Dr. 1810-4

I-11439

UCID- 19864

PSOLV: A CODE FOR CALCULATING THE POTENTIALS
AND DENSITIES IN MFTF-B

Jeffrey A. Colborn

August 17, 1983

Lawrence
Livermore
Laboratory

This is an informal report intended primarily for internal or limited external distribution. The opinions and conclusions stated are those of the author and may or may not be those of the Laboratory.

Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore Laboratory under Contract W-7405-Eng-48.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

UCID--19864

DE84 000821

PSOLV: A CODE FOR CALCULATING THE POTENTIALS AND DENSITIES IN MFTF-B

Jeffrey A. Colborn*

DISCLAIMER

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

*The author is an undergraduate in Engineering Physics at the University of California, Berkeley, and is employed for the summer of 1983 by CDI Corporation to work under contract at LLNL.

MASTER

-1-11

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Bea

CONTENTS

Introduction and Theory	1
Purpose and History of the Code	1
Solving Scheme	1
Model Distribution Functions	4
Calculation of Densities	10
Results	11
Conclusions	12
Acknowledgments	13
Appendices	
A. Phase Space Diagrams for Charged-Particle Classes Present in MFTF-B	14
B. Expressions for the Densities	21
Introduction	21
Densities at X	22
Densities at MXO	23
Densities at MAI	24
Densities at b	29
Densities at A	32
C. Input List	38
D. Sample Output	41
E. Code Listing	46
References	70

INTRODUCTION AND THEORY

Purpose and History of the Code

Code PSOLV solves for potentials and densities at the cardinal points of MFTF-B (see Fig. 1). The code is equipped to handle both the throttle-coil and the axicell geometries. For the throttle-coil case, the potential at point MXO is input, while the potentials and densities at points MAI, b, and A are calculated. For the axicell case, the code must additionally solve for the potentials and densities at points X and MXO. PSOLV is intended primarily for use as a subroutine in TREQ, a code being developed by Rensink² that calculates the densities and potentials at the cardinal points of MFTF-B as a function of time. TREQ is to be used for modeling start-up behavior.

The code on which PSOLV is based was written for TMX-U by Boghosian and Gilmore. Boghosian's version³ solves for the potentials and beta-reduced fields at the cardinal points and tries to solve for ϕ and B everywhere axially by attempting to connect the cardinal points with lines of quasi-neutrality in $\phi - B$ space. Gilmore developed a truncated version, QNOS, which solves quasi-neutrality at the inboard mirror, barrier midplane, and potential peak.⁴ It does not solve for the fields and does not include the conditions $d\phi/dB = 0$ at the potential peak and $(n\tau)_i = (n\tau)_e$, both of which are present in Boghosian's code. Hence, in the truncated version, the number of equations to be solved is reduced from nine to three, which results in faster convergence. This faster convergence makes the truncated version much more suitable for use in time-dependent codes such as TREQ. For this reason, I used this truncated version as a starting point in writing the code for MFTF-B.

Solving Scheme

Code PSOLV calls COSNAF, a NAGFLIB library routine that solves N nonlinear equations of the form $R(\underline{X}) = 0$ where $\underline{X} = (X_1, X_2, \dots, X_N)$ by a hybrid Newton-Raphson method of Powell.⁵ The library routine COSNAF calls RESID1, which calculates the residuals $R_1, R_2, \dots, R_N$ that correspond to the set of variables $\underline{X}$. COSNAF iterates until it finds a point in X -space for which the sum of the squares of the residuals is less than a given tolerance (e.g., 10^{-10}).

Fig. 1. Axial magnetic field and potential profiles for MFTF-B (from Jong, Ref. 1). (A) Axicell geometry. (b) Throttle-coil geometry. The cardinal points are as follows: MXI, inboard axicell mirror; X, axicell midplane; MXO, outboard axicell mirror; MAI, inboard anchor mirror; b, barrier midplane; A, confining potential peak; and MAO, outboard anchor mirror.

For the throttle-coil case, N equals 4, the variables being alpha (the free parameter in the Cohen trapped ion distribution function),⁶ ϕ_{MAI}/T_{ec} , ϕ_b/T_{ec} , and ϕ_c/T_{ec} (see Fig. 1). The potentials are divided by the central-cell electron temperature to yield variables of order unity. Normalizing the variables and residuals to order unity in this way tends to improve the reliability and speed of convergence of COSNAF.⁴ I leave alpha unnormalized because, although it can take on values from $-\infty$ to $+\infty$, it is almost always on the order of ± 1 to 10.

Alpha must be included as a variable for the following reason. Because the trapped and passing regions of phase space are contiguous (see Fig. 4), the distribution functions (DFs) must agree everywhere along their mutual boundary. Hence, the passing- and trapped-ion densities must both be normalized to the central-cell density. Because the trapped-ion density is needed at two points, MAI and b, the densities cannot simply be specified; they must be consistent with each other. Therefore, both must be calculated in the same way, the only free parameter being alpha. Since I want to input the trapped density at b rather than specify alpha, it is best to set the trapped density at b and solve for the value of alpha that yields this particular density. Alpha is then fixed and the trapped density at MAI can be calculated.

In the throttle-coil case, the code solves quasi-neutrality at MAI, b, and A and demands that the calculated trapped-ion density at b be equal to the desired trapped-ion density at b. The residuals are the following:

$$R(1) = \frac{1}{n_{cc}} \sum_j q_j n_j^{MAI} ,$$

$$R(2) = \frac{1}{n_{cc}} \sum_j q_j n_j^b ,$$

$$R(3) = \frac{1}{n_{cc}} \sum_j q_j n_j^A ,$$

$$R(4) = \frac{1}{n_{cc}} (n_T^b - n_{T_0}^b) ,$$

where the superscripts MAI, b, and A indicate the point concerned; the subscripts T and T_0 indicate the calculated and input trapped-ion densities,

respectively; q is in units of e ; and the sum is over classes of charged particles. The sums are divided by the central-cell density to yield residuals of order unity.

In the axicell case, the potential at MXO cannot be specified. The potentials at X and MXO must be solved for and must be consistent with each other and with ϕ_e . Therefore, two more equations are necessary. The additional variables are ϕ_{MXO}/T_{ec} and ϕ_X/T_{ec} and the residuals are the following:

$$R(5) = \frac{1}{n_{cc}} \sum_j q_j n_j^{MXO} ,$$

$$R(6) = \frac{1}{n_{cc}} \sum_j q_j n_j^X .$$

Model Distribution Functions

The various classes of charged particles in MFTF-B are illustrated in Fig. 2. Figure 3 shows, in epsilon-mu space, the regions in which different electron DFs apply, where epsilon and mu are defined as follows:

$$\epsilon \equiv K + \phi$$

$$\mu \equiv \frac{K_{\perp}}{B} .$$

The charge has been suppressed, and K is the kinetic energy of the particle. In all density calculations, I use DFs that were constructed or used previously by Boghosian.⁷

In region I, the passing region, I use Boghosian's "cut-off" Maxwellian. This DF is a Maxwellian with a term subtracted to allow the function to go to zero continuously at the loss boundary,

$$f_I = C [\exp(-\epsilon/T) - \exp(-\mu B_{MAD}/T)] .$$

For the hot and warm electrons (regions II, IIa, III, IV, and V), I use the Cohen model DF for ECRH-dominated electrons, which is constant along rf heating characteristics (see Fig. 2).⁸ Notice that because the DFs must be continuous across boundaries, the function for the hot electrons (regions II, IIa, and III) must be zero everywhere in region IIa; and in region II the

Fig. 2. Phase space diagram for electrons at point A, assuming the transiting electrons are rf dominated (see text and Fig. 3). This diagram is relevant for both the axicell and throttle-coil cases. In all phase diagrams in this report, the lines labeled by point names are the $V_{II} = 0$ lines at that point. The slopes of the lines equal the local B-field, and the intercepts equal plus (for ion diagrams) or minus (for electron diagrams) the local potential. For example, in this diagram, the line labeled MXO is $\epsilon = \mu B_{MXO} - \phi_{MXO}$.

Fig. 3. Particle classes relevant to density calculations at points X, MXO, MAI, b, and A.

function must have a constant value along each characteristic that is equal to the passing DF at the point of intersection of the characteristic and of the line $v_{MXO}^b = 0$. Therefore, the DF in region II is as follows:

$$f_{II} = C \left\{ \exp \left[\frac{\phi_{MXO}}{T} - \left(\frac{B_{MXO}}{B_{MXO} - \beta B_A} \right) \left(\frac{\epsilon - \mu \beta B_A + \phi}{T} \right) \right] \right. \\ \left. - \exp \left[- \left(\frac{B_{MAO}}{B_{MAO} - \beta B_A} \right) \left(\frac{\epsilon - \mu \beta B_A + \phi_{MXO}}{T} \right) \right] \right\} ,$$

where β is an adjustable parameter that can be set equal to one or B_b/B_A . If rf is assumed to dominate at point A, beta equals one; If rf is assumed to dominate at point b, beta equals B_b/B_A .⁸ The region V (warm) DF, similar to the function in region II, must be constant along characteristics and must match the passing electron DF on the line $v_{II}^b = 0$. It is therefore given by the following:

$$f_V = C \left\{ \exp \left[\frac{\phi_b}{T} - \left(\frac{B_b}{B_b - TB_A} \right) \left(\frac{\epsilon - \mu TB_A + \phi_b}{T} \right) \right] \right. \\ \left. - \exp \left[- \left(\frac{B_{MAO}}{B_b - TB_A} \right) \left(\frac{\epsilon - \mu TB_A + \phi_b}{T} \right) \right] \right\} ,$$

where the parameter T is equal to one in the Cohen model, although it can in general differ from β . The model electron DF is constant in regions III and IV. The functions must match each other along their mutual boundary and must match the functions in regions I, II, and V at the point (μ^*, ϵ^*) :

$$f_{III} = f_{IV} = f_I(\mu^*, \epsilon^*) = f_{II}(\mu^*, \epsilon^*) = f_V(\mu^*, \epsilon^*) ,$$

so that

$$f_{III} = f_{IV} = C \left[\exp(-\epsilon^*/T) - \exp(-\mu^* B_{MAO}/T) \right] , \\ = C \left\{ \exp \left[\frac{\phi_{MXO}}{T} - \left(\frac{B_{MXO}}{B_{MXO} - B_b} \right) \left(\frac{\phi_{MXO} - \phi_b}{T} \right) \right] \right. \\ \left. - \exp \left[- \left(\frac{B_{MAO}}{B_{MXO} - B_b} \right) \left(\frac{\phi_{MI} - \phi_b}{T} \right) \right] \right\} .$$

Figure 4 is a phase space diagram for ions at point b. Note that I include the small triangle near the ϵ -intercept of the MXO line in the passing region. However, these ions are not passing in the strict sense: they cannot exist in the central cell because they are below its local $V_{||} = 0$ line. These ions must be scattered into this region of phase space or born outside the central cell. Physically, they are trapped between roughly point A and the central cell; they can actually make it over the throttle coil but not into the center of the machine. I treat these as passing ions because, while it is fairly straightforward to concoct a trapped-ion distribution function from the Cohen model that matches the passing DF on the line $V_{||}^{MXO} = 0$, a reasonable trapped-ion DF that matches the passing DF on a two-sided boundary such as JPL in Fig. 4 has not been constructed.

The passing ion DF is given by

$$f_{PASS} = C \{ \exp(-\epsilon/T) - \exp[-(\mu B_A + \phi_A)/T] \} ,$$

where, as with the passing electron DF, the second term allows the function to go to zero continuously on the loss boundary. The trapped and passing ion DFs must match along the line $V_{||}^{MXO} = 0$. I use Boghosian's modified Cohen DF for the trapped ions,⁹

$$f_{TRAP} = C \left\{ \exp \left[\frac{\epsilon - \alpha (\mu B_{MXO} + \phi_{MXO})}{(\alpha - 1)T} \right] - \exp \left[- \frac{\mu B_A + \phi_A}{T} - \frac{\alpha (\epsilon - \mu B_{MXO} - \phi_{MXO})}{(\alpha - 1)T} \right] \right\} .$$

I use Boghosian's improved sloshing-ion model DF to calculate the sloshing-ion density at point A, given the density at b.¹⁰ In this model Boghosian assumes that the DF goes to zero continuously on the lines $\epsilon = c\mu$ and $\epsilon = d\mu$ (see Figs. A-4 and A-6 in Appendix A). The DF is

$$f_{SLOSH} = C \epsilon^{\lambda} \left(\mu - \frac{\epsilon}{c} \right) \left(\frac{\epsilon}{d} - \mu \right) \exp(-\epsilon/T) ,$$

where λ is a free parameter.

Parameter c must be set equal to B_{MAO} in this model for the following reason. Assume that the upper boundary of the striped region in Fig. A-6a is the line $V_{||}^{MAO} = 0$. Because I assume that the trapped-ion DF goes to zero on the line $V_{||}^A = 0$, no trapped ions can be in the striped region. If I assume

Fig. 4. Phase space diagram for ions at point b. (Sloshing ions are omitted.) The points J, P, and L are referred to in the text.

that c is less than B_{MAO} (it cannot be greater because anything above MAO and A is lost), the heavily striped region - evaluated between A and MAO - will be finite, will contain no ions, but will contain electrons. Hence, although a solution can be found in this case for the potentials at the cardinal points inboard of MAO, quasi-neutrality cannot be solved between A and MAO unless $c = MAO$.³

Calculation of Densities

The densities are obtained by integrating the DFs over the appropriate regions of phase space,

$$n_j = \frac{4\pi B}{\sqrt{2} m^{3/2}} \int d\epsilon \int d\mu \frac{f_j(\epsilon, \mu)}{(\epsilon - \mu B - q_j \phi)^{1/2}} .$$

In Appendix A, I give phase diagrams for each cardinal point. The regions of integration are indicated for each class of charged particle. In Appendix B, I list the expressions for the densities.

The passing densities reduce to sums of the function S, given by

$$S(A, B, C, D, x_1, x_2) \equiv \text{Re} \int_{x_1}^{x_2} e^{-X} (AX + B) (CX + D)^{1/2} dX .$$

The expressions for the trapped-ion densities involve the function T, given by

$$T(A, B, C, D, x_1, x_2) \equiv \text{Re} \int_{x_1}^{x_2} e^{-X} \frac{(AX + B)}{(CX + D)^{1/2}} dX ,$$

and the expression for the sloshing-ion density involves the function U, defined as

$$U(L, A, B) = \text{Re} \int_0^\infty e^{-X} X^L (AX - B)^{1/2} dX .$$

The hot- and warm-electron densities are functions of S, T, or both, depending on the local value of B. The evaluations of S, T, and U involve the complex incomplete gamma function, which in turn involves the complementary error function and Dawson's integral.³

RESULTS

Some preliminary findings are presented in Appendix D. I used results from Jong's equilibrium code as a guide in setting input parameters and in interpreting output.¹¹ Using his results to set the potential estimates and input densities, I was quickly able to get reasonable values for the potentials at X, MX0, MAI, and A. However, it took much manipulating with input parameters to get the potential at b above ground. These variations in parameters are described in the following as well as their effect on the results, particularly on the potential at b.

In varying enc0 and ench0, the variable code names for central-cell warm- and hot-ion densities, I used the central-cell pressure balance equation,

$$T_{IW}n_{IW} + T_{IH}n_{IH} + (n_{IW} + n_{IH}) T_e = \beta_{CC} (2.5 \times 10^{15}) \beta_{CC}^2,$$

where the subscripts IW, IH, and e refer, respectively, to warm ions, hot ions, and electrons; β_{CC} is the central-cell beta; density is per cubic centimeter; and temperatures are given in kiloelectron volts.¹² I set $T_e = 9$ keV and $T_{IW} = 15$ keV and varied β_{CC} and $R \equiv n_{IH}/n_{IW}$. I then obtained n_{IW} from

$$n_{IW} = \frac{(2.5 \times 10^{15}) \beta_{CC} \beta_{CC}^2}{15 + R T_{IH} + 9 (1 + R)}.$$

When everything else is held constant, the effect of lowering β_{CC} from 0.4 to 0.2 is to raise the potentials at b, A, MAI, and X by 2 to 5 kV. Increasing R has the same effect; an increase in R from 0.15 to 0.32 results in a 2-kV increase in ϕ_b . Despite these effects, I was not able to get ϕ_b above about 8 to 10 kV below ground by varying enc0 and ench0 alone.

Raising the n_{IH} and n_{IW} densities of trapped- and sloshing-ions at b also raised ϕ_b . The more noticeable effect of raising the sloshing-ion density was to increase the potential at A. By just varying the sloshing density, I was not able to raise ϕ_b above ground without pushing ϕ_A to around 180 to 200 kV. Raising the trapped-ion density as well did not produce a positive ϕ_b unless unreasonable numbers were used (e.g., a trapped-ion density of 10^{14} cm^{-3}).

The parameter bsb0 is the slope of the lower boundary line in $(\epsilon - \mu)$ space for the integral of the sloshing-ion DF. The sloshing density at b is

input and the sloshing density at A is mapped from point b. The density and potential at A is highest for $bsb0 \approx 1.45$. Values less than one generally produce a potential at A that is lower than the central cell potential.

Changing beta was effective in producing a positive ϕ_b and a potential profile that agrees well with Jong's results.¹⁰ Recall that beta is the parameter in the Cohen hot-electron-model DF that determines the slope of the characteristics in $(\epsilon - \mu)$ space. I changed beta from 1.0 to $B_b/B_A = 0.71$. This change is equivalent to assuming that ECRH dominates at b rather than at A.

