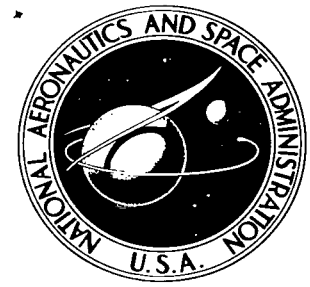
NASA TECHNICAL NOTE

NASA TN D-2959



NASA TN D-2959

MONTE CARLO METHOD FOR THE CALCULATION OF TRANSPORT PROPERTIES IN A LOW-DENSITY IONIZED GAS

by Charles M. Goldstein

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • SEPTEMBER 1965

NASA TN D-2959

MONTE CARLO METHOD FOR THE CALCULATION OF TRANSPORT PROPERTIES IN A LOW-DENSITY IONIZED GAS

By Charles M. Goldstein

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – Price $3.00

CONTENTS

MONTE CARLO METHOD FOR THE CALCULATION OF TRANSPORT PROPERTIES IN A LOW-DENSITY IONIZED GAS

by Charles M. Goldstein

Lewis Research Center

SUMMARY

An introduction to the general Monte Carlo method is presented, along with a discussion of its scope of application to plasma physics. This is followed with a heuristic sketch of the method. The problem of electron flow through a perfect Lorentzian gas in a parallel-plane diode is then formulated. The Monte Carlo solution is discussed in detail along with the relevant computational techniques employed. (Pertinent concepts from the theory of random variables are included in an appendix.)

The effects of mean free path on current-voltage characteristics, density distribution, and potential distribution are presented for two cases - monoenergetic and thermionic emission. Results indicate that electron-neutral elastic collisions can have a significant effect on the current-voltage characteristics for electrode separations as small as one mean free path in the case of thermionic emission, and one-half mean free path in the case of monoenergetic emission.

INTRODUCTION

A major difficulty in the study of low-density ionized gases is the lack of suitable analytical methods for determining the effects of collisions. "Low density" is here defined as those situations in which a characteristic dimension is of the order of a few mean free paths, that is, the regime wherein neither a collisionless nor a continuum approximation can be expected to represent the actual situation. This regime is of importance in low-pressure thermionic diodes, plasma sheaths (probe theory), ion engines, and cross-section measurements.

Much effort has been expended to obtain solutions of the Boltzmann transport equation for low-density neutral gases (ref. 1). Little work has been done on the extension of these methods to low-density ionized gases. Recently, however, Sockol

(ref. 2) has succeeded in numerically integrating the Boltzmann transport equation for a particular low-density ionized gas problem. Unfortunately, the numerical integration is very difficult for even the simplest hard-sphere collision cross section; the feasibility of extending this method for more complex collision cross sections has not as yet been investigated.

This report presents a new method for determining analytically the transport properties in a low-density ionized gas for an arbitrary collision cross section. Results with this method are given for two electron transport problems. The method proposed is, essentially, a consistent-field Monte Carlo method. Since the Monte Carlo method has not been widely applied in the fields of plasma physics or ionized gases, a brief introduction is presented; a review of the pertinent random variable theory is given in appendix A. The ability to use this method effectively is strongly dependent on numerical procedures and "tricks of the trade." A section is therefore included on the various computing techniques; in addition, a complete program listing plus selected flow charts are to be found in appendix C. A discussion of two important computing programs is presented by their author, H. Renkel, in appendix B.

The general method of calculating transport properties is applied herein to the problem of electron transport in a perfect Lorentzian gas with a hard-sphere collision cross section; in particular, the method is employed to obtain electron flux characteristics in a plane-parallel diode including the effect of electron-neutral collisions.

Langmuir (ref. 3) published the first correct solution to the effect of space charge and initial velocities on the potential distribution and thermionic current between parallel-plane electrodes for no collisions (vacuum diode). He also studied (ref. 4) the problem of diffusion of electrons back to the emitter for the case of a very small mean free path. These results are extended herein to the case of electron-neutral collisions for which the mean free path is not necessarily small with respect to the interelectrode separation.

