461) POTENT
461 FORMAT(/,36HAMB POT REQ NOT SATISFIED - POTENT =,E16.6)
      KAMB1=1 $ RETURN
      END
      SUBROUTINE GERR
      USE COMM
      DO 10 J=1,JX
      DO 10 I=1,IYP1
      IF (G(I,J) .GE. 0.) GO TO 10
      G(I,J)=0.
10 CONTINUE
      RETURN
      END
      SUBROUTINE COEF
      USE COMM
C THIS SUBROUTINE COMPUTES SOME OF THE COEFFICIENTS FOR THE ION EQUATION.
      GNI=1./GNORM
      FACTOR=.5*TNORM*GNI
      CHBET=3.491E-31 $ CHA=1.111E+15 $ CHB=1.351E+14
      CHC=5.099E-2 $ CHD=2.092E-53 $ CHV1=1.000E+8
      DO 51 J=2,JX
      ENUM=1.-EXP(-CHBET*(X(J)*VNORM)**4)
      DENOM=CHA+CHB*ABSF(LOGF(VNORM*X(J)/CHV1))+CHC*(VNORM*X(J)-CHV1)**2
      C +CHD*(VNORM*X(J))**B
      HCXR(J)=CXRXB(1)*VNORM*X(J)*ENUM/DENOM
      HCXR(J)=HCXR(J)+CXRXB(4)
51 CONTINUE
      CALL BCHNG
      PGNI=TWOP1*GNI*GAM11
      DO 100 I=2,IY
      DO 101 J=2,JXM1
      A1P=.5*GCOXX(I,J) $ DGXY=GCOXY(I,J) $ DGY=GCOPY(I,J)
      DGX=GCOX(I,J) $ DGY=GCOPY(I,J)
      A2P(I,J)=X21(J)*DGXY-X31(J)*DGY
      A3P=.5*(X41(J)*DGY+X31(J)*DGY)
      B1P=.5*X31(J)*(DGY+CTNY(1)*DGY)+X21(J)*DGY
      B2P=X41(J)*DGY-X31(J)*DGXY+.5*X31(J)*CTNY(1)*DGY+.5*
      X41(J)*CTNY2(1)*DGY
      ALPHAP(I,J)=(A1P/DXP5(J)+B1P*.5)*DX1(J)
      BETAP(I,J)=(A1P/DXM5(J)-B1P*.5)*DX1(J)
      GAMMAP(I,J)=(A3P/DYP5(1)+B2P*.5)*DY1(1)
      DELTAP(I,J)=(A3P/DYM5(1)-B2P*.5)*DY1(1)
      CPB2(I,J)=PGNI*G(I,J)
      CPB2(I,J)=CPB2(I,J)-HCXR(J)*FACTOR
101 CONTINUE
100 CONTINUE
      RETURN
      END
      SUBROUTINE GNADE
      USE COMM
C THIS SUBROUTINE COMPUTES THE NORMALIZED ION DENSITY AND ENERGY.
      EN=0.
      DO 100 J = 1,JX
      DO 100 I = 1, IY
100 EN=EN+G(I,J)*XSQ(J)*SPHER1(I,J)
      EEN=EN
      GN=0.

```

```

DO 101 J=1,JX
DO 101 I=1,IY
GN=GN+G(I,J)*SPHERI(I,J)
101 CONTINUE
GGN=GN
IF (GGN .EQ. 0.) GO TO 1
ENERGY=EEN/GGN
GNORM=GGN
RETURN
1 KGNANDE=1
WRITE (3,2)
2 FORMAT(/,12HZERO DENSITY)
RETURN
END
SUBROUTINE XSWEET
USE COMM
DO 3333 I=1,IYJXA
3333 TEMPG(I)=0.
GNELOG=GNORM
GNELOGI=1./GNELOG
RBGN = RHOP/GNELOG
VC2=PNUMB*POTENT/EIONS
C COMPUTE ION LOSS BOUNDARY.
SS5=SINF(Y(2)-.5*DYM5(2))*2
XMIN=SQRTF(VC2/(BRATIO-1.))
JMIN=1
DO 240 J=2,JXM1
IF (X(J-1) + .5*DXM5(J) .LT. XMIN) JMIN=J
IF (JMIN .LT. J) GO TO 301
240 CONTINUE
301 JMP1=JMIN+1
DO 222 J=JMP1,JXM1
IO(J)=1
PFAC=(1.+VC2/XSQ(J))*BRATIO1
IF (PFAC .LE. SS5) GO TO 222
TLC=ASINF(SQRTF(PFAC))
DO 220 I=2,IY
IJUNK=1
IF (TLC .GT. Y(I)-.5*DYM5(I) .AND. TLC .LE. Y(I)+.5*DYP5(I))
C GO TO 304
220 CONTINUE
304 IO(J)=IJUNK
222 CONTINUE
IMIN=IO(JXM1)
IMP1 = IMIN + 1
DO 223 I=IMIN,IY
JO(I)=JMIN
DO 224 J=JMP1,JXM1
IF (IO(J) .EQ. 1) JO(I)=J
224 CONTINUE
223 CONTINUE
IIF=IO(JMP1)-1
DO 302 II=IMIN,IIF
I=IIF+IMIN-II
IF (JO(I) .EQ. JMIN) JO(I)=JO(I+1)
302 CONTINUE
DO 241 I=1,IMIN
DO 241 J=1,JX
241 G(I,J)=0.
DO 242 I=1,IYP1

```