CONCLUSIONS

Three characteristics of PSOLV are important in determining its useability as a subroutine in time-dependent codes. The most basic consideration is the correctness of the electron model DF that is used. Comparison of QN05 results with early TMX-U data indicate that modifications in the model DF are necessary. Also, the code has an important limitation that may affect its correctness. Unlike Boghosian's code for TMX-U,³ PSOLV only solves quasi-neutrality at the cardinal points. It is therefore possible to find a solution for potentials for which the cardinal points cannot be connected with lines of quasi-neutrality in $(\phi - B)$ space. In other words, a solution of the six-variable system in PSOLV does not necessarily represent a solution for the potential profile.³ Another limitation to the code is that it qualitatively assumes the equilibrium form for the potential profile, i.e., $\phi_A > \phi_X > \phi_e > \phi_{MAI} > \phi_b$. This would probably create a problem if the code were used very near time $t = 0$, because the model is not valid for a perfectly flat profile.

The second consideration in determining the useability of PSOLV is its reliability of convergence. Under most conditions, the code converges within 100 iterations in COSNAF. The vast majority of convergence problems involve alpha, the sixth variable. Alpha is the pumping parameter in the Cohen trapped-ion DF; it takes on values from one (complete pumping) to infinity, down to negative infinity, and up to zero (no pumping). It is therefore possible for the initial estimate for alpha to be on the wrong side of the origin. In this case, the code will not converge, but alpha will get further away from the origin, reaching values on the order of 10^3 in 100 iterations.

In the great majority of the solutions encountered, $|\alpha|$ has been less than twenty. I have never found a solution with $|\alpha|$ greater than 100. It would, therefore, seem appropriate to include a check in the calling program such that, if PSOLV does not converge on a solution in the allowed time, and $|\alpha| \geq 100$, α is set to $-2.0 \cdot \text{sign}(\alpha)$ and PSOLV is called again. This should virtually eliminate convergence problems.

The third consideration is speed. In a start-up code, the time steps would presumably be chosen small enough so that the densities would not change too drastically in one time step. In such a case, preliminary results indicate that PSOLV uses an average of less than 0.01 minutes of CPU time on the Cray computer. It seems reasonable to conclude from this that one cannot expect to call PSOLV more than about 10^3 to 10^4 times in a start-up code. This is more than adequate for use in TREQ.²

ACKNOWLEDGMENTS

Bruce Boghosian and Mike Gilmore contributed a great deal to this work. Most of the theory outlined in this report was developed previously by Bruce for TMX-U; in many cases I was able to make substitutions into his results to arrive at density expressions for MFTF-B. Mike developed a truncated potential solver for TMX-U that I used as a starting point in writing PSOLV. He was also a partner in the debugging process.

I also thank Ray Jong and Gary Porter for their time and assistance.

APPENDIX A

PHASE SPACE DIAGRAMS FOR THE CHARGED-PARTICLE CLASSES PRESENT IN MFTF-B.

In the following phase diagrams, Figs. A-1 through A-6, the lines labeled by point names are the $V_{||} = 0$ lines for that point. The slopes of the lines equal the local B-field, and the intercepts equal either plus (for ion diagrams) or minus (for electron diagrams) the local potential. For example, in an electron diagram, the equation of the line labeled MX0 is

$$\epsilon = \mu B_{MX0} - \phi_{MX0}.$$

Fig. A-1. Phase space diagrams for ions (a) and electrons (b) at point X. Note that the electrons are assumed to "fill in" the small triangle under the cc line, i.e., electrons are scattered into this region of phase space.

Fig. A-2. Phase space diagram for ions (a) and electrons (b) at point MXO.

Fig. A-3. Phase space diagram for ions (a) and electrons (b) at point MAI. In (b), the transiting electrons are assumed rf-dominated. If they are treated as collision-dominated, the passing region fills the entire JLP triangle.

Fig. A-4. Phase space diagram for ions at point b.

Fig. A-5. Phase space diagram for electrons at point b, assuming in (a) that the transiting electrons are rf-dominated and in (b) that the transiting electrons are Maxwellian.

Fig. A-6. Phase space diagram for ions (a) and for electrons (b) at point A. In (b), the transiting electrons are assumed to be Maxwellian.

APPENDIX B EXPRESSIONS FOR THE DENSITIES

INTRODUCTION

The densities are found by integrating the model distribution functions (DFs) over the regions indicated in Appendix A:

$$n = \frac{4\pi B}{\sqrt{2} m^{3/2}} \int d\epsilon \int d\mu \frac{f(\epsilon, \mu)}{(\epsilon - \mu B - \phi)^{1/2}} ,$$

where B is the local field, ϕ is the local potential, $f(\epsilon, \mu)$ is the DF, and m is the mass of the particle concerned. The model distribution functions for the various classes of charged particles are given in the body of this report. The following expressions are valid only for $B_{MXO} > B_X > B_{MAI} > B_{MAO} > B_A > B_b$ and $\phi_A > \phi_X > \phi_e > \phi_{MXO} > \phi_{MAI} > \phi_b$. Most of the densities turn out to be functions of S , T , or U , defined as follows:

$$S(A, B, C, D, x_1, x_2) \equiv \text{Re} \int_{x_1}^{x_2} e^{-X} (AX + B) (CX + D)^{1/2} dX$$

$$T(A, B, C, D, x_1, x_2) \equiv \text{Re} \int_{x_1}^{x_2} e^{-X} \frac{(AX + B)}{(CX + D)^{1/2}} dX$$

$$U(\ell, A, B) = \text{Re} \int_0^\infty e^{-X} X^\ell (AX - B)^{1/2} dX .$$

See Ref. 13 for an algorithm for calculating S , T , and U in terms of Dawson's integral, gamma functions, and the complementary error function.

The following are the expressions for the densities. In all expressions, n_0 is the central-cell density, T_e is the central-cell electron temperature, T_i is the central-cell ion temperature, and T_{SB} is the temperature of the sloshing beam. Unless otherwise noted, all substitutions include implicit expressions, e.g., if the substitution $B_{MXO} \rightarrow B_{MAI}$ is made in a density expression it should also be made in the definitions of all symbols in the expression.

DENSITIES AT X

Passing-Ion Density at X

$$\begin{aligned}
 n = n_0 & \left\{ S \left(0, 1, 1, \frac{-\phi_x}{T_i}, \frac{\phi_x}{T_i}, \frac{\phi_A}{T_i} \right) \right. \\
 & + \left(1 - \frac{B_x}{B_A} \right) S \left[0, 1, \left(1 - \frac{B_x}{B_A} \right), \left(\frac{\phi_A B_x}{T_i B_A} - \frac{\phi_x}{T_i} \right), \frac{\phi_A}{T_i}, \frac{\epsilon^*}{T_i} \right] \Big\} \\
 & \cdot \left\{ S \left[0, 1, 1, \frac{-\phi_e}{T_i}, \frac{\phi_e}{T_i}, \frac{\phi_A}{T_i} \right] + \left(1 - \frac{B_{cc}}{B_A} \right) \right. \\
 & \left. S \left[0, 1, \left(1 - \frac{B_{cc}}{B_A} \right) \left(\frac{\phi_A B_{cc}}{T_i B_A} - \frac{\phi_e}{T_i} \right), \frac{\tilde{\epsilon}_c}{T_i} \right] \right\}^{-1},
 \end{aligned}$$

where

$$\epsilon^* \equiv \phi_x + B_x \left(\frac{\phi_A - \phi_x}{B_x - B_A} \right),$$

and

$$\tilde{\epsilon}_c \equiv \begin{cases} \epsilon_c^* & \text{if } B_{cc} > B_A \\ \infty & \text{if } B_{cc} < B_A \end{cases},$$

where

$$\epsilon_c^* \equiv \phi_e + B_{cc} \left(\frac{\phi_A - \phi_e}{B_{cc} - B_A} \right).$$

B_{cc} is the central-cell field, ϕ_e is the central-cell potential, and the $\{ \}^{-1}$ factor is the ion normalization constant.

Passing-Electron Density at X

$$\begin{aligned}
 n = n_0 & \left\{ S \left(0, 1, 1, \frac{\phi_x}{T_e}, \frac{-\phi_x}{T_e}, 0 \right) + \left(1 - \frac{B_x}{B_{MAO}} \right) \right. \\
 & \cdot S \left[0, 1, \left(1 - \frac{B_x}{B_{MAO}} \right), \frac{\phi_x}{T_e}, 0, \frac{\tilde{\epsilon}}{T_e} \right] \Big\} \\
 & \cdot \left\{ S \left(0, 1, 1, \frac{\phi_e}{T_e}, \frac{-\phi_e}{T_e}, 0 \right) \right. \\
 & \left. + \left(1 - \frac{B_{cc}}{B_{MAO}} \right) S \left[0, 1, \left(1 - \frac{B_{cc}}{B_{MAO}} \right), \frac{\phi_e}{T_e}, 0, \epsilon_c \right] \right\}^{-1},
 \end{aligned}$$

where the $\{ \}^{-1}$ factor is the electron normalization constant,

$$\tilde{\epsilon} = \begin{cases} B_{MAO} \left(\frac{\phi_{MAO} - \phi_x}{B_{MAO} - B_x} \right) - \phi_{MAO} & \text{if } B_x > B_{MAO} \\ \infty & \text{if } B_x < B_{MAO} \end{cases}$$

and $\tilde{\epsilon}_c$ is defined as $\tilde{\epsilon}$ with the substitutions

$$B_x \rightarrow B_{cc}$$

$$\phi_x \rightarrow \phi_e$$

DENSITIES AT MXO

Passing-Ion Density at MXO

This expression is the same as that for the passing ions at X with the following substitutions:

$$\phi_x \rightarrow \phi_{MXO}$$

$$B_x \rightarrow B_{MXO}$$

Passing-Electron Density at MXO

This expression is the same as that for the passing electrons at X with the following substitutions:

$$\phi_X \rightarrow \phi_{MXO}$$

$$B_X \rightarrow B_{MXO}$$

DENSITIES AT MAI

Passing-Ion Density at MAI

$$\begin{aligned} n = n_0 \bigg\{ & S \left(0, 1, 1, \frac{-\phi_{MAI}}{T_i}, \frac{\phi_{MXO}}{T_i}, \frac{\phi_A}{T_i} \right) \\ & - S \left[0, 1, \left(1 - \frac{B_{MAI}}{B_{MXO}} \right), \left(\frac{\phi_{MXO} B_{MAI}}{T_i B_{MXO}} - \frac{\phi_{MAI}}{T_i} \right), \frac{\phi_{MXO}}{T_i}, \frac{\epsilon^*}{T_i} \right] \\ & + \left(1 - \frac{B_{MAI}}{B_A} \right) \cdot S \left[0, 1, \left(1 - \frac{B_{MAI}}{B_A} \right), \left(\frac{\phi_A B_{MAI}}{T_i B_A} - \frac{\phi_{MAI}}{T_i} \right), \frac{\phi_A}{T_i}, \frac{\epsilon^*}{T_i} \right] \\ & + \frac{B_{MAI}}{B_A} \cdot S \left[0, 1, \left(\frac{B_{MXO} - B_{MAI}}{B_A} \right), \left(\frac{-\phi_A (B_{MXO} - B_{MAI})}{T_i B_A} \right. \right. \\ & \left. \left. + \frac{\phi_{MXO}}{T_i} - \frac{\phi_{MAI}}{T_i} \right), \frac{\phi_A}{T_i}, \frac{\epsilon^*}{T_i} \right] \bigg\} \cdot (\text{ion normalization constant}) , \end{aligned}$$

where

$$\epsilon^* \equiv \phi_{MXO} + B_{MXO} \left(\frac{\phi_{MXO} - \phi_A}{B_A - B_{MXO}} \right) .$$

Transiting-Ion Density at MAI

$$n = \frac{n_0 B_{MAI}}{4 T_i^{3/2}} \left\{ \frac{1}{C_2} \left[e^{-d_7} T \left(0, \frac{1}{d_1}, \frac{1}{d_1}, 0, 0, d_1 \epsilon^{***} \right) \right] \right\}$$

$$\begin{aligned}
& + e^{-d_8} T \left(0, \frac{1}{d_2}, \frac{1}{d_2}, 0, d_2 \epsilon'^{**}, d_2 \epsilon'^{*} \right) \\
& - e^{-d_9} T \left(0, \frac{1}{d_3}, \frac{1}{d_3}, 0, 0, d_3 \epsilon'^{*} \right) \Big] \\
& - \frac{1}{C_4} \left[e^{-d_{10}} T \left(0, \frac{1}{d_4}, \frac{1}{d_4}, 0, 0, d_4 \epsilon'^{**} \right) \right. \\
& + e^{-d_{11}} T \left(0, \frac{1}{d_5}, \frac{1}{d_5}, 0, d_5 \epsilon'^{**}, d_5 \epsilon'^{*} \right) \\
& \left. - e^{-d_{12}} T \left(0, \frac{1}{d_6}, \frac{1}{d_6}, 0, 0, d_6 \epsilon'^{*} \right) \right] \Big\} \\
& \cdot (\text{ion normalization constant}) .
\end{aligned}$$

where

$$\begin{aligned}
d_1 & \equiv C_1 + \frac{C_2}{B_{MAI}} , \\
d_2 & \equiv C_1 + \frac{C_2}{B_{MAI}} \left(\frac{B_{MXO} + B_{MAI}}{B_{MXO} - B_{MAI}} \right) , \\
d_3 & \equiv C_1 + \frac{C_2}{B_{MAI}} \left(\frac{B_A + B_{MAI}}{B_A - B_{MAI}} \right) , \\
d_4 & \equiv C_3 + \frac{C_4}{B_{MAI}} , \\
d_5 & \equiv C_3 + \frac{C_4}{B_{MAI}} \left(\frac{B_{MXO} + B_{MAI}}{B_{MXO} - B_{MAI}} \right) , \\
d_6 & \equiv C_3 + \frac{C_4}{B_{MAI}} \left(\frac{B_A + B_{MAI}}{B_A - B_{MAI}} \right) ,
\end{aligned}$$

$$d_7 \equiv \frac{\alpha \phi_{MXO} - \phi_{MAI}}{(\alpha - 1) T_i} ,$$

$$d_8 \equiv \frac{\alpha \phi_{MXO} - \phi_{MAI}}{(\alpha - 1) T_i} + \left(\frac{2C_2}{B_{MXO} - B_{MAI}} \right) (\phi_{MAI} - \phi_{MXO}) ,$$

$$d_9 \equiv \frac{\alpha \phi_{MXO} - \phi_{MAI}}{(\alpha - 1) T_i} + \left(\frac{2C_2}{B_A - B_{MAI}} \right) (\phi_{MAI} - \phi_A) ,$$

$$d_{10} \equiv \frac{(\alpha - 1) \phi_A + \alpha (\phi_{MXO} - \phi_{MAI})}{(\alpha - 1) T_i} ,$$

$$d_{11} \equiv \frac{(\alpha - 1) \phi_A + \alpha (\phi_{MXO} - \phi_{MAI})}{(\alpha - 1) T_i} + \left(\frac{2C_4}{B_{MXO} - B_{MAI}} \right) (\phi_{MAI} - \phi_{MXO}) ,$$

$$d_{12} \equiv \frac{(\alpha - 1) \phi_A + \alpha (\phi_{MXO} - \phi_{MAI})}{(\alpha - 1) T_i} + \left(\frac{2C_4}{B_A - B_{MAI}} \right) (\phi_{MAI} - \phi_A) ,$$

$$c_1 \equiv - \frac{1 + \alpha \frac{B_{MXO}}{B_{MAI}}}{2 (\alpha - 1) T_i} ,$$

$$c_2 \equiv - \frac{\left(1 - \alpha \frac{B_{MXO}}{B_{MAI}} \right) B_{MAI}}{2 (\alpha - 1) T_i} ,$$

$$c_3 \equiv - \frac{\left[\alpha + (\alpha - 1) \frac{B_A}{B_{MAI}} + \alpha \frac{B_{MXO}}{B_{MAI}} \right]}{2 (\alpha - 1) T_i} ,$$

$$c_4 \equiv - \frac{\left[\alpha - (\alpha - 1) \frac{B_A}{B_{MAI}} - \alpha \frac{B_{MXO}}{B_{MAI}} \right] B_{MAI}}{2 (\alpha - 1) T_i} ,$$

$$\epsilon^{***} \equiv \phi_{MXO} - \phi_{MAI} ,$$

$$\epsilon^{*} \equiv (\phi_A - \phi_{MAI}) + \left(\frac{B_A - B_{MAI}}{B_{MXO} + B_{MAI}} \right) (\phi_{MAI} - \phi_{MXO}) .$$