MONTE CARLO METHOD

General Description

The Monte Carlo method is, in general terms, a technique for solving physical and mathematical problems by using random sampling. Although the term "Monte Carlo method" has been subjected to various interpretations, an acceptable statement of the method as applied herein has been given by Donsker and Kac (ref. 5): "The Monte Carlo

approach consists in permitting a 'particle' to play a game of chance, the rules of the game being such that the actual deterministic and random features of the physical process are step by step exactly imitated by the game. By considering very large numbers of particles, one can answer such questions as the distribution of the particles at the end of a certain period of time, the number of particles to escape through a shield of a specific thickness, etc. One important characteristic of the preceding approach is that the functional equation describing the diffusion process is bypassed completely, the probability model used being derived from the process itself."

A short history of Monte Carlo applications is to be found in the paper by Goertzel and Kalos (ref. 6). An excellent review of the basic principles is given in reference 7, and an extensive bibliography has recently been compiled by Kraft and Wensrich (ref. 8). This method has, in recent years, been employed with considerable success to a wide variety of problems, most notably in the area of nuclear shielding problems (viz., neutron transport). These latter problems are linear in the sense that the neutron trajectories are independent of the neutron density. More recently, the method has been extended to certain nonlinear problems in radiation transport (ref. 9).

There have been some applications of the Monte Carlo methods to the investigation of one-dimensional electron (ion) diodes, but these studies are more often referred to as computer-simulated solutions, or "computer experiments." The difference in terminology reflects the fact that these studies approximate the physical model by a finite number of current sheets, which are then followed deterministically through all mutual interactions by the computer. The Monte Carlo method, on the other hand, most frequently implies repeated, stochastically independent trials. A short history of the aforementioned computer experiments is to be found in the paper by Burger (ref. 10). These studies do not take collisions into account, nor does it seem practical to do so because of the demands this type of analysis would impose on computer storage requirements.

Itoh and Musha (ref. 11) employed a Monte Carlo calculation to determine the ionization and excitation coefficients of electrons in a uniform electric field E for given gas pressure P. They also computed drift velocity and mean energy for several values of E/P. Although the authors state that this method can be extended to strong, nonuniform electric fields, it cannot provide a suitable model from which diode characteristics could be obtained since space-charge effects, which introduce a nonlinearity, have not been considered.

Just as the nonlinearity in the radiation transport problem is characterized by a single parameter, the temperature (ref. 9), so the nonlinearity in the charge-particle transport problems is characterized by the potential. Unlike the photons in the former problems, however, the charged particles experience a body force proportional to the first derivative of the potential.

Heuristic Sketch of the Method

First the relatively simple problem of the attenuation of a molecular beam by a homogeneous gas shall be considered. If the actual experiment is performed for a given emission flux Γ_o and the flux reaching the target (or collector) Γ_c is measured, it is reasonable to interpret the ratio Γ_c/Γ_o as the probability that a unit of emitted flux will reach the target. If a knowledge of the scattering probabilities of a single molecule passing through the same gas is assumed, it is possible on a computer to follow a certain number N_o, one at a time, and tally the number N_c that reach the collector (the others are scattered back to the emitter). Then the ratio N_c/N_o would be an approximation to the experimentally determined Γ_c/Γ_o. Since the experimental fluxes may be of the order of 10^{18} particles per square centimeter per second or higher, it is not conceivable, even in this relatively simple situation, to "do the experiment" on the computer. As N_o becomes larger, however, the approximation N_c/N_o becomes better. Statistical analysis provides a means of estimating how good the approximation is.