```

      DO 242 J=1,JMIN
242  G(I,J)=0.
C   FIRST COMPUTE ALONG I=CONST ROWS.
C   STORE ROW IMIN OF ARRAY G.
      DO 111 J=JMP1,JX
111  TEMPG(J)=0.
      DO 100 I=IMPI,IY
      JS=J0(I)
      JSP=JS+1
      JXMJP=JX-JSP
C   COMPUTE E AND F (EP AND FP) FOR ADI SCHEME.
      EP(JS)=0.
      FP(JS)=0.
      DMODP=DYM5(I)/DYP5(I)
      DO 101 J=JSP,JXM1
      ALP1=GEXX(J)/(2.*DXP5(J)*DX(J))
      BET1=ALP1*DXP5(J)/DXM5(J)
      ALBE2=(CEX(J)+GEX(J)*X21(J))/(2.*DX(J))
      GAM1=GEX(J)*X31(J)/(2.*DYP5(I)*DY(I))
      DEL1=GAM1*DYP5(I)/DYM5(I)
      GADE2=GEX(J)*CTNY(I)*X31(J)/(4.*DY(I))
      CC1=THOP1*FE(J)/PNUMB
      COMIX1 = A2P(I,J)/(4.*DX(J)*DYM5(I))
      COMIXE=COMIX1*DMODP
      BMIX= BETAP(I,J)+BET1-ALBE2-COMIX1
      BP= ALPHAP(I,J)+BETAP(I,J)+ALP1+BET1-CPB2(I,J)-CC1 +RBGN
      PSI=(GAMMAP(I,J)+GAM1+GADE2)*(G(I+1,J)-G(I,J))+(DELTAP(I,J)+DEL1-
C   GADE2)*(G(I-1,J)-G(I,J))+(CPB2(I,J)+CC1)*G(I,J)+
C   COMIX1*(TEMPG(J-1)-TEMPG(J+1))+COMIXE*(G(I+1,J+1)-G(I+1,J-1)-
C   G(I,J+1)+G(I,J-1))+RBGN*G(I,J)+SOURCE(I,J)*GNELOGI
      DEN=BP-BMIX*EP(J-1)
      EP(J)=(ALPHAP(I,J)+ALP1+ALBE2+COMIX1)/DEN
101  FP(J)=(PSI+BMIX*FP(J-1))/DEN
      TEMPG(JX)= 0.
C   TEMPG NOW CONTAINS NEWLY COMPUTED ROW I-1.  SET G ON ROW I-1 EQUAL TO TEMPG
C   AND COMPUTE G ON ROW I AND STORE IN TEMPG.
      DO 120 J=JMP1,JXM1
120  G(I-1, J) = TEMPG(J)
      DO 102 J=1,JXMJP
102  TEMPG(JX-J)=EP(JX-J)*TEMPG(JX-J+1)+FP(JX-J)
      DO 226 J=1,JS
226  TEMPG(J)=0.
100  CONTINUE
      DO 110 J=JMP1,JXM1
      G(IY, J) = TEMPG(J)
      G(IYP,J) = G(IYM,J)
110  CONTINUE
C   NOW COMPUTE ALONG J=CONST ROWS.
C   STORE ROW JMIN OF ARRAY G.
      DO 112 I=IMPI,IYP1
112  TEMPG(I)=0.
C   COMPUTE E AND F (EP AND FP) FOR ADI SCHEME.
      DO 105 J=JMP1,JXM1
      EP(IO(J))=0.
      FP(IO(J))=0.
      ISM=IO(J)-1
      IST=IO(J)+1
      ALP1=GEXX(J)/(2.*DXP5(J)*DX(J))
      BET1=ALP1*DXP5(J)/DXM5(J)
      ALBE2=(CEX(J)+GEX(J)*X21(J))/(2.*DX(J))

```