Passing-Electron Density at MAI, Assuming rf Dominated Transiting Electrons

$$\begin{aligned} n = n_0 & \left\{ S \left(0, 1, 1, \frac{\phi_{MAI}}{T_e}, \frac{-\phi_{MAI}}{T_e}, 0 \right) \right. \\ & - S \left[0, 1, \left(1 - \frac{B_{MAI}}{B_{MXO}} \right), \left(\frac{\phi_{MAI}}{T_e} - \frac{B_{MAI} \phi_{MXO}}{B_{MXO} T_e} \right), \frac{\tilde{\epsilon}}{T_e}, \frac{\epsilon^*}{T_e} \right] \\ & + S \left[0, \left(1 - \frac{B_{MAI}}{B_{MAO}} \right), \left(1 - \frac{B_{MAI}}{B_{MAO}} \right), \frac{\phi_{MAI}}{T_e}, 0, \frac{\epsilon^*}{T_e} \right] \\ & \left. + \frac{B_{MAI}}{B_{MAO}} S \left[0, 1, \left(\frac{B_{MXO} - B_{MAI}}{B_{MAO}} \right), \left(\frac{\phi_{MAI} - \phi_{MXO}}{T_e} \right), \frac{\tilde{\mu} B_{MAO}}{T_e}, \frac{\epsilon^*}{T_e} \right] \right\} \\ & \cdot (\text{electron normalization constant}) , \end{aligned}$$

where

$$\epsilon^* \equiv \left(\frac{B_{MAO}}{B_{MXO} - B_{MAO}} \right) \phi_{MXO} ,$$

$$\tilde{\mu} \equiv \left(\frac{\phi_{MXO} - \phi_{MAI}}{B_{MXO} - B_{MAI}} \right) ,$$

$$\tilde{\epsilon} \equiv B_{MAI} \tilde{\mu} - \phi_{MAI} .$$

Passing-Electron Density at MAI, Assuming Collision-Dominated Transiting Electrons

$$n = n_0 \left\{ S \left(0, 1, 1, \frac{\phi_{MAI}}{T_e}, \frac{-\phi_{MAI}}{T_e}, 0 \right) \right.$$

$$+ S \left[0, \left(1 - \frac{B_{MAI}}{B_{MAO}} \right), \left(1 - \frac{B_{MAI}}{B_{MAO}} \right), \frac{\phi_{MAI}}{T_e}, 0, \frac{\tilde{\epsilon}}{T_e} \right] \}$$

• (electron normalization constant) ,

where

$$\tilde{\epsilon} \equiv \left(\frac{B_{MAO}}{B_{MAI} - B_{MAO}} \right) \phi_{MAI} .$$

Region II Hot-Electron Density at MAI, Assuming rf-Dominated Transiting Electrons

$$n = n_0 \left(\frac{B_{MAI}}{BB_A - B_{MAI}} \right) \cdot \sum_{j=1}^2 \frac{a_j}{r_j} \exp \left[- \frac{r_j}{T_e} (\epsilon_1 - \beta B_A \mu_1) \right] \\ \cdot \left\{ S \left[0, 1, \frac{B_{MAI}}{BB_A r_j}, \left(\frac{B_{MAI}}{BB_A T_e} \epsilon_1 + \frac{(BB_A - B_{MAI})}{BB_A T_e} \epsilon_c - \frac{\mu_1 B_{MAI}}{T_e} + \frac{\phi_{MAI}}{T_e} \right), \right. \right. \\ \left. \left. 0, \frac{r_j}{T_e} (B_{MXO} - \beta B_A) (\mu_4 - \mu_1) \right] - S \left[0, 1, \frac{1}{r_j} \left(\frac{B_{MXO} - B_{MAI}}{B_{MXO} - BB_A} \right), \right. \right. \\ \left. \left. \left(\frac{\epsilon_1 - \mu_1 B_{MAI} + \phi_{MAI}}{T_e} \right), 0, \frac{r_j}{T_e} (B_{MXO} - \beta B_A) (\mu_4 - \mu_1) \right] \right\}$$

• (electron normalization constant) ,

where

$$a_1 \equiv \exp \left[\frac{\phi_{MXO}}{T_e} \left(\frac{\beta B_A}{BB_A - B_{MXO}} \right) \right] ,$$

$$a_2 \equiv - \exp \left[\frac{\phi_{MXO}}{T_e} \left(\frac{B_{MAO}}{BB_A - B_{MAO}} \right) \right] ,$$

$$r_1 \equiv \frac{B_{MXO}}{B_{MXO} - \beta B_A} ,$$

$$r_2 \equiv \frac{B_{MAO}}{B_{MAO} - B_A} ,$$

$$\mu_1 \equiv \frac{\phi_{MXO} - \phi_{MAI}}{B_{MXO} - B_{MAI}} ,$$

$$\epsilon_1 \equiv \mu_1 B_{MXO} - \phi_{MXO} ,$$

$$\mu_4 \equiv \frac{\phi_{MXO}}{B_{MXO} - B_{MAO}} ,$$

B is the parameter in the Cohen model hot-electron DF, and ϵ_c is the maximum energy the hot electrons can attain. (See Ref. 14 for a discussion of this energy limit.)

There are no region-III hot electrons at MAI.

DENSITIES AT b

Passing-Ion Density at b

This expression is the same as that for the passing ions at MAI except for the following substitutions.

$$\phi_{MAI} \rightarrow \phi_b ,$$

$$B_{MAI} \rightarrow B_b .$$

Trapped-Ion Density at b

$$\begin{aligned} n = & \frac{n_0 B_b}{4 T_1^{3/2}} \left\{ \frac{1}{c_2} \left[e^{-d_7} T \left(0, \frac{1}{d_1}, \frac{1}{c_1}, 0, 0, d_1 \epsilon'^{**} \right) \right. \right. \\ & + e^{-d_8} T \left(0, \frac{1}{d_2}, \frac{1}{d_2}, 0, d_2 \epsilon'^{**}, d_2 \epsilon'^{*} \right) \\ & \left. \left. + e^{-d_9} T \left(0, \frac{1}{d_3}, \frac{1}{d_3}, 0, d_3 \epsilon'^{*}, \infty \right) \right] \right\} \end{aligned}$$

$$\begin{aligned}
& - \frac{1}{c_4} \left[e^{-d_{10}} T \left(0, \frac{1}{d_4}, \frac{1}{d_4}, 0, 0, d_4 \epsilon'^{**} \right) \right. \\
& + e^{-d_{11}} T \left(0, \frac{1}{d_5}, \frac{1}{d_5}, 0, d_5 \epsilon'^{**}, d_5 \epsilon'^{*} \right) \\
& \left. + e^{-d_{12}} T \left(0, \frac{1}{d_6}, \frac{1}{d_6}, 0, d_6 \epsilon'^{*}, \infty \right) \right] \}
\end{aligned}$$

•(ion normalization constant) ,

where the d's, c's, ϵ'^{*} , and ϵ'^{**} are defined as in the trapped-ion expression for point MAI except for the following substitutions:

$$\phi_{MAI} \rightarrow \phi_b ,$$

$$B_{MAI} \rightarrow B_b .$$

The sloshing-ion density at b is input.

Passing-Electron Density at b, Assuming rf-Dominated the Transiting Electrons

This expression is the same as the corresponding one for MAI except for the following substitutions.

$$\phi_{MAI} \rightarrow \phi_b ,$$

$$B_{MAI} \rightarrow B_b .$$

Region II Hot-Electron Densities at b, Assuming rf-Dominated Transiting Electrons

If $\beta \neq B_b/B_A$, the expression is the same as the corresponding one for MAI except for the following substitutions:

$$\phi_{MAI} \rightarrow \phi_b ,$$

$$B_{MAI} \rightarrow B_b .$$

If $\beta = B_b/B_A$,

$$n = \frac{1}{2} n_0 \sum_{j=1}^2 \frac{a_j}{r_j} \exp \left[-\frac{r_j}{T_e} (\epsilon_1 - \beta B_A \mu_1) \right] \\ \cdot T \left[\frac{-1}{r_j} \left(\frac{B_{MXO}}{B_{MXO} - \beta B_A} \right), \frac{\epsilon_c - \epsilon_1}{T_e}, \frac{1}{r_j}, \left(\frac{\epsilon_1 - \mu_1 B_b + \phi_b}{T_e} \right), 0, \right. \\ \left. \frac{r_j}{T_e} (B_{MXO} - \beta B_A) (\mu_4 - \mu_1) \right] \cdot (\text{electron normalization constant}),$$

where a_j , r_j , μ_1 , μ_4 , ϵ_1 , and ϵ_c are defined as in the corresponding expression for MAI except for the following substitutions.

$$\phi_{MAI} \rightarrow \phi_b,$$

$$B_{MAI} \rightarrow B_b.$$

Region III Hot-Electron Density at b , Assuming rf-Dominated Transiting Electrons

$$n = \frac{2}{3} n_0 \left\{ \exp \left[\frac{\phi_{MXO}}{T_e} - \left(\frac{B_{MXO}}{B_{MXO} - B_b} \right) \left(\frac{\phi_{MXO} - \phi_b}{T_e} \right) \right] \right. \\ \left. - \exp \left[\left(\frac{B_{MAO}}{B_b - B_{MXO}} \right) \left(\frac{\phi_{MXO} - \phi_b}{T_e} \right) \right] \right\} \cdot \left\{ \left(\frac{\beta B_A}{\beta B_A - B_b} \right) \left[\left(1 - \frac{B_b}{\beta B_A} \right) \frac{\epsilon_c}{T_e} \right. \right. \\ \left. \left. + \left(\frac{B_b}{\beta B_A} \frac{\epsilon_1}{T_e} - \frac{\mu_1 B_b}{T_e} + \frac{\phi_b}{T_e} \right) \right]^{3/2} - \left(\frac{\beta B_A}{\beta B_A - B_b} \right) \left(\frac{\epsilon_1 - \mu_1 B_b + \phi_b}{T_e} \right)^{3/2} \right\} \\ \cdot (\text{electron normalization constant}),$$

where

$$\mu_1 \equiv \left(\frac{\phi_{MXO} - \phi_b}{B_{MXO} - B_b} \right)$$

and

$$\epsilon_1 \equiv \mu_1 B_b - \phi_b.$$

Passing- and Hot-Electron Densities at b, Assuming Collision-Dominated
Transiting Electrons

The expressions are the same as those for transiting electrons at b that are rf-dominated, with the following substitutions.

$$\phi_{MXO} \rightarrow \phi_{MAI}$$

$$B_{MXO} \rightarrow B_{MAI}$$

DENSITIES AT A

Sloshing Ion Density at A

$$\begin{aligned} n = n_{sb} \left(\frac{B_b^2}{B_A^2} \right) & \left[\xi_1 (\xi_1 - 5\xi_2) U(\ell + 2, \xi_1, \eta) \right. \\ & + (3\xi_1 + 5\xi_2) \eta U(\ell + 1, \xi_1, \eta) - 4\eta^2 U(\ell, \xi_1, \eta) \\ & - \xi_2 (\xi_2 - 5\xi_1) U(\ell + 2, \xi_2, \eta) - (3\xi_2 + 5\xi_1) \eta U(\ell + 1, \xi_2, \eta) \\ & + 4\eta^2 U(\ell, \xi_2, \eta) \left. \right] \cdot \left[\xi_{10} (\xi_{10} - 5\xi_{20}) U(\ell + 2, \xi_{10}, \eta_0) \right. \\ & + (3\xi_{10} + 5\xi_{20}) \eta_0 U(\ell + 1, \xi_{10}, \eta_0) - 4\eta_0^2 U(\ell, \xi_{10}, \eta_0) \\ & - \xi_{20} (\xi_{20} - 5\xi_{10}) U(\ell + 2, \xi_{20}, \eta_0) \\ & \left. - (3\xi_{20} + 5\xi_{10}) \eta_0 U(\ell + 1, \xi_{20}, \eta_0) + 4\eta_0^2 U(\ell, \xi_{20}, \eta_0) \right] , \end{aligned}$$

where

$$\xi_{10} \equiv 1 - \frac{B_b}{c} ,$$

$$\xi_{20} \equiv 1 - \frac{B_b}{d} ,$$

$$n_0 \equiv \frac{\phi_b}{T_{sb}} ,$$

$$\xi_1 \equiv 1 - \frac{B_A}{c} ,$$

$$\xi_2 \equiv 1 - \frac{B_A}{d} ,$$

$$\eta \equiv \frac{\phi_A}{T_{sb}} .$$

The parameters c and d are the slopes of the lines at which the distribution function goes to zero, and ℓ is the free parameter in the DF.

Passing-Electron Density at A, Assuming rf-Dominated Transiting Electrons

$$\begin{aligned} n = n_0 & \left\{ S \left[0, 1, 1, \frac{\phi_A}{T_e}, \frac{-\phi_b}{T_e}, 0 \right] - S \left[0, 1, \left(1 - \frac{B_A}{B_b} \right), \right. \right. \\ & \cdot \left. \left(\frac{\phi_A}{T_e} - \frac{B_A \phi_b}{B_b T_e} \right), \frac{-\phi_b}{T_e}, \frac{\tilde{\xi}}{T_e} \right] - S \left[0, 1, \left(1 - \frac{B_A}{B_{MX0}} \right), \right. \\ & \left. \left(\frac{\phi_A}{T_e} - \frac{B_A \phi_{MX0}}{B_{MX0} T_e} \right), \frac{\tilde{\xi}}{T_e}, \frac{\xi^*}{T_e} \right] \\ & + \left(1 - \frac{B_A}{B_{MA0}} \right) S \left[0, 1, \left(1 - \frac{B_A}{B_{MA0}} \right), \frac{\phi_A}{T_e}, 0, \frac{\xi^*}{T_e} \right] \\ & + \frac{B_A}{B_{MA0}} S \left[0, 1, \left(\frac{B_b - B_A}{B_{MA0}} \right), \left(\frac{\phi_A - \phi_b}{T_e} \right), 0, \frac{\tilde{\xi} B_{MA0}}{T_e} \right] \end{aligned}$$

$$+ \frac{B_A}{B_{MAO}} S \left[0, 1, \frac{B_{MXO} - B_A}{B_{MAO}}, \frac{\phi_A - \phi_{MXO}}{T_e}, \frac{\mu B_{MAO}}{T_e}, \frac{\epsilon^*}{T_e} \right] \}$$

• (electron normalization constant) ,

where

$$\epsilon^* \equiv \left(\frac{B_{MAO}}{B_{MXO} - B_{MAO}} \right) \phi_{MXO} ,$$

$$\tilde{\mu} \equiv \left(\frac{\phi_{MXO} - \phi_b}{B_{MXO} - B_b} \right) ,$$

$$\tilde{\epsilon} \equiv B_b \tilde{\mu} - \phi_b .$$

Region II Hot-Electron Density at A, Assuming rf-Dominated Transiting Electrons

If $\beta = 1$ the expression is the same as the corresponding one for point b, except the explicit B_b in the fourth argument of T should be changed to B_A .

If $\beta \neq 1$, then

$$\begin{aligned} n = n_0 \left(\frac{B_A}{\beta B_A - B_A} \right) \cdot \sum_{j=1}^2 \frac{a_j}{r_j} \exp \left[\frac{r_j}{T_e} (\beta B_A \mu_1 - \epsilon_1) \right] \\ \cdot \left\{ S \left[0, 1, \frac{B_A}{\beta B_A r_j}, \left(\frac{B_A \epsilon_1}{\beta B_A T_e} + \frac{(\beta B_A - B_A) \epsilon_c}{\beta B_A T_e} - \frac{\mu_1 B_A}{T_e} + \frac{\phi_A}{T_e} \right), 0, \right. \right. \\ \left. \frac{r_j}{T_e} (B_{MXO} - \beta B_A) (\mu_4 - \mu_1) \right] - S \left[0, 1, \frac{1}{r_j} \left(\frac{B_{MXO} - B_A}{B_{MXO} - \beta B_A} \right), \right. \\ \left. \left(\frac{\epsilon_1 - \mu_1 B_A + \phi_A}{T_e} \right), 0, \frac{r_j}{T_e} (B_{MXO} - \beta B_A) (\mu_4 - \mu_1) \right] \} \end{aligned}$$

• (electron normalization constant) ,

where a_j , r_j , μ_1 , μ_4 , ϵ_c , and ϵ_1 are defined as in the expression for the region II hot electron density at b.

Region III Hot-Electron Density at A, Assuming rf-Dominated Transiting Electrons

If $\beta = B_b/B_A$,

$$n = \frac{2}{3} n_0 \left\{ \exp \left[\frac{\phi_{MXO}}{T_e} - \left(\frac{B_{MXO}}{B_{MXO} - B_b} \right) \left(\frac{\phi_{MXO} - \phi_b}{T_e} \right) \right] \right. \\ \left. - \exp \left[\left(\frac{B_{MAO}}{B_b - B_{MXO}} \right) \left(\frac{\phi_{MXO} - \phi_b}{T_e} \right) \right] \right\} \cdot \left\{ \left(\frac{\beta B_A}{\beta B_A - B_b} \right) \left[\left(1 - \frac{B_b}{\beta B_A} \right) \frac{\epsilon_c}{T_e} \right. \right. \\ \left. \left. + \left(\frac{B_b \epsilon_1}{\beta B_A T_e} - \frac{\mu_1 B_b}{T_e} + \frac{\phi_A}{T_e} \right) \right]^{3/2} - \left(\frac{\beta B_A}{\beta B_A - B_b} \right) \left(\frac{\epsilon_1 - \mu_1 B_b + \phi_A}{T_e} \right)^{3/2} \right. \\ \left. - \frac{3}{2} \left(\frac{\phi_A - \phi_b}{T_e} \right)^{1/2} \left(\frac{\epsilon_c - \epsilon_1}{T_e} \right) \right\} \\ \cdot (\text{electron normalization constant}) .$$

If $\beta = 1$,

$$n = \frac{2}{3} n_0 \left\{ \exp \left[\frac{\phi_{MXO}}{T_e} - \frac{B_{MXO}}{(B_{MXO} - B_b)} \left(\frac{\phi_{MXO} - \phi_b}{T_e} \right) \right] \right. \\ \left. - \exp \left[\left(\frac{B_{MAO}}{B_b - B_{MXO}} \right) \left(\frac{\phi_{MXO} - \phi_b}{T_e} \right) \right] \right\} \\ \cdot \left\{ \frac{3}{2} \left(\frac{\epsilon_1 - \beta \mu_1 B_A + \phi_A}{T_e} \right)^{1/2} \left(\frac{\epsilon_c - \epsilon_1}{T_e} \right) - \left(\frac{B_b}{B_b - \beta B_A} \right) \right. \\ \cdot \left[\left(1 - \frac{\beta B_A}{B_b} \right) \frac{\epsilon_c}{T_e} + \left(\frac{\beta B_A \epsilon_1}{B_b T_e} - \frac{\beta \mu_1 B_A}{T_e} + \frac{\phi_A}{T_e} \right) \right]^{3/2} \\ \left. + \left(\frac{B_b}{B_b - \beta B_A} \right) \left[\frac{\epsilon_1 - \beta \mu_1 B_A + \phi_A}{T_e} \right]^{3/2} \right\} \\ \cdot (\text{electron normalization constant}) ,$$

where μ_1 , ϵ_1 , and ϵ_c are defined as in the corresponding expression for point b.