For the case of charged particles flowing through a gas, the situation is complicated by the nonlinearity introduced by the space charge. That is, the flow of charged particles is not only influenced by collisions with the gas molecules, but also by the potential field; the potential field, itself, is a function of the density of charged particles. This is the situation considered herein. To start, for a given collector potential $\mathscr{V}(L)$ a potential distribution $\mathscr{V}(x)$ is assumed. An approximation to the current reaching the collector, N_c/N_o, is then obtained as in the molecular beam case. In addition, however, the contribution to the density, at preselected data points, of each charged particle is also tallied. These densities are used to solve Poisson's equation for a new potential distribution. The process is then repeated (i.e., iteration is performed on the potential distribution) until the potential distribution "converges." Convergence must here be considered only in a statistical sense; when further iterations produce only random fluctuations in the potential distribution, "convergence" is assumed. Random fluctuations are, of course, to be expected, since only a small statistical sample N_o of the total flux is considered. After convergence has been achieved, succeeding iterations may be considered, in the parlance of statistical analysis, as independent trials; each resulting approximation N_c/N_o may be considered a sample mean (appendix A). An analysis of the sample means provides a way of estimating the accuracy of the result.

FORMULATION OF ELECTRON DIODE PROBLEM

The physical model of an infinite parallel-plate diode is depicted in figure 1. In the same figure the types of scattering that may occur for a monotonic potential distribution

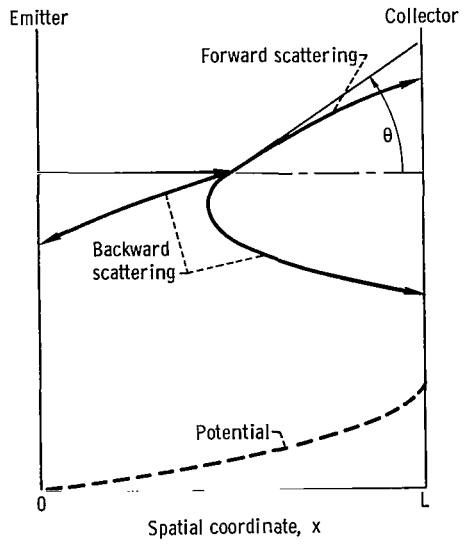


Figure 1. - Diode model and types of scatter.

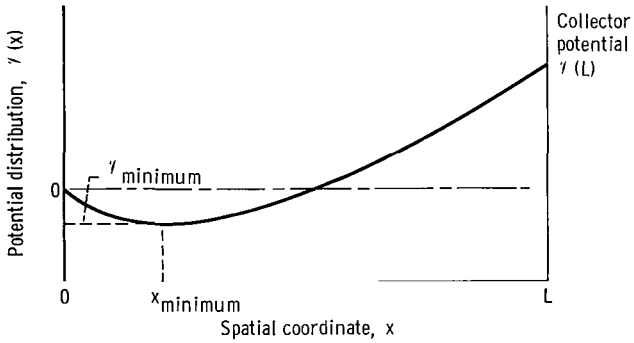


Figure 2. - General potential distribution.

are shown. In figure 2 a typical potential distribution is shown. When a potential minimum exists as indicated in this figure, a certain portion of thermionically emitted electrons will be rejected back to the emitter even in the absence of collisions. The existence of a potential minimum less than both emitter and collector potentials defines the space-charge-limited regime of diode operation.

The perfect Lorentzian gas assumption implies an infinite mass target particle, and hence the laboratory system becomes equivalent to the center of mass system. Since hard-sphere collisions result in isotropic scattering in the center of mass system, the equivalence of the two systems in this case results in isotropic scattering in the laboratory system.

Isotropic scattering means, by definition, that the probability of scattering into unit solid angle is the same for all angles. The probability distribution function (hereafter, p.d.f., see appendix A) is therefore the constant $1/4\pi$. Hence the probability of scattering into solid angle $d\Omega$ is $d\Omega/4\pi$. In terms of the scattering angle θ, this becomes

$$\frac{d\Omega}{4\pi} = \frac{\sin\theta}{2}\, d\theta \tag{1}$$

Consequently, the p.d.f. of scattering into angle $d\theta$ is simply $\sin\theta/2$.