```

CC1=TWOP1*FE(J)/PNUMB
DO 106 I=1ST,IY
  GAM1=GEX(J)*X31(J)/(2.*DYP5(I)*DY(I))
  DEL1=GAM1*DYP5(I)/DYM5(I)
  GADE2=GEX(J)*CTNY(I)*X31(J)/(4.*DY(I))
  COMIX1 = A2P(I,J)/(4.*DY(I)*DXM5(J))
  COMIXE = COMIX1*DXM5(J)/DXP5(J)
  DMIX= DELTAP(I,J)+DEL1-GADE2-COMIXI
  BP= GAMMAP(I,J)+DELTAP(I,J)+GAM1+DEL1-CPB2(I,J)-CC1 + RBGN
  CH1= (ALPHAP(I,J)+ALP1+ALBE2)*(G(I,J+1)-G(I,J))+(BETAP(I,J)+BET1-
  1ALBE2)*(-G(I,J)+G(I,J-1))+(CPB2(I,J)+CC1)*G(I,J)+COMIXI*(-TEMPG
  2(I+1)+TEMPG(I-1))+COMIXE*(G(I+1,J+1)-G(I-1,J+1)-G(I+1,J)+G(I-1,J))
  3 +RBGN*G(I,J) + SOURCE(I,J)/GNELOG
  DEN=BP-DMIX*EP(I-1)
  EP(I)=(GAMMAP(I,J)+GAM1+GADE2+COMIXI)/DEN
106 FP(I) = (CH1 + DMIX * FP(I - 1))/DEN
C TEMPG NOW CONTAINS NEWLY COMPUTED ROW J-1. SET G ON ROW J-1 EQUAL TO TEMPG
C AND COMPUTE G ON ROW J AND STORE IN TEMPG.
DO 121 I=1MP1,IYP1
121 G(I,J-1) = TEMPG(I)
  TEMPG(IYP1) = (EP(IYM1)*FP(IY) + FP(IYM1))/(1.- EP(IYM1)*EP(IY))
  IYM=IYP1-10(J)
DO 107 I=1,IYM
107 TEMPG(IYP1-I) = EP(IYP1-I)*TEMPG(IYP1-I+1) + FP(IYP1-I)
DO 228 I=1,ISM
228 TEMPG(I) = 0.
105 CONTINUE
DO 113 I=1MP1,IYP1
  G(I, JXM1) = TEMPG(I)
113 CONTINUE
  CALL GERR
  IF (OCP.EQ. 0.) RETURN
  RHOP2=.5*RHOP
DO 400 I=1,IYP1 $ DO 400 J=1,JX
400 GST2(I,J)=0.
DO 401 I=1MP1,IY
  JS=JO(I) $ JSP=JS+1 $ EP(JS)=0. $ FP(JS)=0.
DO 402 J=JSP,JXM1
  TXSD=2.*XSQ(J)*DX(J)
  ALP=DBIP(I,J+1)/TXSD $ GAM=-DBIP(I,J-1)/TXSD
  BET=RHOP2 $ DEL=RHOP2*G(I,J)
  DEN=BET-GAM*EP(J-1)
  EP(J)=ALP/DEN $ FP(J)=(DEL+GAM*FP(J-1))/DEN
402 CONTINUE
  GST2(I,JX)=0. $ DO 403 JJ=JS,JXM1
  J=JS+JXM1-JJ
403 GST2(I,J)=EP(J)*GST2(I,J+1)+FP(J)
401 CONTINUE
DO 404 J=1,JX
DO 399 I=1,IYP1
399 G(I,J)=0.
404 GST2(IYP1,J)=GST2(IYM1,J)
DO 405 JJ=JMP1,JXM1
  J=JMP1+JXM1-JJ $ IS=10(J) $ ISP=IS+1
  EP(IS)=0. $ FP(IS)=0.
DO 406 I=ISP,IY
  TXSD=2.*XSQ(J)*SINN(I)*DY(I)
  ALP=DB2P(I+1,J)/TXSD $ GAM=-DB2P(I-1,J)/TXSD
  BET=RHOP2 $ DEL=RHOP2*GST2(I,J)
  DEN=BET-GAM*EP(I-1)

```