Passing-Electron Density and Regions II and III Hot-Electron Densities at A
Assuming Collision-Dominated Transiting Electrons

The expressions are the same as those for rf-dominated transiting electrons except for the following substitutions:

$$\phi_{MXO} \rightarrow \phi_{MAI} ,$$

$$B_{MXO} \rightarrow B_{MAI} .$$

Region IV Warm-Electron Density at A, Assuming rf-Dominated Transiting Electrons

$$\begin{aligned} n = \frac{2}{3} n_0 \left\{ \frac{3}{2} \left(\frac{B_A}{B_{MXO} - B_b} \right) \left(\frac{\phi_{MXO} - \phi_b}{T_e} \right) \left(\frac{\phi_A}{T_e} - \tilde{\epsilon} \right)^{1/2} \right. \\ \left. - \left(\frac{B_A}{B_b - B_A} \right) \left[\left(\frac{B_b - B_A}{B_{MXO} - B_b} \right) \left(\frac{\phi_{MXO} - \phi_b}{T_e} \right) + \left(\frac{\phi_A - \phi_b}{T_e} \right) \right]^{3/2} \right\} \\ \cdot \left\{ \exp \left[\frac{\phi_{MXO}}{T_e} - \left(\frac{B_{MXO}}{B_{MXO} - B_b} \right) \left(\frac{\phi_{MXO} - \phi_b}{T_e} \right) \right] \right. \\ \left. - \exp \left[\left(\frac{B_{MAO}}{B_b - B_{MXO}} \right) \left(\frac{\phi_{MXO} - \phi_b}{T_e} \right) \right] \right\} \\ \cdot (\text{electron normalization constant}) , \end{aligned}$$

where

$$\tilde{\epsilon} \equiv \gamma B_A \mu^* - \epsilon^* ,$$

$$\mu^* \equiv \frac{\phi_{MXO} - \phi_b}{B_{MXO} - B_b} ,$$

$$\epsilon^* \equiv \mu^* B_{MXO} - \phi_{MXO} ,$$

and γ is assumed to be one.

Region V Warm-Electron Density at A Assuming rf-Dominated Transiting Electrons

Assuming $\gamma = 1$,

$$n = \frac{n_0}{2} \left(\frac{B_A}{B_A - B_b} \right) \sum_{j=1}^2 \frac{a_j}{r_j} \exp \left(\frac{r_j \tilde{\epsilon}}{r_e} \right) \\ \cdot \tau \left[-\frac{1}{r_j}, \frac{\tilde{\epsilon} - \phi_b}{r_e}, \frac{1}{r_j}, \frac{\phi_A - \tilde{\epsilon}}{r_e}, 0, \frac{r_j (\tilde{\epsilon} - \phi_b)}{r_e} \right]$$

• (electron normalization constant) ,

where $\tilde{\epsilon}$ is defined as in the region IV calculation and

$$a_1 \equiv \exp \left[\frac{\phi_b}{r_e} \left(\frac{B_A}{B_A - B_b} \right) \right],$$

$$a_2 \equiv - \exp \left[\frac{\phi_b}{r_e} \left(\frac{B_{MAO}}{B_A - B_b} \right) \right],$$

$$r_1 \equiv \frac{B_b}{B_b - B_A} ,$$

$$r_2 \equiv \frac{B_{MAO}}{B_b - B_A} .$$

Regions IV and V Warm-Electron Densities, Assuming Collision-Dominated Transiting Electrons

The expressions are the same as for rf-dominated case, except for the following substitutions:

$$\phi_{MXO} \rightarrow \phi_{MAI} ,$$

$$B_{MXO} \rightarrow B_{MAI} .$$

APPENDIX C

INPUT LIST

Integer Variables

- `iquit` Can equal zero or one. If `iquit` equals one, the dropfile and output file are closed and control is returned to the terminal.
- `lsb0` Can equal any integer. It is the free parameter (ℓ) in the Boghosian sloshing-ion model distribution function. It is generally set equal to one.
- `lprnt0` Can equal any positive integer. It is the number of iterations the code makes in C05NAF between diagnostic steps.
- `mxcl0` Can equal any integer $\geq$ `lprnt0`. It is the maximum number of iterations the code will perform in C05NAF before halting and returning a failure message.
- `ifail` Can equal any integer from zero to six. See the description of C05NAF in the NAGFLIB manual.⁵
- `irenew` Can equal zero or one. If it equals one, the code will look for a solution starting at the point where it left off on the previous run.
- `maxwell` Can equal zero or one. If it equals zero, the transiting electrons are assumed to be rf dominated; for `maxwell` equal to one, they are treated as collision dominated.
- `axicell` Can equal zero or one. If it equals zero, the code will solve for the throttle-coil case; if it equals one, it will solve for the axicell case.

Real Variables

- `enc0` Can be any positive real number. It is the central-cell warm-ion density (cm^{-3}).
- `ench0` Can be any positive real number. It is the central-cell hot-ion density (cm^{-3}).
- `ensb0` Can be any positive real number. It is the sloshing-ion density at point b (cm^{-3}).
- `enitb0` Can be any positive real number. It is the trapped-ion density at point b (cm^{-3}).

enihx Can be any positive real number. It is the hot-ion density at point X (c_{th}^{-3}).
 tic0 Can be any positive real number. It is the central-cell warm-ion temperature (keV).
 tec0 Can be any positive real number. It is the central-cell electron temperature (keV).
 tsb0 Can be any positive real number. It is the temperature of the sloshing-ion beam (keV).
 eco0 Can be any real number greater than 170. It is the maximum energy of the hot electrons (keV).¹⁴
 bvac0c0 Can be any positive real number. It is the vacuum B-field in the central cell (T).
 bvacmai0 Can be any real number greater than bvac0c0. It is the vacuum B-field at MAI (T).
 bvacmao0 Can be any positive real number less than bvacmai0. It is the vacuum B-field at MAO (T).
 bvac0a0 Can be any positive real number less than bvacmao0. It is the vacuum B-field at A (T).
 bvac0b0 Can be any positive real number less than bvac0a0. It is the vacuum B-field at b (T).
 bvac0x0 Can be any real number greater than bvac0c0. It is the vacuum B-field at X (T).
 bvacmx0 Can be any real number greater than bvac0x0. It is the vacuum B-field at MXO (T).
 asb0 Must be equal to bvacmao0. It is the slope of the upper boundary of the sloshing ion integral in ($\epsilon - \mu$) space (T).
 bxb0 Can be any real number between zero and asb0. It is the slope of the lower boundary of the sloshing ion integral in ($\epsilon - \mu$) space (T).
 beta0 Can be equal to one or B_b/B_A . It is the parameter in the Cohen hot-electron distribution function. If it equals one, ECRH is dominating at A; if it equals B_b/B_A , ECRH is dominating at b (dimensionless).
 gamma0 Must be equal to one. It is the parameter for warm electrons that corresponds to beta0 for hot electrons (dimensionless).
 phie Can be any positive real number. It is the potential of the central cell (kV).

phic Can be any positive real number. It is the first-iteration estimate for ϕ_A minus ϕ_e (kV).
 phimai Can be any positive real number less than ϕ_e . It is the first-iteration estimate for ϕ_{MAI} (kV).
 phib Can be any real number less than phimai. It is the first-iteration guess at ϕ_b (kV).
 ftol0 Can be any real number, generally on the order of 10^{-10} . It is the maximum allowed value for the sum of the squares of the residuals if the code is to signal a solution (dimensionless).
 dlta0 Can be any real number, generally on the order of 10^{-6} . It specifies a suitable step length for the root finder (dimensionless).
 stpmx0 Can be any real number, generally set much greater than one. It is an estimate of the Euclidean distance between the expected solution and the starting point in the root-finding routine (dimensionless).
 phimxo Can be any real number between phie and phimai. It is the potential at MX0 (first-iteration guess in the axicell mode) (kV).
 phix Can be any real number between phie and phie + phic. It is the first-iteration guess at ϕ_x in the axicell mode (kV).
 alpha Can be any real number not in the closed interval $[0,1]$. It is the first-iteration guess at the Cohen filling parameter for trapped ions (dimensionless).

APPENDIX D
SAMPLE OUTPUT

The following are two sets of output. The input list is given before the results for each run. The input variable names are described in Appendix C.

```

alpha=      4.026692e+00
iquit      = 0
enc0       = 1.93e+13
ench0      = 960000000000.
ensb0      = 0.5e+13
tic0       = 15.
tsb0       = 80.
asb0       = 3.3
bsb0       = 1.45
lsb0       = 1
tec0       = 9.
beta0      = 0.71
gamma0     = 1.
eco0       = 400.
bvac0c0    = 1.
bvacmai0   = 3.3
bvac0b0    = 1.
bvacmao0   = 3.
phis       = 31.3
ftol0      = 1.e-10
dlta0      = 1.e-06
stpx0      = 100.
iprnt0     = 20
mxcl0      = 100
ifail      = 1
irenew     = 1
enitb0     = 0.2e+13
bvac0a0    = 1.4
phimci     = 16.6
phib       = 0.85
phic       = 30.
bvacmxo0   = 12.
phimxo     = 27.
alpha      = 2.
phix       = 35.
bvac0x0    = 4.
maxwell    = 1
axicell    = 1
enihx      = 5.e+13
s

```

upon 1 iterations in c05naf:

the independent variables are:

2.554692e+00	2.112851e-01	3.723300e+00
3.323262e+00	4.493932e+00	4.026692e+00

the residuals are:

-1.756684e-06	-1.999994e-01	-1.235681e-01
1.776691e-07	1.082525e-07	1.110083e-07

the sum of squares of the residuals is: 5.526884e-02

time spent in c05naf = 3.848150e-03 (min.)

total time spent in c05naf at this point = 3.876216e-01 (min.)

total time used at this point = 6.935571e-01 (min.)

upon return from c05naf:

the independent variables are:

2.547332e+00	1.200486e-01	3.440192e+00
3.297553e+00	4.484347e+00	3.856175e+00

the residuals are:

3.890486e-06	1.655481e-06	1.131289e-06
-1.567840e-07	-1.372142e-07	-6.371314e-09

the left half of the jacobian is:

-2.173421e+00	5.384560e-02	-1.427296e-01
1.953360e-01	1.625357e-04	-2.399559e+00
-5.141393e-01	-4.352427e-01	-1.457115e-02
8.516567e-02	3.812708e-04	-1.784378e+00
-2.679395e-01	2.424880e+00	-6.115923e+00
-4.881532e-02	1.989531e-03	9.226975e+00

the right half of the jacobian is:

-2.731319e-02	2.301991e-01	-5.811365e-01
-6.683684e-01	7.768272e-05	6.309274e-01
-1.014245e-02	8.565230e-02	-2.161583e-01
-5.648286e-03	-3.063696e-01	2.381674e-01
-1.731115e-01	1.300267e+00	-3.366431e+00
-2.259806e+00	2.946981e-03	-4.735703e+01

the sum of the squares of the residuals is 4.065425e-12

ifail = 0

the ion densities at the cardinal points are:

enipmai= 1.481032e+12	enipb= 2.922969e+11	
enisb0= 5.000000e+12	enisa= 1.210423e+13	enitb= 2.000000e+12
enitmai= 5.256730e+12	enipaxo=	
1.393978e+13		
enipx= 4.742550e+12	enihx= 5.000000e+13	

the electron densities at the cardinal points are:

enepmai= 6.737762e+12	enepb= 2.155588e+11	enepa= 5.136497e+10
enema= 1.050339e+13	enehb= 7.076730e+12	eneha= 1.549456e+12
enehmai= 0.	enepaxo=	
1.393978e+13		
enepx= 5.474255e+13		

the potentials are (kev):

```

phimai= 2.292595e+01  phib= 1.060437e+00  phic= 3.096173e+01
phimxo= 2.967797e+01  phix=
4.035512e+01
alpha= 3.856175e+00
iquit = 0
enc0 = 1.93e+13
ench0 = 960000000000.
ensb0 = 0.4e+13
tic0 = 15.
tsb0 = 80.
asb0 = 3.3
bsb0 = 1.45
lsb0 = 1
tec0 = 9.
beta0 = 0.71
gamma0 = 1.
eco0 = 400.
bvac0c0 = 1.
bvacmai0 = 3.3
bvac0b0 = 1.
bvacmac0 = 3.
phis = 31.3
fto10 = 1.e-10
dita0 = 1.e-06
stpmx0 = 100.
iprnt0 = 20
axc10 = 100
ifail = 1
irenew = 1
enitb0 = 0.2e+13
bvac0a0 = 1.4
phimai = 16.6
phib = 0.85
phic = 30.
bvacmx0 = 12.
phimxo = 27.
alpha = 2.
phix = 35.
bvac0x0 = 4.
maxwell = 1
axicell = 1
enihx = 5.e+13
$

```

upon 1 iterations in c05naf:

```

the independent variables are:
2.547332e+00  1.200486e-01  3.440192e+00
3.297553e+00  4.484347e+00  3.856175e+00
the residuals are:
3.890486e-06  -2.499979e-01  -1.254313e-01
-1.567840e-07  -1.372142e-07  -8.371314e-09
the sum of squares of the residuals is: 7.823199e-02

time spent in c05naf = 3.843567e-03 (min.)

```

total time spent in c05naf at this point = 3.914652e-01 (min.)
total time used at this point = 7.056148e-01 (min.)

upon return from c05naf:

the independent variables are:

2.536916e+00	1.409763e-02	3.060959e+00
3.259004e+00	4.470249e+00	3.625522e+00

the residuals are:

3.293937e-07	6.862722e-06	2.748621e-06
-1.173652e-06	-9.308290e-07	-1.960677e-07

the left half of the jacobian is:

-2.176637e+00	5.373968e-02	-1.828253e-01
1.984749e-01	7.491794e-05	-2.356558e+00
-5.180517e-01	-3.914997e-01	-4.168676e-02
7.797753e-02	1.620788e-04	-2.187696e+00
-2.919261e-01	2.258047e+00	-7.417589e+00
-1.435158e-01	1.495616e-04	9.097387e+00

the right half of the jacobian is:

-3.357027e-02	2.431482e-01	-7.995367e-01
-6.884542e-01	-1.310221e-04	6.754330e-01
-1.207881e-02	8.829286e-02	-2.901981e-01
-1.003799e-02	-3.098262e-01	2.600984e-01
-1.763606e-01	1.245323e+00	-4.144427e+00
-2.129580e+00	1.282252e-03	-4.146797e+01

the sum of the squares of the residuals is 5.704271e-11
ifail = 0

the ion densities at the cardinal points are:

enipmai= 1.403175e+12	enipb= 2.697094e+11	
enisb0= 4.000000e+12	enisc= 9.857187e+12	enitb= 2.000004e+12
enitmai= 5.251869e+12	enipmxc=	
1.331547e+13		
enipx= 3.949685e+12	enihx= 5.000000e+13	

the electron densities at the cardinal points are:

enepmai= 6.655037e+12	enepb= 1.825002e+11	enepa= 4.432281e+10
enewa= 8.498554e+12	enehb= 6.087186e+12	eneha= 1.314257e+12
enehmai= 0.	enepmxc=	
1.331550e+13		
enepx= 5.394870e+13		

the potentials are (kev):

phimai= 2.283225e+01	phib= 1.268787e-01	phic= 2.754863e+01
phimxc= 2.933104e+01	phix=	
4.023224e+01		
alpha= 3.625522e+00		