The assumption of hard-sphere collisions also implies a constant mean free path λ. Now if that group of electrons that has just collided is considered, then the fraction of these electrons that will suffer collisions in distance ℓ_c is (ref. 12, p. 102, eq. (98a))

$$1 - e^{-\alpha \ell_c} \tag{2}$$

where

$$\left.\begin{aligned} \ell &\equiv s/L \\ \alpha &\equiv L/\lambda \end{aligned}\right\} \tag{3}$$

and s is the path length and L the electrode spacing.

Thermionic Emission

For this case it is assumed that electrons are emitted with a half-Maxwellian velocity distribution

$$f(u, V)\, du\, dV = (4/\sqrt{\pi})Ve^{-(u^2 + V^2)}\, du\, dV \tag{4}$$

where

$$\left.\begin{aligned} u &\equiv v_x/\sqrt{2kT/m} \\ V &\equiv \frac{\sqrt{v_y^2 + v_z^2}}{\sqrt{2kT/m}} \end{aligned}\right\} \tag{5}$$

The one-dimensional Poisson's equation, in dimensionless variables, becomes

$$\varphi''(y) = C\, n(y) \tag{6}$$

where

$$\left.\begin{aligned} y &\equiv \frac{x}{L} \\ \varphi &\equiv \frac{eV}{kT} \\ n &\equiv \frac{\hat{n}}{n_o} \end{aligned}\right\} \tag{7}$$

and

$$C = 8(\pi/2kT)^{3/2} m^{1/2} e J_o L^2 \tag{8}$$

Monoenergetic Emission

The analysis for monoenergetic emission directly parallels that for thermionic collision with a few minor changes. The dimensionless variables u, V, and φ are now defined as

$$\left.\begin{aligned} u &= \frac{v_x}{v_o} \\ V &= \frac{\sqrt{v_y^2 + v_z^2}}{v_o} \\ \varphi &= \frac{2eV}{mv_o^2} \end{aligned}\right\} \tag{9}$$

The constant parameter C in Poisson's equation (eq. (6)) becomes

$$C = \frac{8\pi e L^2 J_o}{mv_o^3} \tag{10}$$

MONTE CARLO SOLUTION

Thermionic Emission

Initial conditions. - It must be emphasized that the test "electrons" are not chosen from the half-Maxwellian distribution (eq. (4)). Although test "electrons" are mentioned, the statistics are obtained for units of electron flux - not units of charge. Hence, the initial velocities must be chosen from the distribution of flux in velocity space

$$\sqrt{\pi}\, uf(u, V)\, du\, dV = 4uVe^{-(u^2+V^2)}\, du\, dV \tag{11}$$

from equation (4). Since the u and V components of velocity are independent, the respective marginal distributions (see ref. 13, p. 287) $F_u(V)$ and $F_V(u)$ can be obtained:

$$F_u(V) \equiv \int_0^\infty du \int_0^V 4uVe^{-(u^2+V^2)}\, dV \tag{12}$$

$$F_V(u) = \int_0^\infty dV \int_0^u 4uVe^{-(u^2+V^2)}\, du \tag{13}$$

But these marginal distributions are simply the cumulative distribution functions (hereinafter, c.d.f.) for u and V, respectively. From equations (A16) and (A15),

$$\left.\begin{aligned} u^2 &= -\ln(R_u) \\ V^2 &= -\ln(R_V) \end{aligned}\right\} \tag{14}$$

where R_u and R_V are random numbers between 0 and 1. Equations (14) are then used to determine the initial velocities of each test electron.

Distance to collisions. - The distance to collision must be obtained at the start of each new electron trajectory (i.e., on emission from emitter or after a collision).