```

EP(I)=ALP/DEN $ FP(I)=(DEL+GAM*FP(I-1))/DEN
406 CONTINUE
G(IYPI,J)=(EP(IYM1)*FP(IY)+FP(IYM1))/(1.-EP(IYM1)*EP(IY))
DO 407 I1=IS,IY
  I1=IS+IY-I1
407 G(I,J)=EP(I)*G(I+1,J)+FP(I)
405 CONTINUE
CALL GERR
RETURN
END
SUBROUTINE PREEL
USE COMM
BEE=.75*EMASS*AMU*VNORM**2/(ECON*EIONS*ENERGE)
XEMAX=SQRTF(-LOGF(FEMIN)/BEE)
IF (XEMAX.LT. XMAX) XEMAX=XMAX
H=XEMAX/FLOATF(JXEM1)
XE(1)=0. $ XE(JXE)=XEMAX
XE(2)=H $ RAM=1.
DO 1 J=3,JXEM1
  1 XE(J)=XE(J-1)+RAM*(XE(J-1)-XE(J-2))
  WRITE (3,10) ((J,XE(J)),J=1,JXE)
10 FORMAT(/,7HXE MESH,/, (15,E16.6))
DO 2 J=2,JXE
  2 DXEM(J)=XE(J)-XE(J-1)
  DXEM(1)=0. $ DXEM(JXEP1)=0.
  DO 3 J=1,JXE
    XE2(J)=XE(J)**2
  3 DXE(J)=.5*(DXEM(J)+DXEP(J))
  DO 4 J=1,JXE
    4 FEE(J)=EXP(-BEE*XE2(J))
    S=0. $ DO 5 J=1,JXE
      5 S=S+FEE(J)*XE2(J)*DXE(J)
    S=FOURPI*S
    FACT=DENSE/S
    DO 6 J=1,JXE
      6 FEE(J)=FACT*FEE(J)
      XOS2=EES/(EMOIM*EIONS)
      XOS=SQRTF(XOS2)
      DO 7 J=1,JXE
        7 DELE(J)=EXP(-XSOE*(XE(J)-XOS)**2)
        S=0. $ DO 8 J=1,JXE
          8 S=S+DELE(J)*XE2(J)*DXE(J)
          FACT=TNORM/(FOURPI*DENSNO*S)
          DO 9 J=1,JXE
            9 DELE(J)=FACT*DELE(J)
            COE=PCURROE $ DCOE=DPCURROE*DENSNO
            DCESE=PCURROSE*SQRTF(DENSNO)
            RETURN
          END
        SUBROUTINE GAMMAIN(INDEX)
        USE COMM
        IF (INDEX.EQ. 1) GO TO 1
        ALP=1./137. $ RO=2.82E-13 $ CLIGHT=3.E+10
        GACON1=2.*ALP/(SQRTF(3.*PI)*RO*CLIGHT)
        GACON2=GACON1/(3.7943E-4*EMASS*AMU)
        GAM1IS=.5*PMASS*AMU*GACON2
        GAMIES=PMASS*EMASS*AMU*GACON2/(PMASS+EMASS)
        GAMEIS=GAMIES
        GAMEES=.5*EMASS*AMU*GACON2
1 CONTINUE

```

```

DEBY=SQRTF(ENERGE*EIONS*1.E+3/(DENSE*DENSNO))
SI=ENERGY/PMASS $ SE=ENERGE/EMASS
SE1=SI $ IF (SE.GT. SI) SE1=SE
EEA=EIONS*ECON/AMU
SI=SI*EEA $ SE=SE*EEA $ SE1=SE1*EEA
SIS=SQRTF(SI) $ SES=SQRTF(SE) $ SEIS=SQRTF(SE1)
GAM3=GAM1IS*DEBY*SIS $ GAM1I=LOGF(GAM3)-.5
GAM3=GAMIES*DEBY*SEIS $ GAMIE=LOGF(GAM3)-.5
GAMEI=GAMIE
GAM3=GAMEES*DEBY*SES $ GAMEE=LOGF(GAM3)-.5
RETURN
END
SUBROUTINE GELECT
USE COMM
CALL MOMENTS
EMCON=1.-1./EMOIM
DO 1 J=2,JXM1
EFE=FEI(J)
FE(J)=EFE*GAMIE/(GNORM*PNUMB*EMOIM)
EMOFE=EMOF(J) $ ENOFE=ENOF(J) $ EEOF=EEOF(J)
CEX(J)=-GAMIE*FOURPI*ENOFE*EMCON/(GNORM*XSQ(J)*PNUMB**2)
GEX(J)=GAMIE*FOURPI*((2.*X(J)*EMOFE-X2I(J)*EEOF)/3.*ENOFE)/
C (GNORM*PNUMB**2)
GEXX(J)=GAMIE*FOURPI*2.*(EEOF*X3I(J)+EMOFE)/(3.*GNORM*PNUMB**2)
1 CONTINUE
RETURN
END
SUBROUTINE MOMENTS
USE COMM
DO 1 J=1,JX
DO 2 JJ=1,JXEM1
IF (X(J).LT. XE(JJ).OR. X(J).GE. XE(JJ+1)) GO TO 2
JP=JJ $ GO TO 3
2 CONTINUE
JP=JXEM1
3 CONTINUE
W1=(X(J)-XE(JP))/DXEP(JP) $ W0=1.-W1
FEI(J)=W0*FEE(JP)+W1*FEE(JP+1)
1 CONTINUE
S=0. $ DO 19 JJ=JP,JXEM1
J=JP+JXEM1-JJ
SM=.5*DXEP(J)*(XE(J)*FEE(J)+XE(J+1)*FEE(J+1))
S=S+SM
19 CONTINUE
EM1=S-W1*SM
EMOF(JX)=EM1 $ ENOF(1)=0. $ EEOF(1)=0.
DO 4 J=2,JX
JJ=JX+1-J
T1=.5*DXP5(JJ)*(X(JJ)*FEI(JJ)+X(JJ+1)*FEI(JJ+1))
T2=.5*DXM5(J)*(XSQ(J-1)*FEI(J-1)+XSQ(J)*FEI(J))
T4=.5*DXM5(J)*(XSQ(J-1)**2*FEI(J-1)+XSQ(J)**2*FEI(J))
EMOF(JJ)=EMOF(JJ+1)+T1 $ ENOF(J)=ENOF(J-1)+T2
EEOF(J)=EEOF(J-1)+T4
4 CONTINUE
RETURN
END
SUBROUTINE IONISOT
USE COMM
DO 1 J=1,JX
S=0. $ DO 2 I=1,IY

```