APPENDIX E
CODE LISTING

```

1  *PRECOMP %ME,SP%ME
2  *CFT I=SP%ME,L=SL%ME,B=SB%ME
3  *LDR I=SB%ME,K=%NERUN,NL=SN%ME,LIB=(NAG)
4  C
5  C
6  C
7  PROGRAM PSOLV
8  C
9  DIMENSION XH(6),RM(6),AJM(6,6),WM(102),LISTERR(25),LABERR(25)
10 C ,INSG(10),XOUT(6)
11 C
12 INTEGER AXICELL
13 C
14 EXTERNAL RESID1,MONIT1
15 C
16 COMMON/COMAI/ENCO,ENS80,ENIT80,TICO,TS80,AS80,BS80,LS80,
17 C ENECO,TECO,BETA0,GAMMA0,ECD0,BVACOC0,BVACHAIO,BVACOB0,BVACHA00
18 C BVACOA0,PHIE,ENIPHA1,ENIPB,ENISA,ENEPHA1,ENEPB,
19 C ENEPHA,ENEWA,ENEHB,ENEHA,BVACHX00,BVACOX0,ENITB,ENITHA1,
20 C ENEHHA1,ENIPX0,ENIPX,ENEPX0,ENEPX,MAXWELL,AXICELL,ENIHX
21 C
22 COMMON/COMAO/ NCTIME(4)
23 C
24 C
25 NANELIST/INP1/ IOUIT,ENCO,ENCHO,ENS80,TICO,TS80,AS80,
26 C BS80,LS80,TECO,BETA0,GAMMA0,ECD0,BVACOC0,BVACHAIO,BVACOB0,
27 C BVACHA00,PHIE,FTOLO,DLTA0,STPIX0,IPRNT0,KXCLO,IFAIL,IYRENEW,
28 C ENIT80,BVACOA0,PHIHA1,PHIB,PHIC,BVACHX00,PHIX0,ALPHA,
29 C PHIX,BVACOX0,MAXWELL,AXICELL,ENIHX
30 C
31 C
32 NE=25
33 C
34 NANELIST/INP2/ IDROP,IOUT,NINPUT
35 C
36 DATA LISTERR/501,502,503,504,505,506,507,508,509,510,511,
37 C 512,513,514,515,517,101,102,151,152,155,156,14,15,16/
38 C
39 CALL LINK("UNITS9=TTY//")
40 ASSIGN 998 TO IHELP
41 ASSIGN 100 TO LERRET
42 IZ=IZSETPI2(IHELP,1)
43 DO 1100 I=1,25
44 LABERR(I)=LERRET
45 1100 CONTINUE
46 CALL ERRCTL(LISTERR,LABERR,25)
47 CALL ERRMSG(0,1,1)
48 WRITE (59,50)
49 50 FORMAT(//,"INPUT DROFFILE, OUTPUT, AND INPUT FILE NAMES:")
50 READ(59,INP2)
51 CALL CREATE(3,IOUT,3,-1)
52 WRITE (3,40) IDROP,IOUT,NINPUT
53 40 FORMAT(/,"IDROP= ",A8,5X,"IOUT= ",A8,5X,"NINPUT= ",A8)
54 CALL DROFFILE(IDROP)
55 CALL OPEN(2,NINPUT,0,ILEN)
56 READ (2,INP1)
57 GO TO 100
58 998 IZ=IZGETHR(INSG,6,0,NRCV)
59 IZ=IZSETPI2(IHELP,1)

```

```

60      IZ=IZRELPI(1)
61      100 WRITE (S9,110)
62      110 FORMAT(//,"INPUT NEW NAMELIST INPUT:")
63      READ (S9,INP1)
64      WRITE (3,INP1)
65      IF (IQUIT .NE. 0) GO TO 999
66      IF (IRENEW .EQ. 1) THEN
67          IRENEW=0
68          GO TO 200
69      ENDIF
70      IF (AXICELL .NE. 0) THEN
71          NM=6
72          IWM=102
73      ELSE
74          NM=4
75          IWM=52
76      ENDIF
77      XM(1)=PHIMAI/TECO
78      XM(2)=PHIB/TECO
79      XM(3)=PHIC/TECO
80  C
81      IF (AXICELL .EQ. 0) THEN
82          XM(4)=ALPHA
83      ELSE
84          XM(4)=PHIMXO/TECO
85          XM(5)=PHIX/TECO
86          XM(6)=ALPHA
87      END IF
88      200 CONTINUE
89      ENECO=ENCO+ENCHO
90      CALL TIMGET(0)
91      CALL COSNAF( NM, XM, RM, FRM, AJM, NM, WM, IWM, FTOL0,
92      X      DLTA0, STPMX0, RESID1, MONIT1, IPANT0, MXCLO,
93      X      IFAIL)
94      CALL TIMGET(1)
95      FCTIME1=1.E-6*FLOAT(NCTIME(3))/60.
96      FCTIME2=1.E-6*FLOAT(NCTIME(1))/60.
97      FCTIME3=1.E-6*FLOAT(NCTIME(2))/60.
98      IF (AXICELL .EQ. 0) GO TO 595
99      GO TO 599
100     595 WRITE (S9,601) FCTIME1,FCTIME2,FCTIME3
101     WRITE (3,601) FCTIME1,FCTIME2,FCTIME3
102     601 FORMAT(//,"TIME SPENT IN COSNAF = ",E16.6,2X,"(MIN.)",/,
103     C "TOTAL TIME SPENT IN COSNAF AT THIS POINT = ",E16.6,2X,"(MIN.)"
104     C //,"TOTAL TIME USED AT THIS POINT = ",E16.6,2X,"(MIN.)")
105     WRITE (S9,301) (XM(J),J=1,NM),(RM(J),J=1,NM), ((AJM(I,J),J=1,NM)
106     C I=1,NM),FRM,IFAIL
107     WRITE (3,301) (XM(J),J=1,NM),(RM(J),J=1,NM), ((AJM(I,J),J=1,NM),
108     C I=1,NM),FRM,IFAIL
109     301 FORMAT(//,"UPON RETURN FROM COSNAF:",//,
110     C "THE INDEPENDENT VARIABLES ARE:",//,
111     C 4E16.6,/, "THE RESIDUALS ARE:",/,4E16.6,/,
112     C "THE JACOBIAN IS:",/,4E16.6,/,4E16.6,/,4E16.6,/,
113     C 4E16.6,/,
114     C "THE SUM OF THE SQUARES OF THE RESIDUALS IS ",E16.6,/,
115     C "IFAIL = ",I5)
116     WRITE (S9,351) ENIPHA1,ENIPB,ENSS0,ENISA,ENITB,ENITHAI

```

```

117      WRITE (3,351) ENIPMAI,ENIPB,ENSSO,ENISA,ENITB,ENITHAI
118 351 FORMAT(///,"THE ION DENSITIES AT THE CARDINAL POINTS ARE:")
119      C ///,"ENIPMAI= ",E16.6,4X,"ENIPB= ",E16.6
120      C ///,"ENISSO= ",E16.6,4X,"ENISA= ",E16.6,4X,"ENITB= ",E16.6,
121      C 4X,"ENITHAI= ",E16.6)
122      WRITE (59,361) ENEPMAI,ENEPB,ENEP A,ENEWA,ENEHB,ENEHA,ENEHMAI
123      WRITE (3,361) ENEPMAI,ENEPB,ENEP A,ENEWA,ENEHB,ENEHA,ENEHMAI
124 361 FORMAT(///,"THE ELECTRON DENSITIES AT THE CARDINAL POINTS ARE:")
125      C ///,"ENEPMAI= ",E16.6,4X,"ENEPB= ",E16.6,4X,"ENEP A= ",E16.6
126      C ///,"ENEWA= ",E16.6,4X,"ENEHB= ",E16.6,4X,"ENEHA= ",E16.6,
127      C 4X,"ENEHMAI= ",E16.6)
128      DO 371 I=1,NM-1
129      XOUT(I)=XM(I)*TECO
130 371 CONTINUE
131      XOUT(4)=XM(4)
132      WRITE (59,381) (XOUT(J),J=1,3)
133      WRITE (3,381) (XOUT(J),J=1,3)
134      WRITE (59,390) XOUT(4)
135      WRITE (3,390) XOUT(4)
136 381 FORMAT(///,"THE POTENTIALS ARE (KEV):",///,
137      C "PHIMAI= ",E16.6,5X,"PHIB= ",E16.6,5X,"PHIC= ",E16.6)
138      IFAIL=1
139      GO TO 389
140 599 WRITE (59,600) FCTIME1,FCTIME2,FCTIME3
141      WRITE (3,600) FCTIME1,FCTIME2,FCTIME3
142 600 FORMAT(///,"TIME SPENT IN COSNAF = ",E16.6,2X,"(MIN.)",/,
143      C "TOTAL TIME SPENT IN COSNAF AT THIS POINT = ",E16.6,2X,"(MIN.)")
144      C ///,"TOTAL TIME USED AT THIS POINT = ",E16.6,2X,"(MIN.)")
145      WRITE (59,300) (XM(J),J=1,NM),(RM(J),J=1,NM), ((AJH(I,J),J=1,NM)
146      C I=1,NM),FRM,IFAIL
147      WRITE (3,300) (XM(J),J=1,NM),(RM(J),J=1,NM), ((AJH(I,J),J=1,NM),
148      C I=1,NM),FRM,IFAIL
149 300 FORMAT(///,"UPON RETURN FROM COSNAF:",///,
150      C "THE INDEPENDENT VARIABLES ARE:",//,3E16.6,/,
151      C 3E16.6,/,,"THE RESIDUALS ARE:",//,3E16.6,/,3E16.6,/,
152      C "THE LEFT HALF OF THE JACOBIAN IS:",//,3E16.6,/,3E16.6,/,3E16.6,
153      C 3E16.6,/,3E16.6,/,3E16.6,/,
154      C "THE RIGHT HALF OF THE JACOBIAN IS:",//,3E16.6,/,3E16.6,/,3E16.6
155      C 3E16.6,/,3E16.6,/,3E16.6,/,
156      C "THE SUM OF THE SQUARES OF THE RESIDUALS IS ",E16.6,/,
157      C "IFAIL = ",IS)
158      WRITE (59,350) ENIPMAI,ENIPB,ENSSO,ENISA,ENITB,ENITHAI,ENIPMXO,
159      C ENIPX,ENIHX
160      WRITE (3,350) ENIPMAI,ENIPB,ENSSO,ENISA,ENITB,ENITHAI,ENIPMXO,
161      C ENIPX,ENIHX
162 350 FORMAT(///,"THE ION DENSITIES AT THE CARDINAL POINTS ARE:")
163      C ///,"ENIPMAI= ",E16.6,4X,"ENIPB= ",E16.6
164      C ///,"ENISSO= ",E16.6,4X,"ENISA= ",E16.6,4X,"ENITB= ",E16.6,
165      C 4X,"ENITHAI= ",E16.6,4X,"ENIPMXO= ",E16.6,4X,/,,"ENIPX= ",E16.6,
166      C 4X,"ENIHX= ",E16.6)
167      WRITE (59,360) ENEPMAI,ENEPB,ENEP A,ENEWA,ENEHB,ENEHA,ENEHMAI
168      C ,ENEPMXO,ENEPX
169      WRITE (3,360) ENEPMAI,ENEPB,ENEP A,ENEWA,ENEHB,ENEHA,ENEHMAI
170      C ,ENEPMXO,ENEPX

```

```

171 360 FORMAT(//,"THE ELECTRON DENSITIES AT THE CARDINAL POINTS ARE,"
172 C ///,"ENEPAI= ",E16.6,4X,"ENEPB= ",E16.6,4X,"ENEPA= ",E16.6
173 C ///,"ENENA= ",E16.6,4X,"ENEHB= ",E16.6,4X,"ENENA= ",E16.6,
174 C 4X,"ENEMAI= ",E16.6,4X,"ENEMXD= ",E16.6,4X,/,,"ENEPX= ",E16.6)

175 DO 370 I=1,NM-1
176 XOUT(I)=XM(I)*TECO
177 370 CONTINUE
178 XOUT(6)=XM(6)
179 WRITE (59,380) (XOUT(J),J=1,5)
180 WRITE (3,380) (XOUT(J),J=1,5)
181 WRITE (59,390) XOUT(6)
182 WRITE (3,390) XOUT(6)
183 380 FORMAT(//,"THE POTENTIALS ARE (KEV):",//,
184 C "PHINAI= ",E16.6,5X,"PHIB= ",E16.6,5X,"PHIC= ",E16.6,
185 C 5X,"PHIMXD= ",E16.6,5X,"PHIX= ",E16.6)
186 IFAIL=1
187 389 CONTINUE
188 390 FORMAT("ALPHA= ",E16.6)
189 GO TO 170
190 999 CALL CLOSE(3)
191 CALL CLOSE(2)
192 CALL QUIT(1)
193 END
194 C
195 C
196 FUNCTION IZSETPI2(X,Y)
197 DIMENSION BETA(4)
198 BETA(1)=30400B
199 BETA(3)=Y
200 BETA(4)=X
201 CALL IZRAY(BETA,4)
202 RETURN
203 END
204 C
205 C
206 SUBROUTINE TIMGET(K)
207 COMMON/COMAD/ NCTIME(3)
208 KA1=0
209 KA2=0
210 KA3=0
211 CALL TIMEUSED(KA1,KA2,KA3)
212 KATOT=KA1+KA2+KA3
213 KADIF=KATOT-NCTIME(2)
214 IF (K.EQ. 0) GO TO 10
215 NCTIME(K)=NCTIME(K)+KADIF
216 10 NCTIME(2)=KATOT
217 NCTIME(3)=KADIF
218 RETURN
219 END
220 C
221 C
222 C
223 C
224 SUBROUTINE RESID1(N,XC,RC)
225 C
226 C
227 DIMENSION XC(N),RC(N)
228 C
229 COMMON/COMAI/ENCO,ENSB0,ENITB0,TICO,TSB0,ASB0,BSB0,LSB0,

```

```

230      $      ENECO,TECO,BETA0,GAMMA0,ECO0,BVACOCO,BVACHAIO,
231      $      BVACOB0,BVACHAO0,BVACOA0,PHIE,ENIPMAI,ENIPB,ENISA,
232      $      ENEPMAI,ENEPB,ENEPB,ENEPB,ENEPB,ENEPB,ENEPB,BVACHX00,
233      $      BVACOX0,
234      $      ENITB,ENITHAI,ENEHMAI,ENIPMX0,ENIPX,ENEPMX0,ENEPX
235      $      ,MAXWELL,AXICELL,ENIPX
236
237      C
238      C.....GET INDEPENDENT VARIABLES:
239      PHIMAI=XC(1)*TECO
240      PHIB=XC(2)*TECO
241      PHIC=XC(3)*TECO
242      IF (AXICELL .NE. 0) THEN
243      PHIMX0=XC(4)*TECO
244      PHIX=XC(5)*TECO
245      ALPHA=XC(6)
246      ELSE
247      ALPHA=XC(4)
248      END IF
249      PHIA=PHIE+PHIC
250
251      C
252      C.....GET VARIABLES FROM COMMON BLOCK COMAI:
253      ENC=ENCO
254      ENEC=ENECO
255      ENSB=ENSB0
256      ENISB=ENSB
257      TIC=TICO
258      TSB=TSB0
259      ASB=ASB0
260      BSB=BSB0
261      LSB=LSB0
262      TEC=TECO
263      BETA=BETA0
264      GAMMA=GAMMA0
265      ECO=ECO0
266      BVACOC=BVACOCO
267      BVACHAI=BVACHAIO
268      BVACOB=BVACOB0
269      BVACHAO=BVACHAO0
270      BVACHX0=BVACHX00
271      BPLAOX=BVACOX0
272      BVACOA=BVACOA0
273      BPLMAI=BVACHAI
274      BPLMAO=BVACHAO
275      BPLAOB=BVACOB
276      BPLAOC=BVACOC
277      BPLAOA=BVACOA
278      BPLMX0=BVACHX0
279
280      C
281      C.....CALL SUBROUTINES FOR DENSITIES AND PRESSURES:
282      C
283      IF (AXICELL .EQ. 0) GO TO 100
284
285      C
286      CALL PASSION(ENC,TIC,PHIE,PHIC,PHIMX0,PHIMX0,BPLAOC,BPLAOA,
287      $      BPLMX0,BPLMX0,ENIPMX0)
288      CALL PASSION(ENC,TIC,PHIE,PHIC,PHIX,PHIX,BPLAOC,BPLAOA,
289      $      BPLAOX,BPLAOX,ENIPX)
290      CALL PASSELE(ENEC,TEC,PHIE,PHIB,PHIMX0,PHIMX0,BPLAOC,BPLAOB,
291      $      BPLMAO,BPLMX0,BPLMX0,MAXWELL,1,ENEPMX0)
292      CALL PASSELE(ENEC,TEC,PHIE,PHIB,PHIX,PHIX,PHIMX0,BPLAOC,BPLAOB,
293      $      BPLMAO,BPLAOX,BPLMX0,MAXWELL,1,ENEPX)