Equation (2) can also be interpreted as the probability that an electron will suffer a collision in a distance $\ell \leq \ell_c$. This, however, is just the definition of the c.d.f. $F(\ell_c)$ (see appendix A). Hence, from equation (A16) can be obtained a relation between the random numbers R_ℓ and the distribution of path length to collision:

$$R_\ell = 1 - e^{-\alpha \ell_c} \tag{15}$$

where

$$\begin{aligned} \ell_c &= -\left(\frac{1}{\alpha}\right) \ln(1 - R_\ell) \\ &= -\left(\frac{1}{\alpha}\right) \ln(R_\ell) \end{aligned} \tag{16}$$

Scattering angle. - If a collision takes place, then the scattering angle θ must be

determined. From the p.d.f. (eq. (1)), the c.d.f. $F(\theta)$ can immediately be obtained (compare eq. (A3) in appendix A):

$$F(\theta) = \frac{1 - \cos \theta}{2} \tag{17}$$

But in this case the c.d.f. can take on the values $-1 \leq F(\theta) \leq 1$ (forward and backward scattering). Hence, in order to choose randomly from this range (see eq. (A16)), let

$$\cos \theta = 1 - 2R_\theta \tag{18}$$

where once again $0 \leq R_\theta \leq 1$. Equation (18) is the final result since only $\cos^2\theta$ (and $\sin^2\theta = 1 - \cos^2\theta$) is of interest in the actual computations.

Charge density. - The data points y_i are selected by the curve-fitting subroutine (see appendix B where y_i corresponds to the arguments x_α). The contribution of the k^{th} test electron (unit of flux) of velocity $u_k(y)$ to the charge density at each y_i is

$$n_k(y_i) = \frac{1}{\sqrt{\pi}\, u_k(y_i)} \tag{19}$$

where

$$u(y) = \sqrt{u_o^2 + \varphi(y) - \varphi(y_o)} \tag{20}$$

y_o is the position of the last "event" (collision or emission), and u_o is the initial velocity immediately after the last event (i.e., at the beginning of a new trajectory).

The tallied density at a data point y_i for a total of N_o histories is then

$$n(y_i) = \frac{1}{\sqrt{\pi}\, N_o} \sum_k \frac{1}{u_k(y_i)} \tag{21}$$

where the sum over k may be greater than, equal to, or less than N_o because of collisions and turning points in the potential field.

Current to collector. - The ratio of current density to the collector J to the emitted current density J_o for each iteration is computed from the relation

$$\frac{J}{J_o} = \frac{N_c}{N_o} \tag{22}$$

where N_c is the number of test electrons reaching the collector.

Monoenergetic Emission

The solution for monoenergetic emission is exactly the same as for thermionic emission with two exceptions. First, instead of choosing from an initial distribution of velocities the initial conditions are, for every electron, $u = 1$ and $V = o$, and second, the density for N_o histories becomes (cf. eq. (14))

$$n(y_i) = \frac{1}{N_o} \sum_k \frac{1}{u_k(y_i)} \tag{23}$$

COMPUTATIONAL METHODS

The original program was based on the assumption that Monte Carlo computations would be limited to only a few collisions because of the requirement of reasonable computor execution times. Hence, this program was optimized for $L/\lambda < 1$. The results proved this assumption overly pessimistic, but the program was not revised for the present report.

Evaluation of Potential and Potential Minimum

After a curve fit of the density is obtained (subroutine CHEBY, appendix B), the density distribution is given by a power series in y:

$$n(y) = a_o + a_1 y + a_2 y^2 + \ldots + a_n y^n$$

$$= \sum_{j=0}^{k} a_j y^j \tag{24}$$

where k is the degree of the fit.

The potential is obtained by substituting equation (24) in Poisson's equation (eq. (6)) and integrating n(y) term by term:

$$\varphi(y) = c_o + cy + C\sum_{j=0}^{k} \frac{a_j}{(j+1)(j+2)} y^{j+2} \tag{25}$$

But since

$$\varphi(o) = 0$$

equation (25) can be written

$$\varphi(y) = \sum_{j=1}^{k+2} c_j y^j \tag{26}$$

$$\left.\begin{aligned} c_j &= \frac{a_{j-2}}{j(j-1)} \qquad i \geq 2 \\ c_1 &= \varphi(1) - \sum_{j=2}^{k+2} c_j \end{aligned}\right\} \tag{27}$$

After the a_j have been determined (subroutine CHEBY), the c_j are computed in subroutine COEF(2).