```

2 S=S+CYNT(1)*SINN(1)*G(1,J)
1 TEMPG(J)=.5*S
  DO 5 J=1,JXE
  DO 3 JJ=1,JXM1
    IF (XE(J) .LT. X(JJ) .OR. XE(J) .GE. X(JJ+1)) GO TO 3
    JP=JJ $ GO TO 4
3 CONTINUE
  FIE(J)=0. $ GO TO 5
4 CONTINUE
  W1=(XE(J)-X(JP))/DXP5(JP) $ W0=1.-W1
  FIE(J)=W0*TEMPG(JP)+W1*TEMPG(JP+1)
5 CONTINUE
  RETURN
  END
  SUBROUTINE GETCURE
  USE COMM
  CURRE=COE+DCOE*DENSE+DCESE*SQRTF(DENSE)
  REE=EIONS*ENERGE $ RDE=DENSNO*DENSE $ DNNE=CXRXB(2)
  REE2=REE**2
  SIGA=9.109E+4 $ SIGB=6.535E+5 $ SIGC=1.330E+7
  SIGVNE=REE2/(SIGA+SIGB*REE+SIGC*REE2)
  CURRE=CURRE+SIGVNE*DNNE*RDE
  RETURN
  END
  SUBROUTINE MAXE
  USE COMM
  TECON=FOURPI/(PNUMB**2*EMOIM)**2
  TIE=TECON*PNUMB**2*GAME1 $ TEE=TECON*GAMEE
  VC2=POTENT/EIONS $ XCN2=VC2/EMOIM $ XCN=SQRTF(XCN2)
  QNODT=1./DT
  CALL GETCURE
  ENF1=0. $ EEFE=0. $ ENFE=0. $ EEFE=0.
  ALPE(1)=0. $ BETE(1)=0. $ GAME(1)=0.
  DO 1 J=2,JXE
    ENF1=ENF1+.5*DXEM(J)*(XE2(J-1)*FIE(J-1)+XE2(J)*FIE(J))
    ENFE=ENFE+.5*DXEM(J)*(XE2(J-1)*FEE(J-1)+XE2(J)*FEE(J))
    EEFE=EEFE+.5*DXEM(J)*(XE2(J-1)**2*FIE(J-1)+XE2(J)**2*FIE(J))
    EEFE=EEFE+.5*DXEM(J)*(XE2(J-1)**2*FEE(J-1)+XE2(J)**2*FEE(J))
    ALPE(J)=TIE*ENF1*EMOIM+TEE*ENFE
    BETE(J)=(TIE*EEFE+TEE*EEFE)/(3.*XE(J))
    GAME(J)=TIE*(ENF1-EEFE)/(3.*XE2(J))+TEE*(ENFE-EEFE)/(3.*XE2(J))
1 CONTINUE
  EMF1=0. $ EMFE=0.
  DO 2 JJ=2,JXE
    J=JXEP1-JJ+1
    EMF1=EMF1+.5*DXEP(J)*(XE(J)*FIE(J)+XE(J+1)*FIE(J+1))
    EMFE=EMFE+.5*DXEP(J)*(XE(J)*FEE(J)+XE(J+1)*FEE(J+1))
    BETE(J)=BETE(J)+XE2(J)*(TIE*EMF1+TEE*EMFE)/3.
    PROB=0. $ IF (XE(J) .LE. XCN) GO TO 3
    REFF=BRATIO/(1.-XCN2/XE2(J))
    PROB=2.3025851/LOGF(REFF)
3 CONTINUE
  GAME(J)=GAME(J)+2.*XE(J)*(TIE*EMF1+TEE*EMFE)/3.
  GAME(J)=.5*GAME(J)*PROB/XE(J)
2 CONTINUE
  DO 4 J=2,JXEM1
    ANP2=.25*(ALPE(J)+ALPE(J+1)) $ ANM2=.25*(ALPE(J-1)+ALPE(J))
    BNP2=.5*(BETE(J)+BETE(J+1))/DXEP(J)
    BNM2=.5*(BETE(J-1)+BETE(J))/DXEM(J)
    AA(J)=(ANP2+BNP2)/(XE2(J)*DXE(J))

```