```

```

290      100 CALL SLSHION(ENSS,TSB,ASB,BSB,LSB,PHIB,PHIA,BPLA0B,
291      $              BPLA0A,4,ENISA)
292      C
293      IF (MAXWELL.EQ. 1) THEN
294      C
295      CALL PASSELE(ENEC,TEC,PHIE,PHIB,PHIMAI,PHIMAI,
296      $              BPLA0C,BPLA0B,BPLMA0,BPLMAI,BPLMAI,
297      $              MAXWELL,2,ENEPMI)
298      CALL PASSELE(ENEC,TEC,PHIE,PHIB,PHIB,PHIMAI,
299      $              BPLA0C,BPLA0B,BPLMA0,BPLA0B,BPLMAI,
300      $              MAXWELL,3,ENEPMI)
301      CALL PASSELE(ENEC,TEC,PHIE,PHIB,PHIA,PHIMAI,
302      $              BPLA0C,BPLA0B,BPLMA0,BPLA0A,BPLMAI,
303      $              MAXWELL,4,ENEPA)
304      CALL WARNELE(ENEC,TEC,GAMMA,PHIE,PHIMAI,PHIB,PHIA,
305      $              BPLA0C,BPLMAI,BPLA0B,BPLA0A,BPLMA0,
306      $              BPLA0A,4,ENEWA)
307      CALL HOTELEC(ENEC,TEC,BETA,ECO,PHIE,PHIMAI,PHIB,PHIB,
308      $              BPLA0C,BPLMAI,BPLA0B,BPLA0A,BPLMA0,
309      $              BPLA0B,3,ENEHB)
310      CALL HOTELEC(ENEC,TEC,BETA,ECO,PHIE,PHIMAI,PHIB,PHIA,
311      $              BPLA0C,BPLMAI,BPLA0B,BPLA0A,BPLMA0,
312      $              BPLA0A,4,ENEHA)
313      ENEHMAI=0.
314      C
315      ELSE
316      C
317      CALL PASSELE(ENEC,TEC,PHIE,PHIB,PHIMAI,PHIMXO,
318      $              BPLA0C,BPLA0B,BPLMA0,BPLMAI,BPLMXO,
319      $              MAXWELL,2,ENEPMI)
320      CALL PASSELE(ENEC,TEC,PHIE,PHIB,PHIB,PHIMXO,
321      $              BPLA0C,BPLA0B,BPLMA0,BPLA0B,BPLMXO,
322      $              MAXWELL,3,ENEPMI)
323      CALL PASSELE(ENEC,TEC,PHIE,PHIB,PHIA,PHIMXO,
324      $              BPLA0C,BPLA0B,BPLMA0,BPLA0A,BPLMXO,
325      $              MAXWELL,4,ENEPA)
326      CALL WARNELE(ENEC,TEC,GAMMA,PHIE,PHIMXO,PHIB,PHIA,
327      $              BPLA0C,BPLMXO,BPLA0B,BPLA0A,BPLMA0,
328      $              BPLA0A,4,ENEWA)
329      CALL HOTELEC(ENEC,TEC,BETA,ECO,PHIE,PHIMXO,PHIB,PHIB,
330      $              BPLA0C,BPLMXO,BPLA0B,BPLA0A,BPLMA0,
331      $              BPLA0B,3,ENEHB)
332      CALL HOTELEC(ENEC,TEC,BETA,ECO,PHIE,PHIMXO,PHIB,PHIMAI,
333      $              BPLA0C,BPLMXO,BPLA0B,BPLA0A,BPLMA0,BPLMAI,3,ENEHMAI)
334      CALL HOTELEC(ENEC,TEC,BETA,ECO,PHIE,PHIMXO,PHIB,PHIA,
335      $              BPLA0C,BPLMXO,BPLA0B,BPLA0A,BPLMA0,
336      $              BPLA0A,MAXWELL,4,ENEHA)
337      C
338      END IF
339      C
340      CALL TRAPION(ENC,TIC,ALPHA,PHIE,PHIMXO,PHIC,PHIMAI,BPLA0C,
341      $              BPLMXO,BPLA0A,BPLMAI,3,ENITHAI)
342      CALL TRAPION(ENC,TIC,ALPHA,PHIE,PHIMXO,PHIC,PHIB,BPLA0C,
343      $              BPLMXO,BPLA0A,BPLA0B,4,ENITB)
344      CALL PASSION(ENC,TIC,PHIE,PHIC,PHIMAI,PHIMXO,
345      $              BPLA0C,BPLA0A,BPLMAI,BPLMXO,ENIPMAI)
346      CALL PASSION(ENC,TIC,PHIE,PHIC,PHIB,PHIMXO,
347      $              BPLA0C,BPLA0A,BPLA0B,BPLMXO,ENIPB)
348      C

```

```

349 C.....COMPUTE RESIDUALS:
350 IF (AXICELL .EQ. 0) THEN
351 C
352 C CONSTRAINT ON ALPHA AT POINT B:
353 RC(4)=(ENITB0-ENITB)/ENC
354 ELSE
355 C
356 C CONSTRAINT ON ALPHA AT POINT B:
357 RC(6)=(ENITB0-ENITB)/ENC
358 C
359 C QUASINEUTRALITY AT POINT X:
360 RC(5)=(ENIPX+ENIHX-ENEPX)/ENC
361 C
362 C QUASINEUTRALITY AT POINT MX0:
363 RC(4)=(ENIPMX0-ENEPMX0)/ENC
364 C
365 END IF
366 C
367 C QUASINEUTRALITY AT POINT A:
368 RC(3)=(ENISA-ENEPA-ENEWA-ENEHA)/ENC
369 C
370 C QUASINEUTRALITY AT POINT B:
371 RC(2)=(ENIPB+ENITB+ENISB-ENEPB-ENEHB)/ENSB
372 C
373 C QUASINEUTRALITY AT POINT MAI:
374 RC(1)=(ENIPMAI+ENITMAI-ENEHMAI-ENEPMAI)/ENC
375 C
376 C
377 999 RETURN
378 END
379 C
380 SUBROUTINE MONIT1(NM,XCM,RCH,FCM,NCLM)
381 C
382 C
383 COMMON/COMAI/ENCO,ENSB0,ENITB0,TICO,TSB0,ASB0,BSB0,LSB0,
384 C ENECO,TECO,BETA0,GAMMA0,ECD0,BVACOCO,BVACMAIO,BVACQBO,BVACMAO,
385 C BVACQAO,PHIE,ENIPMAI,ENIPB,ENISA,ENEPMAI,ENEPB,
386 C ENEPA,ENEWA,ENEHB,ENEHA,BVACMX0,BVACOX0,ENITB,ENITMAI,
387 C ENEHMAI,ENIPMX0,ENIPX,ENEPMX0,ENEPX,MAXWELL,AXICELL,ENIHX
388 C
389 DIMENSION XCM(NM),RCH(NM)
390 IF (AXICELL .EQ. 0) GO TO 200
391 WRITE (59,100) NCLM,(XCM(J),J=1,NM),(RCH(J),J=1,NM),FCM
392 WRITE (3,100) NCLM,(XCM(J),J=1,NM),(RCH(J),J=1,NM),FCM
393 100 FORMAT(//,"UPON",I10," ITERATIONS IN COSNAF:",//,
394 C"THE INDEPENDENT VARIABLES ARE:",//,3E16.6,/,3E16.6,/,
395 C"THE RESIDUALS ARE:",//,3E16.6,/,3E16.6,/,
396 C"THE SUM OF SQUARES OF THE RESIDUALS IS:",2X,E16.6)
397 GO TO 999
398 200 WRITE (59,300) NCLM,(XCM(J),J=1,NM),(RCH(J),J=1,NM),FCM
399 WRITE (3,300) NCLM,(XCM(J),J=1,NM),(RCH(J),J=1,NM),FCM
400 300 FORMAT(//,"UPON",I10," ITERATIONS IN COSNAF:",//,
401 C"THE INDEPENDENT VARIABLES ARE:",//,4E16.6,/,
402 C"THE RESIDUALS ARE:",//,4E16.6,/,
403 C"THE SUM OF SQUARES OF THE RESIDUALS IS:",2X,E16.6)
404 999 RETURN
405 END
406 C
407 C

```



```

408 C
409 SUBROUTINE PASSION(ENC,TIC,PHIE,PHIC,PHIO,PHIMXO,
410 S BPLAOC,BPLAOA,BPLAO,BPLMXO,DENS)
411 C
412 C
413 DATA ETAINF/9.9E99/
414 REAL MUSTAR
415 C
416 C.....COMPUTE NORMALIZATION COEFFICIENT:
417 ETAE=PHIE/TIC
418 ETAC=PHIC/TIC
419 ETAA=ETAE+ETAC
420 PHIA=ETAA*TIC
421 IF (ETAA .LE. ETAE) THEN
422 DENS=0.
423 RETURN
424 ENDIF
425 RCA=BPLAOC/BPLAOA
426 OMCA=1.-RCA
427 EB1=RCA*ETAA-ETAE
428 IF(OMCA.LT.0.) THEN
429 ETATILC=ETAE-RCA*ETAC/OMCA
430 ELSE
431 ETATILC=ETAINF
432 ENDIF
433 DENOM=SFUN(0.,1.,1.,-ETAE,0.,ETAE,ETAA)+
434 S SFUN(0.,OMCA,OMCA,EB1,0.,ETAA,ETATILC)
435 DENNORM=ENC/DENOM
436 C
437 ETAO=PHIO/TIC
438 ROX=BPLAO/BPLMXO
439 ETAMXO=PHIMXO/TIC
440 OMROX=1.-ROX
441 EB3=ROX*ETAMXO-ETAO
442 EB4=(BPLMXO-BPLAO)/BPLAOA
443 EBS=-EB4*ETAA+ETAMXO-ETAO
444 ROA=BPLAO/BPLAOA
445 OMROA=1.-ROA
446 EB2=ROA*ETAA-ETAO
447 MUSTAR=(PHIA-PHIMXO)/(BPLMXO-BPLAOA)
448 ETAEPS=(BPLMXO*MUSTAR+PHIMXO)/TIC
449 DENS=DENNORM*(SFUN(0.,1.,1.,-ETAO,0.,ETAMXO,ETAA)
450 S -SFUN(0.,1.,OMROX,EB3,0.,ETAMXO,ETAEPS)
451 S +SFUN(0.,OMROA,OMROA,EB2,0.,ETAA,ETAEPS)
452 S +SFUN(0.,ROA,EB4,EBS,0.,ETAA,ETAEPS))
453 C
454 C
455 RETURN
456 END
457 C
458 C
459 C
460 C
461 SUBROUTINE TRAPION(ENC,TIC,ALPHA,PHIE,PHIMXO,PHIC,PHIO,BPLAOC,
462 S BPLMXO,BPLAOA,BPLAO,IREG,DENS)
463 C
464 C
465 DATA ETAINF/9.9E99/
466 C
467 C.....COMPUTE NORMALIZATION COEFFICIENT:

```

```

468      PHIA=PHIE+PHIC
469      ETAE=PHIE/TIC
470      ETAC=PHIC/TIC
471      ETAA=PHIA/TIC
472      ETAMXO=PHIMXO/TIC
473      ETAO=PHIO/TIC
474      RCA=BPLAOC/BPLAOA
475      OMRCA=1.-RCA
476      EB1=RCA*ETAA-ETAE
477      IF(OMRCA.LT.0.) THEN
478          ETATILC=ETAE-RCA*ETAC/OMRCA
479      ELSE
480          ETATILC=ETAINF
481      ENDIF
482      DENOM=SFUN(0.,1.,1.,-ETAE,0.,ETAE,ETAA)+
483      $ SFUN(0.,OMRCA,OMRCA,EB1,0.,ETAA,ETATILC)
484      DENNORM=ENC*BPLAO/(4.*TIC**1.5*DENOM)
485  C
486  C.....MAKE SURE REGION NUMBER IS 3 OR 4, AND (ALPHA.GT.1.):
487      IF(((IREG.NE.3).AND.(IREG.NE.4)).OR.((ALPHA.LE.1.).AND.
488      $ (ALPHA.GE.0.))) THEN
489          DENS=0.
490          GO TO 999
491      ENDIF
492  C
493  C
494  C.....COMPUTE QUANTITIES USEFUL TO ALL THREE CASES:
495      EPSS=PHIMXO-PHIO
496      EPS=(PHIA-PHIO)+(BPLAOA-BPLAO)*(PHIA-PHIMXO)/
497      $ (BPLMXO-BPLAOA)
498      OMALPHA=1.-ALPHA
499      RMXO=BPLMXO/BPLAO
500      RAO=BPLAOA/BPLAO
501      OMRAO=1.-RAO
502      OMA2T=2.*OMALPHA*TIC
503      C1=(1.+ALPHA*RMXO)/OMA2T
504      C2=BPLAO*(1.-ALPHA*RMXO)/OMA2T
505      C3=(ALPHA-OMALPHA*RAO+ALPHA*RMXO)/OMA2T
506      C4=BPLAO*(ALPHA+OMALPHA*RAO-ALPHA*RMXO)/OMA2T
507      RRMXO=(RMXO+1.)/(RMXO-1.)
508      D1=C1+C2/BPLAO
509      D2=C1+C2*RRMXO/BPLAO
510      D4=C3+C4/BPLAO
511      D5=C3+C4*RRMXO/BPLAO
512      D7=(ETAO-ALPHA*ETAMXO)/OMALPHA
513      D8=D7+2.*C2*(PHIO-PHIMXO)/(BPLMXO-BPLAO)
514      D10=(OMALPHA*ETAA-ALPHA*(ETAMXO-ETAO))/OMALPHA
515      D11=D10+2.*C4*(PHIO-PHIMXO)/(BPLMXO-BPLAO)
516      OOC2=1./C2
517      OOC4=1./C4
518      OOD1=1./D1
519      OOD2=1./D2
520      OOD4=1./D4
521      OOD5=1./D5
522      D1EPSS=D1*EPSS
523      D2EPSS=D2*EPSS
524      D2EPS=D2*EPS
525      D4EPSS=D4*EPSS
526      D5EPSS=D5*EPSS
527      D5EPS=D5*EPS

```

```

528      D112=(D1-C1-C2/BPLA0)/D1**2
529      D212=(D2-C1-C2/BPLA0)/D2**2
530      D434=(D4-C3-C4/BPLA0)/D4**2
531      D534=(D5-C3-C4/BPLA0)/D5**2
532      O87=(1.+D8-D7)/D2
533      O1110=(1.+D11-D10)/D5
534  C
535  C.....SELECT APPROPRIATE CASE:
536      IF(BPLA0.LT.BPLA0A) GO TO 100
537      IF(BPLA0.EQ.BPLA0A) GO TO 200
538      IF(BPLA0.GT.BPLA0A) GO TO 300
539  C
540  C.....CASE 1: (BPLA0.LT.BPLA0A)
541      100 RRA=(BPLA0A+BPLA0)/(BPLA0A-BPLA0)
542          D3=C1+C2*RRR/BPLA0
543          D6=C3+C4*RRR/BPLA0
544          D9=D7+2.*C2*(PHI0-ETA**TIC)/(BPLA0A-BPLA0)
545          D12=D10+2.*C4*(PHI0-ETA**TIC)/(BPLA0A-BPLA0)
546          O003=1./D3
547          O006=1./D6
548          D3EPS=D3*EPS
549          D6EPS=D6*EPS
550          D312=(D3-C1-C2/BPLA0)/D3**2
551          D634=(D6-C3-C4/BPLA0)/D6**2
552          O97=(1.+D9-D7)/D3
553          O1210=(1.+D12-D10)/D6
554          DENS=DENORM*(OOC2*
555      $      (TFUN(0.,O001,O001,0.,D7,0.,D1EPSS)+
556      $      TFUN(0.,O002,O002,0.,D8,D2EPSS,D2EPS))+
557      $      TFUN(0.,O003,O003,0.,D9,D3EPS,ETA**INF))-
558      $      OOC4*
559      $      (TFUN(0.,O004,O004,0.,D10,0.,D4EPSS)+
560      $      TFUN(0.,O005,O005,0.,D11,D5EPSS,D5EPS))+
561      $      TFUN(0.,O006,O006,0.,D12,D6EPS,ETA**INF)))
562      GO TO 999
563  C
564  C.....CASE 2: (BPLA0.EQ.BPLA0A)
565      200 DENS=DENORM*(OOC2*
566      $      (TFUN(0.,O001,O001,0.,D7,0.,D1EPSS)+
567      $      TFUN(0.,O002,O002,0.,D8,D2EPSS,D2EPS))-
568      $      OOC4*
569      $      (TFUN(0.,O004,O004,0.,D10,0.,D4EPSS)+
570      $      TFUN(0.,O005,O005,0.,D11,D5EPSS,D5EPS)))
571      GO TO 999
572  C
573  C.....CASE 3: (BPLA0.GT.BPLA0A)
574      300 RRA=(BPLA0A+BPLA0)/(BPLA0A-BPLA0)
575          D3=C1+C2*RRR/BPLA0
576          D6=C3+C4*RRR/BPLA0
577          D9=D7+2.*C2*(PHI0-ETA**TIC)/(BPLA0A-BPLA0)
578          D12=D10+2.*C4*(PHI0-ETA**TIC)/(BPLA0A-BPLA0)
579          O003=1./D3
580          O006=1./D6
581          D3EPS=D3*EPS
582          D6EPS=D6*EPS
583          D312=(D3-C1-C2)/D3**2
584          D634=(D6-C3-C4)/D6**2
585          O97=(1.+D9-D7)/D3
586          O1210=(1.+D12-D10)/D6
587          DENS=DENORM*(OOC2*