Originally, equation (26) was employed (with k usually equal to 10) each time $\varphi(y)$ was evaluated, but this proved too time consuming. For this reason it was decided to tabulate $\varphi(y)$ at the beginning of each iteration and use the tabulated values whenever possible. The interelectrode space was subdivided into 1024 regions, and the 1025 values of $\varphi(y)$ were tabulated in subroutine MINPHI. At the same time, φ was tested at each evaluation for the minimum value. Hence, the location of the potential minimum was ascertained within $\pm 1/2048$ of the interelectrode separation. In addition, φ was tabulated at data points y_i where the density was to be tallied. The results of tabulating the potential was an eight-fold (and greater) decrease of execution time.

Choosing from the Distribution $e^{-X}\,dx$

It is pointed out in appendix A that choosing random values X_k from the distribution whose p. d. f. is $e^{-X}\,dx$ is equivalent to choosing random numbers R_k from the uniform distribution (eq. (A7)) and using equation (A15):

$$X_k = -\ln(R_k) \tag{28}$$

In the present problem, it is possible to identify the random variables u^2 and V^2 with X, and ℓ_c with $(1/\alpha)X$ (eqs. (14) and (16), respectively). The random numbers R_k are obtained from a pseudo-random-number generator of the congruence-method type (ref. 5). This random generator is part of the computor library here at Lewis Research Center.

Although the desired random variable can be obtained directly from equation (28), it was decided to tabulate the X_k instead. A table (1025 entries) was constructed of X_k (subroutine CUMVEL) at the beginning of the program. The table look-up is five times as fast as employing equation (28) each time.

Location of Collision

If a distance to collision ℓ_c is given, the location of the collision y_c is obtained by solving

$$\ell_c = \int_{y_o}^{y_c} \sqrt{1 + \frac{V^2}{u_o^2 + \varphi(y') - \varphi(y_o)}}\, dy' \tag{29}$$

Two methods were used to minimize the number of times the integrand, and specifically $\varphi(y)$, need be evaluated. First, Simpson's rule was used in a search routine to allow the use of the tabulated values of $\varphi(y)$ (see Evaluation of Potential and Potential Minimum section, p. 10), and then the step size (in the use of Simpson's rule) was made to depend on the ratio V^2/u^2.

The procedure employed for obtaining a reasonable step size can be best explained by an example. Assume that y_c falls between any two points y_o and y_f. For a straightforward application of Simpson's rule, three values are needed of the integrand in equation (29) at three equidistant values of y: y_1, y_2, y_3. Initially $y_1 = y_o$. The $\varphi(y)$ has already been tabulated at 1025 values of y given by

$$y_m = \frac{m}{1024} \qquad m = 0,\ 1,\ 2\ .\ .\ .\ 1024 \tag{30}$$

Consequently, y_o and y_f will always be selected equal to tabulated values of y_{m_o} and y_{m_f}. Hence, the first estimate of step size Δ in units of m is given by

$$\Delta = \left[\frac{m_f - m_o}{4}\right] \tag{31}$$

where [ ] refers to the integral value. A second estimate of step size (obtained as a result of trial and error computations) is given by

$$\Delta' = 2^{4-M}$$

where

$$M = \left[\log_{10}(V^2/u^2)\right]$$

Then, the step size is taken as the minimum of the two estimates.

If the value of Δ from equation (31) is zero (i.e., distance to collision is less than four steps), then Δ is set equal to

$$\Delta = \left[\frac{m_f - m_o}{2}\right]$$

If this should be zero, then the collision location y_c is arbitrarily set equal to y_{m_f}.

CONVERGENCE AND STANDARD DEVIATION

It was observed, during tests of the program, that convergence (in the statistical sense, p. 4) was obtained in the first few iterations. Since the succeeding iterations are treated as independent trials, the problem arises in a production run of just how to decide when convergence occurs. This was done in the following manner.