```

BB(J)=ONODT-((ANP2-ANM2-BNP2-BNM2)/DXE(J)-GAME(J))/XE2(J)
CC(J)=(-ANM2+BNM2)/(XE2(J)*DXE(J))
DD(J)=ONODT*FEE(J)+CURRE*DELE(J)
4 CONTINUE
ANP2=.5*ALPE(2) $ BNP2=BETE(2)/DXEM(2)
EE(1)=(BNP2+ANP2)/(BNP2-ANP2)
FF(1)=0.
DO 11 J=2,JXEM1
DEN=BB(J)-CC(J)*EE(J-1)
EE(J)=AA(J)/DEN
(J)=(DD(J)+CC(J)*FF(J-1))/DEN
11 CONTINUE
ANP2=.5*(ALPE(JXEM1)+ALPE(JXE))
BNP2=(BETE(JXEM1)+BETE(JXE))/DXEP(JXEM1)
ABN2=(BNP2+ANP2)/(BNP2-ANP2)
FEE(JXE)=FF(JXEM1)/(ABN2-EE(JXEM1))
DO 12 JJ=1,JXEM1
J=JXE-JJ
FEE(J)=EE(J)*FEE(J+1)+FF(J)
12 CONTINUE
DO 14 J=1,JXE
IF (FEE(J).LT. 0.) FEE(J)=0.
14 CONTINUE
IF (DCP.EQ. 0.) GO TO 20
FACT=-DCP/3.
SUMC=BETE(1)+BETE(2) $ DENOM=SUMC-DXEM(2)*ALPE(1)
EE(1)=(SUMC+ALPE(2)*DXEM(2))/DENOM $ FF(1)=0.
DO 21 J=2,JXEM1
TXSD=2.*XE2(J)*DXE(J)
ALP=XE(J+1)*XE2(J+1)*FACT/TXSD
GAM=-XE(J-1)*XE2(J-1)*FACT/TXSD
BET=ONODT $ DEL=ONODT*FEE(J)
DEN=BET-GAM*EE(J-1)
EE(J)=ALP/DEN $ FF(J)=(DEL+GAM*FF(J-1))/DEN
21 CONTINUE
FEE(JXE)=0. $ DO 22 JJ=1,JXEM1
J=JXE-JJ
22 FEE(J)=EE(J)*FEE(J+1)+FF(J)
DO 23 J=1,JXE
IF (FEE(J).LT. 0.) FEE(J)=0.
23 CONTINUE
20 CONTINUE
S2=0. $ S4=0.
DO 13 J=1,JXE
S2=S2+DXE(J)*XE2(J)*FEE(J)
S4=S4+DXE(J)*XE2(J)**2*FEE(J)
13 CONTINUE
IF (S2.EQ. 0) GO TO 15
DENSE=S2*FOURPI
ENERGE=EMOIM*S4/S2
RETURN
15 KMAXE=1
RETURN
END
SUBROUTINE QHOLD2
USE COMM
DIMENSION SIGMA(JXA)
DO 1 J=1,JX
S=0.
DO 2 I=1,IY

```