```

```

588 $ (TFUN(0.,0001,0001,0.,D7,0.,D1EPS))+
589 $ TFUN(0.,0002,0002,0.,D8,D2EPSS,D2EPS)-
590 $ TFUN(0.,0003,0003,0.,D9,0.,D3EPS))-
591 $ OOC4*
592 $ (TFUN(0.,0004,0004,0.,D10,0.,D4EPSS)+
593 $ TFUN(0.,0005,0005,0.,D11,D5EPSS,D5EPS)-
594 $ TFUN(0.,0006,0006,0.,D12,0.,D6EPS)))
595 C
596 999 RETURN
597 END
598 C
599 C
600 C
601 C
602 SUBROUTINE SLSHION(ENO,T0,ALPHA,BETA,L,PHIB,PHIO,BPLA0B,
603 $ BPLA0,IREG,DENS)
604 C
605 C
606 C....COMPUTE NORMALIZATION COEFFICIENT:
607 XI10=1.-BPLA0B/ALPHA
608 XI20=1.-BPLA0B/BETA
609 ETA0=PHIB/T0
610 XNORM=XI10*(XI10-5.*XI20)*UFUN(L+2,XI10,ETA0)+
611 $ (3.*XI10+5.*XI20)*ETA0*UFUN(L+1,XI10,ETA0)-
612 $ 4.*ETA0**2*UFUN(L,XI10,ETA0)-
613 $ XI20*(XI20-5.*XI10)*UFUN(L+2,XI20,ETA0)-
614 $ (3.*XI20+5.*XI10)*ETA0*UFUN(L+1,XI20,ETA0)+
615 $ 4.*ETA0**2*UFUN(L,XI20,ETA0)
616 XNORM=1./XNORM
617 C
618 C....MAKE SURE REGION NUMBER IS 3, 4 OR 5, AND (ALPHA.NE.BETA):
619 IF(((IREG.NE.3).AND.(IREG.NE.4)).AND.(IREG.NE.5)).OR.
620 $ (ALPHA.EQ.BETA)) THEN
621 DENS=0.
622 GO TO 999
623 ENDIF
624 C
625 C....COMPUTE DENSITY AND PRESSURES:
626 XI1=1.-BPLA0/ALPHA
627 XI2=1.-BPLA0/BETA
628 ETA=PHIO/T0
629 DENS=(ENO*XNORM)*(BPLA0B/BPLA0)**2*
630 $ (XI1*(XI1-5.*XI2)*UFUN(L+2,XI1,ETA)+
631 $ (3.*XI1+5.*XI2)*ETA*UFUN(L+1,XI1,ETA)-
632 $ 4.*ETA**2*UFUN(L,XI1,ETA)-
633 $ XI2*(XI2-5.*XI1)*UFUN(L+2,XI2,ETA)-
634 $ (3.*XI2+5.*XI1)*ETA*UFUN(L+1,XI2,ETA)+
635 $ 4.*ETA**2*UFUN(L,XI2,ETA))
636 C
637 999 RETURN
638 END
639 C
640 C
641 C
642 C
643 SUBROUTINE PASSELE(ENC,TEC,PHIE,PHIB,PHIO,PHIX0,BPLA0C,
644 $ BPLA0B,BPLA0D,BPLA0E,BPLMX0,NAKWEIL,IREG,DENS)
645 C
646 C

```

```

647     REAL MUSTAR,MUTWDL
648     DATA ETAINF/9.9E99/
649   C
650   C.....COMPUTE NORMALIZATION CONSTANTS:
651     ETAE=PHIE/TEC
652     RCMAO=BPLAO/BPLMAO
653     OMRCMAO=1.-RCMAO
654     IF (OMRCMAO.LT.0.) THEN
655       ETATILC=-ETAE/OMRCMAO
656     ELSE
657       ETATILC=ETAINF
658     ENDIF
659     DENOM=SFUN(0.,1.,1.,ETAE,0.,-ETAE,0.)+
660     $ OMRCMAO*SFUN(0.,1.,OMRCMAO,ETAE,0.,0.,ETATILC)
661     DENNORM=ENC/DENOM
662   C
663   C.....SELECT EXPRESSIONS APPROPRIATE TO REGION NUMBER:
664     IF (IREG.EQ. 2 ) GO TO 200
665     IF (IREG.EQ. 1 ) GO TO 100
666     GO TO 300
667   C
668   C.....REGION 1:
669     100 ETAO=PHIO/TEC
670     ROMAO=BPLAO/BPLMAO
671     OMROMAO=1.-ROMAO
672     ETAEPS=BPLMAO*PHIO/((BPLAO-BPLMAO)*TEC)
673     IF (BPLAO.LE. BPLMAO) THEN
674       ETAEPT=ETAINF
675     ELSE
676       ETAEPT=ETAEPS
677     END IF
678     DENS=DENNORM*(SFUN(0.,1.,1.,ETAO,0.,-ETAO,0.)
679     $ +SFUN(0.,OMROMAO,OMROMAO,ETAO,0.,0.,ETAEPT))
680     GO TO 999
681   C
682   C.....REGION 2:
683     200 IF (MAXWELL.NE. 0) GO TO 100
684     ETAO=PHIO/TEC
685     ROMAO=BPLAO/BPLMAO
686     OMROMAO=1.-ROMAO
687     ROMXO=BPLAO/BPLMXO
688     OMROMXO=1.-ROMXO
689     EPSTAR=BPLMAO*PHIMXO/(BPLMXO-BPLMAO)
690     MUSTAR=EPSTAR/BPLMAO
691     MUTWDL=(PHIMXO-PHIO)/(BPLMXO-BPLAO)
692     IF (MUTWDL.LE. 0) THEN
693       MUTWDL=0.
694     EPTWDL=-PHIMXO
695     ELSE
696       EPTWDL=BPLAO*MUTWDL-PHIO
697     END IF
698     ETAEPT=EPTWDL/TEC
699     ETAMUT=MUTWDL/TEC
700     ETAMXO=PHIMXO/TEC
701     ETAEPS=EPSTAR/TEC
702     EB2=ETAO-ROMXO*ETAMXO
703     EB5=BPLMAO*ETAMUT
704     EB6=(BPLMXO-BPLAO)/BPLMAO
705     EB7=ETAO-ETAMXO
706     DENS=DENNORM*(SFUN(0.,1.,1.,ETAO,0.,-ETAO,0.)

```

```

707 $ -SFUN(0.,1.,OMROMXO,EB2,0.,ETAEP,ETAEP)
708 $ +SFUN(0.,OMROMAO,OMROMAO,ETA0,0.,0.,ETAEP)
709 $ +SFUN(0.,ROMAO,EB6,EB7,0.,EB5,ETAEP)
710 GO TO 999
711 C
712 C.....REGION 3:
713 300 ETA0=PHIO/TEC
714 ROMAO=BPLAO/BPLMAO
715 OMROMAO=1.-ROMAO
716 ROMXO=BPLAO/BPLMXO
717 OMROMXO=1.-ROMXO
718 ROB=BPLAO/BPLAOB
719 OMROB=1.-ROB
720 ETAB=PHIB/TEC
721 EB1=ETA0-ROB*ETAB
722 EPSTAR=BPLMAO*PHIMXO/(BPLMXO-BPLMAO)
723 MUSTAR=EPSTAR/BPLMAO
724 MUTIDL=(PHIMXO-PHIB)/(BPLMXO-BPLAOB)
725 EPTIDL=BPLAOB*MUTIDL-PHIB
726 ETAEP=EPTIDL/TEC
727 ETAMUT=MUTIDL/TEC
728 ETAMXO=PHIMXO/TEC
729 ETAEP=EPSTAR/TEC
730 EB2=ETA0-ROMXO*ETAMXO
731 EB3=BPLAOB/BPLMAO-ROMAO
732 EB4=ETA0-ETAB
733 EB5=BPLMAO*ETAMUT
734 EB6=(BPLMXO-BPLAO)/BPLMAO
735 EB7=ETA0-ETAMXO
736 DENS=DENNORM*(SFUN(0.,1.,1.,ETA0,0.,-ETAB,0.)
737 $ -SFUN(0.,1.,OMROB,EB1,C.,-ETAB,ETAEP)
738 $ -SFUN(0.,1.,OMROMXO,EB2,0.,ETAEP,ETAEP)
739 $ +SFUN(0.,OMROMAO,OMROMAO,ETA0,0.,0.,ETAEP)
740 $ +SFUN(0.,ROMAO,EB3,EB4,0.,0.,EB5)
741 $ +SFUN(0.,ROMAO,EB6,EB7,0.,EB5,ETAEP))
742 GO TO 999
743 C
744 C
745 C.....REGION 6:
746 500 DENS=0.
747 C
748 999 RETURN
749 END
750 C
751 C
752 C
753 C
754 SUBROUTINE WARNELE(ENEC,TEC,GAMFAC,PHIE,PHIMXO,PHIB,PHIO,BPLAOC,
755 $ BPLMXO,BPLAOB,BPLAOA,BPLMAO,BPLAO,IREG,DENS)
756 C
757 REAL MUSTR
758 DIMENSION A(2),R(2)
759 DATA ETAINF /9.9E99/
760 IF (IREG.NE. 4 .AND. IREG.NE. 5) THEN
761 DENS=0.
762 RETURN
763 ENDIF
764 IF (BPLAO .LE. BPLAOB) THEN
765 DENS=0.

```

```

766          GO TO 40
767      ENDIF
768      C
769      C      COMPUTE NORMALIZATION
770      C
771      ETAE=PHIE/TEC
772      ROCHAO=BPLAOC/BPLMAO
773      OMR=1.-ROCHAO
774      DENOM=SFUN(0.,1.,1.,ETAE,0.,-ETAE,0.)+OMR*SFUN(0.,1.,OMR,ETAE,
775      $ 0.,0.,ETAINF)
776      DENNORM=ENEC/DENOM
777      MUSTR=(PHIMXO-PHIB)/(BPLMXO-BPLAOB)
778      EPSSTR=MUSTR*BPLMXO-PHIMXO
779      EPSTLD=(GAMFAC*BPLAOA*MUSTR-EPSSSTR)/TEC
780      RATIOB1=BPLAO/(BPLAOB-BPLAO)
781      RATIOB2=(GAMFAC*BPLAOA-BPLAO)/(BPLMXO-BPLAOB)
782      RATIOB3=(BPLAOB-BPLAO)/(BPLMXO-BPLAOB)
783      ETA1=(PHIMXO-PHIB)/TEC
784      ETA2=PHIO/TEC-EPSTLD
785      ETA3=(PHIO-PHIB)/TEC
786      ETA4=EPSTLD-PHIB/TEC
787      C
788      C      COMPUTE REGION IV CONTRIBUTION
789      C
790      RDIS1=-BPLMXO/(BPLMXO-BPLAOB)
791      RDIS2=RDIS1*BPLMAO/BPLMXO
792      ETAS=PHIMXO/TEC
793      DISFUNC=EXP(ETAS+RDIS1*ETA1)-EXP(RDIS2*ETA1)
794      IFLAG=0
795      ATEST=ABS(1.-GAMFAC*BPLAOA/BPLAO)
796      IF (ATEST .LE. 1.E-14) IFLAG=1
797      IF (IFLAG .EQ. 1) GO TO 10
798      RATIOB4=BPLAO/(GAMFAC*BPLAOA-BPLAO)
799      ARG1=RATIOB2*ETA1+ETA2
800      ARG1=AMAX1(ARG1,1.E-16)
801      ARG2=AMAX1(ETA2,1.E-16)
802      TERMXO=RATIOB4*(ARG1**1.5-ARG2**1.5)
803      GO TO 15
804      10  RATIOB5=1.5*BPLAO/(BPLMXO-BPLAOB)*ETA1
805      ARG1=AMAX1(0.,ETA2)
806      TERMXO=RATIOB5*SQRT(ARG1)
807      15  ARG3=RATIOB3*ETA1+ETA3
808      ARG3=AMAX1(ARG3,1.E-16)
809      TERMAO=RATIOB1*ARG3**1.5
810      DENS=2.*DISFUNC*(TERMXO-TERMAO)*DENNORM/3.
811      DENS=AMAX1(DENS,0.)
812      C
813      C      COMPUTE REGION V CONTRIBUTION
814      C
815      40  R(1)=BPLAOB/(BPLAOB-GAMFAC*BPLAOA)
816      R(2)=R(1)*BPLMAO/BPLAOB
817      RATIOB6=GAMFAC*BPLAOA*R(1)/BPLAOB
818      RATIOB7=(BPLAO-BPLAOB)/(GAMFAC*BPLAOA-BPLAOB)
819      ETAS=-ETA4*RATIOB7
820      ETA7=ETAS+ETA3
821      ETAB=PHIB/TEC
822      A(1)=EXP(-RATIOB6*ETAB)
823      A(2)=-EXP(-R(2)*ETAB)
824      SUM=0.
825      IF (IFLAG .EQ. 1) GO TO 45

```

```

826      DO 42 J=1,2
827      FAC=R(J)*ETA4
828      FAC1=1./R(J)
829      FAC2=FAC1*RATIOB7
830      TERMXO=SFUN(0.,1.,FAC2,ETA7,0.,0.,FAC)
831      TERMAO=SFUN(0.,1.,FAC1,ETA2,0.,0.,FAC)
832      SUM=SUM+A(J)*FAC1*EXP(R(J)*EPSTLD)*((TERMXO-TERMAO)
833 42      CONTINUE
834      SUM=SUM*RATIOB4
835      GO TO 50
836 45      DO 47 J=1,2
837      FAC=R(J)*ETA4
838      FAC1=1./R(J)
839      TERMXO=TFUN(-1.,FAC,FAC1,ETA2,0.,0.,FAC)
840      SUM=SUM+A(J)*FAC1**2*TERMXO*EXP(R(J)*EPSTLD)
841 47      CONTINUE
842      SUM=-0.5*SUM*RATIOB6
843 50      DENS=DENS+SUM*DENNORM
844      RETURN
845      END
846      C
847      C
848      C
849      C
850      SUBROUTINE HOTELEC(ENEC,TEC,BETA,EPSC,PHIE,PHIXO,PHIB,
851      S      PHIO,BPLAOC,BPLMXO,BPLAOB,BPLAOA,
852      S      BPLMAO,BPLAO,IRES,DENS)
853      C
854      REAL MUONE,MU4
855      DIMENSION A(2),AA(2),R(2)
856      DATA ETAINF /9.E9/,ONETH /3.333333333333E-01/
857      IF (IRES.NE. 3 .AND. IRES.NE. 4 .AND. IRES.NE. 5) THEN
858          DENS=0.
859          RETURN
860      ENDIF
861      C
862      C      COMPUTE NORMALIZATION
863      C
864      DENS=0.
865      ETAE=PHIE/TEC
866      ROCHAO=BPLAOC/BPLMAO
867      OMR=1-ROCHAO
868      DENOM=SFUN(0.,1.,1.,ETAE,0.,-ETAE,0.)+
869 1 OMR*SFUN(0.,1.,OMR,ETAE,0.,0.,ETAINF)
870      DENNORM=ENEC/DENOM
871      MUONE=(PHIXO-PHIB)/(BPLMXO-BPLAOB)
872      EPSONE=MUONE*BPLMXO-PHIXO
873      MU4=PHIXO/(BPLMXO-BPLMAO+1.E-15)
874      FAC1=(BPLMXO-BETA*BPLAOA)*(MU4-MUONE)/TEC
875      FAC1=AA(1)(FAC1,0.)
876      RATIOB2=(BPLMXO-BPLAO)/(BPLMXO-BETA*BPLAOA)
877      RATIOB3=BPLAO/(BETA*BPLAOA)
878      ATEST=ABS(1.-RATIOB3)
879      IFLAGBA=0
880      IF (ATEST.LE. 1.E-14) IFLAGBA=1
881      ETA1=EPSONE/TEC
882      ETA2=EPSC/TEC
883      ETA3=MUONE*BPLAO/TEC
884      ETA6=(EPSONE-BETA*BPLAOA*MUONE)/TEC
885      ETAS=PHIO/TEC

```



```

886      ETA4=PHIMX0/TEC
887      ETA7=ETA2-ETA1
888      ETAS=ETA6+ETA5
889  C
890  C      COMPUTE REGION II CONTRIBUTION
891  C
892      R(1)=BPLMX0/(BPLMX0-BETABPLA0A)
893      R(2)=BPLMA0/(BPLMA0-BETABPLA0A)
894      RATIOB4=R(1)*BETABPLA0A/BPLMX0
895      A(1)=RATIOB4*ETA4+R(1)*ETA6
896      A(2)=R(2)*(ETA4+ETA6)
897      AA(1)=1.
898      AA(2)=-1.
899      FAC3=ETA1-ETA3+ETA5
900      SUM=0.
901      IF (IFLAGBA .EQ. 1) GO TO 10
902      RATIOB5=1.-RATIOB3
903      RATIOB6=BPLA0/(BPLA0-BETABPLA0A)
904      FAC2=RATIOB3*ETA1+RATIOB5*ETA2-ETA3+ETA5
905      DO 5 J=1,2
906      FAC1L=1./R(J)
907      FAC2L=RATIOB3*FAC1L
908      FAC3L=RATIOB2*FAC1L
909      FAC4L=FAC1*R(J)
910      TERMX0=SFUN(0.,1.,FAC2L,FAC2,A(J),0.,FAC4L)
911      TERMA0=SFUN(0.,1.,FAC3L,FAC3,A(J),0.,FAC4L)
912      SUM=SUM+AA(J)*(TERMX0-TERMA0)*FAC1L
913  5      CONTINUE
914      SUM=-SUM*RATIOB6
915      GO TO 15
916  10      DO 12 J=1,2
917      FAC1L=1./R(J)
918      FAC3L=R(J)*FAC1
919      FAC4L=-R(1)*FAC1L
920      TERMX0=TFUN(FAC4L,ETA7,FAC1L,FAC3,A(J),0.,FAC3L)
921      SUM=SUM+AA(J)*FAC1L*TERMX0
922  12      CONTINUE
923      SUM=0.5*SUM
924  15      DENS=SUM/DENORM
925  C      COMPUTE REGION III CONTRIBUTION
926  C
927  999      CONTINUE
928      ETAS=(PHIMX0-PHIB)/TEC
929      ARG1=ETAS*BPLMX0/(BPLMX0-BPLA0B)
930      ARG2=-ARG1*BPLMA0/BPLMX0
931      DISFUNC=(EXP(ETA4-ARG1)-EXP(ARG2))
932      IFLAGBB=0
933      RATIOB8=BPLA0/BPLA0B
934      ATEST3=ABS(1.-RATIOB8)
935      IF (ATEST3 .LE. 1.E-14) IFLAGBB=1
936      IF (IFLAGBA .EQ. 1) GO TO 40
937      ARG01=AMAX1(0.,FAC2)
938      ARG02=AMAX1(0.,FAC3)
939      RATIOB7=BETABPLA0A/(BETABPLA0A-BPLA0)
940      DENSB= RATIOB7*(ARG01**1.5-ARG02**1.5)
941      GO TO 45
942  40      ARG01=AMAX1(0.,ETA6)
943      DENSB=1.5*SQRT(ARG01)*ETA7
944  45      IF (IFLAGBB .EQ. 1) GO TO 50
945      ETA10=(1.-RATIOB8)*ETA2+RATIOB8*ETA1-ETA3+ETA5