Each case (given anode potential) was run for a given number of iterations, for example, 15. At the end of each iteration the sample means $n(y_i)$ and J/J_o (see eqs. (21) to (23)) were stored. Each of these stored values is analogous to an experi-

TABLE I. - EFFECT OF VARIOUS PARAMETERS ON STANDARD DEVIATION AND EXECUTION TIME

Item	Type of emission	Electrode spacing to mean free path ratio, L/λ	Dimensionless collector potential, $\varphi(1)$	Current density ratio, J/J_o	Standard deviation, σ_J	Sample size	Number of iterations	Collisions (for one iteration)	Execution time, min
1	Mono-	0.1	0.75	0.961	0.001	5 000	10	483	2.42
2	ener-	↓	2.0	.972	.0012	5 000	10	503	2.46
3	getic	↓	2.0	.971	.0019	2 000	5	213	.51
4	↓	↓	4.0	.986	.0005	10 000	10	1 079	5.28
5	↓	.5	4.0	.918	.0015	2 000	10	1 257	2.48
6	↓	1.0	4.0	.839	.0015	1 000	10	1 483	2.09
7	Thermi-	5.0	12.0	.358	.009	1 000	15	8 475	17.7
8	onic	5.0	32.0	.672	.003	1 000	18	16 367	37.65
9	↓	.1	10.2	.942	.0016	1 000	10	147	1.49
10	↓	.1	10.2	.942	.0008	10 000	10	1 198	14.18

mental data point. Carrying the analogy further, at the end of 15 runs (iterations) there were sets of 15 data points for each of the sample means. If this were an experiment, it would be expected that each set of 15 data points would have a certain amount of "scatter" due to random error. In the present situation, however, the iterations before convergence will produce data points with a nonrandom error. The problem then becomes one of simply eliminating the iterations that introduce a nonrandom error. This was accomplished by obtaining the sample mean and standard deviation (see appendix A) of each set of 15 data points. Then from each set only those points were retained that were within three standard deviations of the sample mean. The final values of sample means and standard deviation (given in table I) were obtained from the remaining data points. In all cases, the number of iterations treated as independent trials was of the order of ten.

RESULTS

Thermionic Emission

The effect of mean free path on the current-voltage characteristic is shown in figure 3. The solid line, $L/\lambda = 0$, represents the collisionless solution of Langmuir (ref. 3). The Monte Carlo calculations indicated along this curve were undertaken as a check on the computer program. These particular results were obtained with 5000 histories per iteration and ten iterations. The execution time for each point on the curve varied between 2.5 and 4.0 minutes.