```

2 S=S+CYNT(1)*SINF(Y(1))*G(1,J)
1 TEMPG(J)=S
WRITE (3,16) ((J,TEMPG(J)),J=1,JX)
16 FORMAT(/,10HVELOC DIST,/, (15,E16.6))
UM2=5./3.
UM=SQRTF(UM2)
S=0.
DO 3 J=1,JX
3 S=S+CINT(J)*XSQ(J)*TEMPG(J)
DENS=S
FACTOR=UM*VNORM/DENS
UMU=UM2*EIONS
CALL FINDSIG(UMU,SIGMA)
DO 4 J=1,JX
ENGY=UMU*XSQ(J)
WRITE (3,5) ENGY,SIGMA(J)
5 FORMAT(7HENERGY=,E16.6,5X,4HSIG=,E16.6)
4 CONTINUE
S=0.
DO 6 J=1,JX
S=S+CINT(J)*SIGMA(J)*TEMPG(J)*X(J)*XSQ(J)
6 CONTINUE
SIGV=FACTOR*S
WRITE (3,7) SIGV
7 FORMAT(/,17HHOLDREN SIGMAYEE=,E16.6)
RETURN
END
SUBROUTINE QFUTCH2
USE COMM
COMMON/MQCAL/LT,TEMPGT(JXA)
DIMENSION SIGMA(JXA)
DIMENSION ANSWER(2)
DO 1 J=1,JX
S=0.
DO 2 I=1,IY
2 S=S+CYNT(1)*SINF(Y(1))*G(1,J)
1 TEMPG(J)=S
SQIP5=SQRTF(1.5)
TEMPGT(1)=TEMPG(1)
DO 162 J=2,JX
QJX3=SQIP5*X(J)
DO 163 JJ=1,JXM1
IF (X(JJ) .LE. QJX3 .AND. QJX3 .LT. X(JJ+1)) GO TO 164
163 CONTINUE
TEMPGT(J)=0. $ GO TO 162
164 RATIO=(QJX3-X(JJ))/DXP5(JJ)
TEMPGT(J)=(1.-RATIO)*TEMPG(JJ)+RATIO*TEMPG(JJ+1)
162 CONTINUE
WRITE (3,631) ((JL,TEMPG(JL),TEMPGT(JL)),JL=1,JX)
631 FORMAT(15,2E16.6)
UMU=EIONS
DO 30 LT=1,2
IF (LT .EQ. 1) GO TO 31
CALL FINDSIG(UMU,SIGMA)
DO 34 J=1,JX
ENGY=UMU*XSQ(J)
WRITE (3,35) ENGY,SIGMA(J)
35 FORMAT(7HENERGY=,E16.6,5X,4HSIG=,E16.6)
34 CONTINUE
GO TO 32

```

```

31 CONTINUE
DO 33 J=1,JX
33 SIGMA(J)=1.
32 CONTINUE
QANS=0.
DO 20 J=1,JX
20 QANS=QANS+CINT(J)*EVALS(J,SIGMA)
ANSWER(LT)=QANS
30 CONTINUE
SIGV=VNORM*ANSWER(2)/ANSWER(1)
WRITE (3,7) SIGV
7 FORMAT(/,15HFUTCH SIGMAYEE=,E16.6)
RETURN
END
FUNCTION EVALS(J,SIGMA)
USE COMM
COMMON/MQCAL/LT,TEMPGT(JXA)
DIMENSION SIGMA(1)
DIMENSION STFG(JXA),UPART(JXA)
DO 1 KK=1,J
1 STFG(KK)=(XSQ(J)-XSQ(KK))*GETFG(J,KK)
UPART(1)=0.
DO 2 KK=2,J
2 KK=2,J
ADDIT=.5*DXM5(KK)*(STFG(KK-1)+STFG(KK))
UPART(KK)=UPART(KK-1)+ADDIT
2 CONTINUE
SUM=0.
DO 4 KK=2,J
4 KK=2,J
ARGM=X(KK-1)*LT*SIGMA(KK-1)*UPART(KK-1)
ARGP=X(KK)*LT*SIGMA(KK)*UPART(KK)
SUM=SUM+.5*DXM5(KK)*(ARGM+ARGP)
4 CONTINUE
EVALS=SUM
RETURN
END
FUNCTION GETFG(J,KK)
USE COMM
COMMON/MQCAL/LT,TEMPGT(JXA)
ARG1=.5*(X(J)+X(KK))
ARG2=.5*(X(J)-X(KK))
J1=0 $ J2=0
DO 1 JM=1,JXM1
1 IF (X(JM) .LE. ARG1 .AND. ARG1 .LT. X(JM+1)) J1=JM
IF (X(JM) .LE. ARG2 .AND. ARG2 .LT. X(JM+1)) J2=JM
1 CONTINUE
IF (ARG1 .GE. X(JX)) J1=JXM1
IF (ARG2 .GE. X(JX)) J2=JXM1
FRAC1=(ARG1-X(J1))/DXP5(J1)
FRAC2=(ARG2-X(J2))/DXP5(J2)
ANS1=(1.-FRAC1)*TEMPG(J1)+FRAC1*TEMPG(J1+1)
ANS2=(1.-FRAC2)*TEMPG(J2)+FRAC2*TEMPG(J2+1)
ANS3=(1.-FRAC2)*TEMPG(J2)+FRAC2*TEMPG(J2+1)
ANS4=(1.-FRAC1)*TEMPG(J1)+FRAC1*TEMPG(J1+1)
GETFG=ANS1*ANS2+ANS3*ANS4
RETURN
END
SUBROUTINE FINDSIG(UMU,SIGMA)
USE COMM
DIMENSION SIGMA(1)
SIGMA(1)=0.

```