```

```

946      ARG03=AMAX1(0.,ETA10)
947      ARG04=AMAX1(0.,FAC3)
948      RATIOB8=BPLA08/(BPLA08-BPLA0)
949      DENS88=RATIOB8*(ARG04**1.5-ARG03**1.5)
950      GO TO 55
951 50  ETA11=(PHI0-PHIB)/TEC
952      ARG03=AMAX1(0.,ETA11)
953      DENS88=-1.5*SQRT(ARG03)*ETA7
954 55  DENS=DENS+(DENS8A+DENS88)*2.*ONETH=DISFUNC=DENNORM
955      RETURN
956      END
957
958
959
960 C
961 C
962 C
963 C
964      FUNCTION SFUN(A,B,C,D,X0,X1,X2)
965      C
966      C
967 C.....THIS FUNCTION COMPUTES THE REAL PART OF THE INTEGRAL FROM
968      C X1 TO X2 OF (A*T+B)*EXP(-X0-T)*SQRT(C*T+D) WITH RESPECT TO T.
969      C ALL ARGUMENTS MUST BE REAL. THIS ROUTINE CALLS THE COM-
970      C PLEX INCOMPLETE GAMMA FUNCTION ROUTINE, GAMMAINC.
971      C
972      COMPLEX GAMMAINC,CANS
973      C
974      IF(C.EQ.0.0) THEN
975          IF(D.GT.0.0) THEN
976              CANS=(A*(GAMMAINC(2.,X0,X1)-GAMMAINC(2.,X0,X2))+
977              $          B*(GAMMAINC(1.,X0,X1)-GAMMAINC(1.,X0,X2)))*SQRT(D)
978              SFUN=REAL(CANS)
979              GO TO 999
980          ELSE
981              SFUN=0.
982              GO TO 999
983          ENDIF
984      ELSE
985          Y0=-D/C
986          IF(C.GT.0.0) THEN
987              SIG1=AMAX1(Y0,X1)-Y0
988              SIG2=AMAX1(Y0,X2)-Y0
989              IF(SIG1.EQ.SIG2) THEN
990                  SFUN=0.
991                  GO TO 999
992              ELSE
993                  X3=Y0+X0
994                  CANS=SQRT(C)*
995                  $          (A*(GAMMAINC(2.5,X3,SIG1)-GAMMAINC(2.5,X3,SIG2))+
996                  $          (A*Y0+B)*(GAMMAINC(1.5,X3,SIG1)-
997                  $          GAMMAINC(1.5,X3,SIG2)))
998                  SFUN=REAL(CANS)
999                  GO TO 999
1000             ENDIF
1001         ELSE
1002             SIG1=AMIN1(Y0,X1)-Y0
1003             SIG2=AMIN1(Y0,X2)-Y0
1004             IF(SIG1.EQ.SIG2) THEN
1005                 SFUN=0.0

```

```

1006             GO TO 999
1007         ELSE
1008             X3=Y0+X0
1009             CANS=(0.,-1.)*SQRT(-C)*
1010             $      (A*(GAMMAINC(2.5,X3,SIG1)-GAMMAINC(2.5,X3,SIG2))+
1011             $      (A*Y0+B)*(GAMMAINC(1.5,X3,SIG1)-
1012             $      GAMMAINC(1.5,X3,SIG2)))
1013             SFUN=REAL(CANS)
1014             GO TO 999
1015         ENDIF
1016     ENDIF
1017 ENDIF
1018 C
1019 999 RETURN
1020 END
1021 C
1022 C
1023 C
1024 C
1025 FUNCTION TFUN(A,B,C,D,X0,X1,X2)
1026 C
1027 C
1028 C.....THIS FUNCTION COMPUTES THE REAL PART OF THE INTEGRAL FROM
1029 C      X1 TO X2 OF (A*(T+B)*EXP(-X0-T))/SQRT(C*T+D) WITH RESPECT TO T.
1030 C      ALL ARGUMENTS MUST BE REAL. THIS ROUTINE CALLS THE COM-
1031 C      PLEX INCOMPLETE GAMMA FUNCTION ROUTINE, GAMMAINC.
1032 C
1033     COMPLEX GAMMAINC,CANS
1034 C
1035     IF(C.EQ.0.0) THEN
1036         IF(D.GT.0.0) THEN
1037             CANS=(A*(GAMMAINC(2.,X0,X1)-GAMMAINC(2.,X0,X2))+
1038             $      B*(GAMMAINC(1.,X0,X1)-GAMMAINC(1.,X0,X2)))/SQRT(D)
1039             TFUN=REAL(CANS)
1040             GO TO 999
1041         ELSE
1042             IF(D.EQ.0.) STOP
1043             TFUN=0.
1044             GO TO 999
1045         ENDIF
1046     ELSE
1047         Y0=-D/C
1048         IF(C.GT.0.0) THEN
1049             SIG1=AMAX1(Y0,X1)-Y0
1050             SIG2=AMAX1(Y0,X2)-Y0
1051             IF(SIG1.EQ.SIG2) THEN
1052                 TFUN=0.
1053                 GO TO 999
1054             ELSE
1055                 X3=Y0+X0
1056                 CANS=(1./SQRT(C))*
1057                 $      (A*(GAMMAINC(1.5,X3,SIG1)-GAMMAINC(1.5,X3,SIG2))+
1058                 $      (A*Y0+B)*(GAMMAINC(0.5,X3,SIG1)-
1059                 $      GAMMAINC(0.5,X3,SIG2)))
1060                 TFUN=REAL(CANS)
1061                 GO TO 999
1062             ENDIF
1063         ELSE
1064             SIG1=AMIN1(Y0,X1)-Y0
1065             SIG2=AMIN1(Y0,X2)-Y0

```

```

1066             IF(SIG1.EQ.SIG2) THEN
1067                 TFUN=0.0
1068                 GO TO 999
1069             ELSE
1070                 X3=Y0+X0
1071                 CANS=((0.,1.)/SQRT(-C))*
1072                     $      (A*(GAMMAINC(1.5,X3,SIG1)-GAMMAINC(1.5,X3,SIG2))+
1073                     $      (A*Y0+B)*(GAMMAINC(0.5,X3,SIG1)-
1074                     $      GAMMAINC(0.5,X3,SIG2)))
1075                 TFUN=REAL(CANS)
1076                 GO TO 999
1077             ENDDIF
1078         ENDIF
1079     ENDIF
1080 C
1081     999 RETURN
1082     END
1083 C
1084 C
1085 C
1086 C
1087     FUNCTION UFUN(L,A,B)
1088     $
1089     C
1090     C.....THIS FUNCTION COMPUTES THE REAL PART OF THE INTEGRAL FROM
1091     C ZERO TO INFINITY OF T**L*EXP(-T)*SQRT(A*T-B) WITH RESPECT
1092     C TO T. BOTH A AND B ARE REAL, WHILE L MUST BE AN INTEGER.
1093     C THIS ROUTINE CALLS THE COMPLEX INCOMPLETE GAMMA FUNCTION
1094     C ROUTINE, GAMMAINC.
1095     C
1096     COMPLEX GAMMAINC,SUM,TERM
1097 C
1098     IF(A.EQ.0.) THEN
1099         IF(B.GE.0.) THEN
1100             UFUN=0.
1101             GO TO 999
1102         ELSE
1103             UFUN=SQRT(-B)*GAMMA(L+1.)
1104             GO TO 999
1105         ENDIF
1106     ENDIF
1107     IF(B.EQ.0.) THEN
1108         IF(A.GE.0.) THEN
1109             UFUN=SQRT(A)*GAMMA(L+1.5)
1110             GO TO 999
1111         ELSE
1112             UFUN=0.
1113             GO TO 999
1114         ENDIF
1115     ENDIF
1116     Y0=B/A
1117     IF(A.GT.0.) THEN
1118         SIG=AMAX1(0.,Y0)-Y0
1119         SUM=0.
1120         DO 100 J=0,L
1121             TERM=GAMMAINC(J+1.5,0.,SIG)*Y0**(L-J)/
1122             $      (GAMMA(J+1.)*GAMMA(L-J+1.))
1123             SUM=SUM+TERM
1124             100 CONTINUE
1125             SUM=SQRT(A)*EXP(-Y0)*GAMMA(L+1.)*SUM

```



```

1186         IF(X.GE.0.0) THEN
1187             IF(M.GE.1) THEN
1188                 IF(M.EQ.1) THEN
1189                     GAMMAINC=SQRTPI*ERFC(X0,XSQ)
1190                     GO TO 999
1191                 ELSE
1192                     SUM=0.0
1193                     DO 200 J=0,MN302
1194                         RJ=J
1195                         SUM=SUM+X**((RJ+0.5)/GAMMA(RJ+1.5)
1196 200                     CONTINUE
1197                     GAMMAINC=GAMMA(A)*(EXP(-X0-X)*SUM+ERFC(X0,XSQ))
1198                     GO TO 999
1199                 ENDIF
1200             ELSE
1201                 STOP
1202             ENDIF
1203         ELSE
1204             IF(M.GE.1) THEN
1205                 IF(M.EQ.1) THEN
1206                     GAMMAIR=SQRTPI*EXP(-X0)
1207                     GAMMAII=-2.*EXP(-X0-X)*DAWSON(XSQ)
1208                     GAMMAINC=CMPLX(GAMMAIR,GAMMAII)
1209                     GO TO 999
1210                 ELSE
1211                     SUM=0.0
1212                     DO 300 J=0,MN302
1213                         RJ=J
1214 300                     SUM=SUM+((-1)**J*XAB**((RJ+0.5)/GAMMA(RJ+1.5)
1215                     CONTINUE
1216                     GAMMAIR=GAMMA(A)*EXP(-X0)
1217                     GAMMAII=GAMMA(A)*EXP(-X0-X)*(SUM-2.*DAWSON(XSQ)/
1218 8                     SQRTPI)
1219                     GAMMAINC=CMPLX(GAMMAIR,GAMMAII)
1220                     GO TO 999
1221                 ENDIF
1222             ELSE
1223                 STOP
1224             ENDIF
1225         ENDIF
1226     ENDIF
1227 C
1228 999 RETURN
1229 END
1230 C
1231 C
1232 C
1233 C *****
1234 C FUNCTION GAMMA(X)
1235 C *****
1236 C
1237 C.....THIS FUNCTION COMPUTES THE GAMMA FUNCTION FOR HALF-INTEGER
1238 C OR FOR POSITIVE INTEGER ARGUMENTS. FOR NON-POSITIVE INTEGER
1239 C ARGUMENTS, A DIVIDE BY ZERO OCCURS. THE INTEGER OR HALF-
1240 C INTEGER CLOSEST TO THE REAL ARGUMENT X IS TAKEN AS THE
1241 C DESIRED ARGUMENT FOR THE GAMMA FUNCTION.
1242 C
1243 C DATA SQRTPI/1.77245385090552/
1244 C
1245 C PNHALF=SIGN(0.5,X)

```

```

1246      M=2.*X+PMHALF
1247      ZFIN=X-0.5
1248      IF (MOD(M,2).EQ.0) THEN
1249          PRODUCT=1.
1250          Z=1.
1251      ELSE
1252          PRODUCT=SQRTPI
1253          Z=PMHALF
1254      ENDIF
1255      IF (M.EQ.1) GO TO 300
1256      IF (M.LE.0) GO TO 200
1257      100 PRODUCT=PRODUCT*Z
1258          Z=Z+1.
1259          IF (Z.LT.ZFIN) GO TO 100
1260          GO TO 300
1261      200 PRODUCT=PRODUCT/Z
1262          Z=Z-1.
1263          IF (Z.GT.ZFIN) GO TO 200
1264      300 GAMMA=PRODUCT
1265      C
1266      999 RETURN
1267      END
1268      C
1269      C
1270      C
1271      C
1272      FUNCTION ERFC(X0,X)
1273      C
1274      C
1275      C.....THIS FUNCTION COMPUTES THE COMPLEMENTARY ERROR FUNCTION
1276      C      OF X, DEFINED AS THE INTEGRAL FROM X TO INFINITY OF
1277      C      EXP(-X0-T**2) WITH RESPECT TO T. IT HAS THE CORRECT ASYMP-
1278      C      TOTIC BEHAVIOR AS X GOES TO INFINITY.
1279      C
1280      DIMENSION C(20)
1281      DATA P/1.8050171E-1/
1282      DATA C/1.0183718E-1, 1.0183718E-1, 1.0017821E-1, 9.6860268E-2,
1283      $      9.1964428E-2, 8.5652845E-2, 7.8162144E-2, 6.9790212E-2,
1284      $      6.0877146E-2, 5.1781843E-2, 4.2856408E-2, 3.4420921E-2,
1285      $      2.6741277E-2, 2.0012497E-2, 1.4349346E-2, 9.7851744E-3,
1286      $      6.2788185E-3, 3.7283429E-3, 1.9895034E-3, 8.9625580E-4/
1287      C
1288      T=1./(1.+P*X)
1289      SUM=C(20)
1290      DO 100 I=19,1,-1
1291          SUM=SUM*T+C(I)
1292      100 CONTINUE
1293      ERFC=T*SUM*EXP(-X0-X**2)
1294      C
1295      999 RETURN
1296      END
1297      C
1298      C
1299      C
1300      C
1301      FUNCTION DAWSON(X)
1302      C
1303      C
1304      C.....THIS FUNCTION COMPUTES THE DAWSON'S INTEGRAL OF X, DEFINED AS
1305      C      EXP(-X**2) TIMES THE INTEGRAL FROM 0 TO X OF EXP(T**2) WITH

```

```

1306 C RESPECT TO T. IT HAS THE CORRECT ASYMPTOTIC BEHAVIOR AS X
1307 C GOES TO INFINITY, AND THE CORRECT POWER SERIES BEHAVIOR TO
1308 C ONE PART IN 10**7 AS X GOES TO ZERO. IN BETWEEN, IT IS GOOD
1309 C TO SIX SIGNIFICANT FIGURES.
1310 C
1311 C DIMENSION C(20)
1312 C DATA P/6.3588589E-2/
1313 C DATA C/3.2697117E-2, 3.7706927E-2, -1.6544347E-2, 3.9386218E-1,
1314 $ -1.2920521E00, 2.4189490E00, -4.1029331E-1, -4.2650118E00,
1315 $ 1.1536146E00, 1.1004229E01, -6.0822008E00, -1.1462535E01,
1316 $ 2.9762841E00, 5.1527746E00, 1.6053263E01, -8.9967799E00,
1317 $ -1.7754438E01, 1.0590269E01, 4.3500041E00, -2.9155930E00/
1318 C
1319 C SUM0=0.5*P
1320 C T=1./((1.+P**X**2)
1321 C SUMAI=C(20)
1322 C DO 100 I=19,1,-1
1323 C SUMAI=SUMAI*T+C(I)
1324 C 100 CONTINUE
1325 C DAWSON=X*T*(SUMAI*T+SUM0)
1326 C
1327 C 999 RETURN
1328 C END
1329 C
1330 C
1331 C
1332 C

```


REFERENCES

1. R. A. Jong, MFTF-B Performance Calculations, Lawrence Livermore National Laboratory, Report UCID-19621 (1982), p. 13.
2. M. E. Rensink, private communication (August 1983).
3. B. M. Boghosian, private communication (July 1983).
4. J. M. Gilmore, private communication (1983).
5. NAG Fortran Library Manual, Mark 6, Volume 1, C05NAF, NAGFLIB: 1331/384:ME5:Jan. 76.
6. R. H. Cohen, "Axial Potential Profiles in Thermal-Barrier Cells," Nucl. Fusion 21 (2), 209 (1981).
7. B. M. Boghosian, Lawrence Livermore National Laboratory, unpublished notes (January 7 through May 11, 1983).
8. R. H. Cohen, Potentials in Thermal Barriers with Strong Electron Cyclotron Heating, Lawrence Livermore National Laboratory, Livermore, CA, Rept. UCRL-88939 (1983).
9. B. M. Boghosian, Lawrence Livermore National Laboratory, unpublished notes (February 15, 1983).
10. B. M. Boghosian, Lawrence Livermore National Laboratory, unpublished notes (March 31, 1983).
11. R. A. Jong, MFTF: A Computer Program for Calculating the MARS-Mode Operating Parameters for MFTF-B, Lawrence Livermore National Laboratory, Report UCID-19611 (1983).
12. R. A. Jong, Lawrence Livermore National Laboratory, private communication (1983).
13. B. M. Boghosian, Lawrence Livermore National Laboratory, unpublished notes (June 9, 1983).
14. J. W. Shearer, Doppler-Shifted Resonance Absorption by Hot Electrons, Lawrence Livermore National Laboratory, Livermore, CA, Rept. UCRL-86780 (1981).