```

DO 1 J=2,JX
EPS=UMU*X(J)**2
IF (EPS .GE. 20.) GO TO 41
TEMPQ=2.25E+4*EXP(-44.4/SQRTF(EPS))/EPS $ GO TO 40
41 IF (EPS .GE. 30.) GO TO 42
TEMPQ=4.62E-4*EPS*EPS-0.137 $ GO TO 40
42 IF (EPS .GE. 50.) GO TO 43
TEMPQ=6.70E-4*EPS*EPS-0.320 $ GO TO 40
43 IF (EPS .GE. 87.) GO TO 44
TEMPQ=8.55E-02*EPS-2.93 $ GO TO 40
44 IF (EPS .GE. 107.) GO TO 45
TEMPQ=1.98E+4/((EPS-107.))**2+4000.) $ GO TO 40
45 IF (EPS .GE. 137.) GO TO 46
TEMPQ=3.56E+4/((EPS-107.))**2+7200.) $ GO TO 40
46 IF (EPS .GE. 185.) GO TO 47
TEMPQ=8.70-0.0314*EPS $ GO TO 40
47 IF (EPS .GE. 400.) GO TO 48
TEMPQ=7.050E+3/(EPS*SQRTF(EPS)) $ GO TO 40
48 TEMPQ=255./(EPS-110.)
40 CONTINUE
SIGMA(J)=1.E-24*TEMPQ
1 CONTINUE
RETURN
END
SUBROUTINE RESTART
USE COMM
CALL QHOLD2
CALL QFUTCH2
CALL PLOTE
CALL OFFMON
DIMENSION ND(5) $ ND(1)=0 $ ND(2)=1HF $ ND(3)=6 $ ND(4)=0
ND(5)=1 $ CALL HSP(ND)
CALL CRTID(IGHAM RESTART,1,1)
RETURN
END
SUBROUTINE OFFMON
COMMON/GOBCOM/IX(41)
IX(24)=IX(24) .INT. 777777777777770077B
IADD= .LOC. RSTRT
CALL QUIT(IADD)
RSTRT RETURN
END
SUBROUTINE INTGRL(N,F,EP,ANS)
DIMENSION EP(N),F(N),CONST(100)
HPLUS=EP(3)-EP(2)
HMINUS=EP(2)-EP(1)
CONST(1)=(2.*HMINUS**3-HPLUS**3+3.*HMINUS**2*HPLUS)/
C (6.*HMINUS*(HPLUS+HMINUS))
NN1=N-1
NN2=N-2
DO 11 I=3,NN2,2
HPLUS=EP(I)-EP(I-1)
HMINUS=EP(I-1)-EP(I-2)
HPLUSP=EP(I+2)-EP(I+1)
HMINUSP=EP(I+1)-EP(I)
CONST(1)=(2.*HPLUS**3-HMINUS**3+3.*HPLUS**2*HMINUS)/
C (6.*HPLUS*(HPLUS+HMINUS)) +
C (2.*HMINUSP**3-HPLUSP**3+3.*HMINUSP**2*HPLUSP) /
C (6.*HMINUSP*(HPLUSP+HMINUSP))
11 CONTINUE

```

```

      A=.5*N
      B=N/2
      L=0
      IF ((A-B) .GT. 1.E-3) L=1
C   IF L=1 N IS ODD.
      I=N-1
      IF (L .EQ. 1) I=1
      HPLUS=EP(I)-EP(I-1)
      HMINUS=EP(I-1)-EP(I-2)
      CONST(I)=(2.*HPLUS**3-HMINUS**3+3.*HPLUS**2*HMINUS) /
C   (6.*HPLUS*(HPLUS+HMINUS))
      IF (L .EQ. 0) CONST(I)=CONST(I)+(EP(N)-EP(N-1))/3.
      IF (L .EQ. 0) CONST(N)=0.
      DO 12 I=2,NN1,2
      HPLUS=EP(I+1)-EP(I)
      HMINUS=EP(I)-EP(I-1)
      CONST(I)=(HPLUS**3+HMINUS**3+3.*HPLUS*HMINUS*(HPLUS+HMINUS)) /
C   (6.*HPLUS*HMINUS)
      12 CONTINUE
C   CONSTANTS NOW COMPLETE--I=1,N.
      SUM=0.
      DO 14 I=1,N
      SUM=SUM+CONST(I)*F(I)
      14 CONTINUE
      ANS=SUM
      RETURN
      END
      SUBROUTINE MAXMIN(X,N,X1,X0)
C
C   TO FIND MAXIMUM AND MINIMUM VALUES OF ARRAY X
C
C   DIMENSION X(8192)
      BIG=X(1)
      DO 1 I=2,N
      1 BIG=MAX1F(BIG,X(I))
      X1=BIG
      SML=X(1)
      DO 2 I=2,N
      2 SML=MIN1F(SML,X(I))
      X0=SML
      RETURN
      END

```