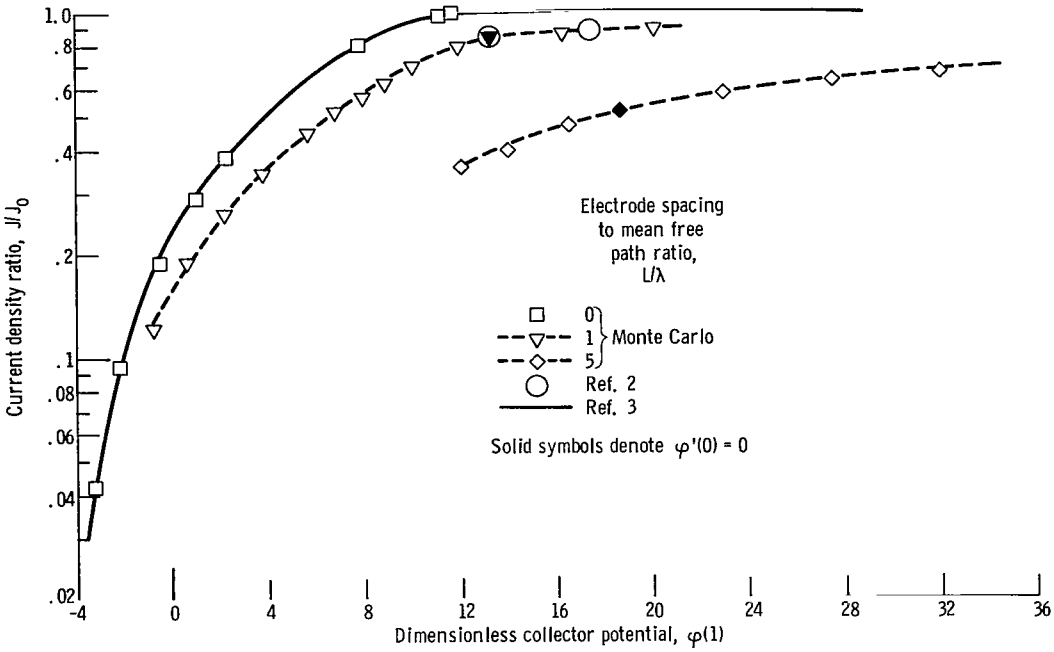


Figure 3. - Effect of mean free path on current-voltage characteristics for thermionic emission. Dimensionless constant C = 50.

The two solid data points on the curves for $L/\lambda = 1$ and 5 represent the conditions where the slope of the potential is zero at the emitter. The O's on the curve $L/\lambda = 1$ indicate the results of an independent solution of Boltzmann's transport equation for this problem (ref. 2).

The curve for $L/\lambda = 5$ was not extended to lower $\varphi(1)$ because of a loss in precision in the curve-fitting routine program (appendix B) used to fit the density distribution. A more flexible routine is being developed.

The effect of potential on the electron density distribution is shown in figure 4. From the emitter out to about one mean free path, the density of the higher energy electrons is less than that of the lower energy electrons as would be expected under conditions of no collisions. The actual decrease in the magnitude of the density at the emitter surface, however, indicates that in the higher potential case more of the backscattered electrons are being turned about by the potential field before reaching the emitter. This can be best understood by considering the effect of an accelerating potential field on the cone of capture at the emitter for backward scattering (see fig. 5). This cone of capture may be defined by a polar angle θ^*. It will suffice to consider a first collision whereby the electron has initial energy of $u_o^2 + V_o^2$ and the collision occurs at x_c. The magnitude of the x-component of velocity after scatter becomes

$$u^2 = \left[u_o^2 + V_o^2 + \varphi(x_c)\right]\cos^2\theta \tag{32}$$

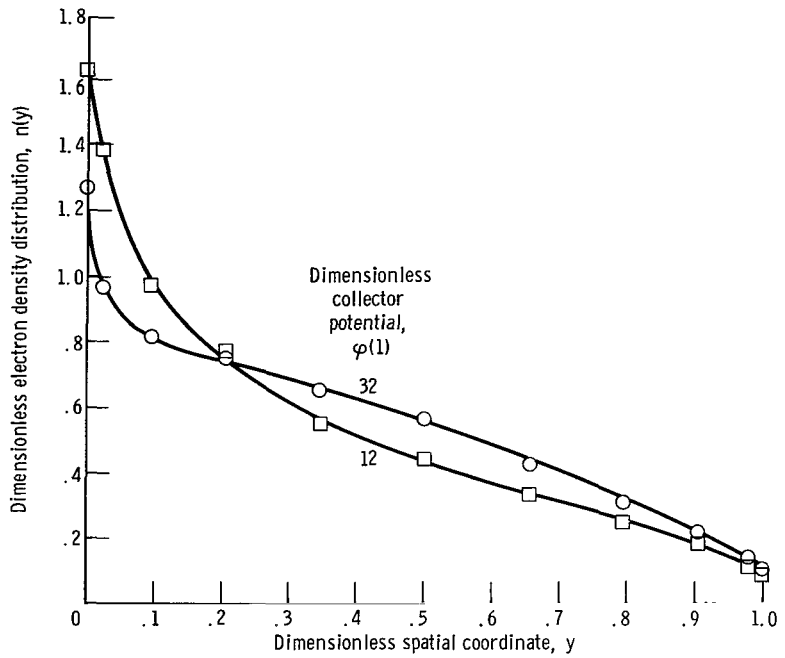


Figure 4. - Effect of anode potential on electron density distribution for thermionic emission. Dimensionless constant C = 50; electrode spacing to mean free path ratio L/λ = 5.

If the electron is to reach the emitter against the monotonic potential field $\varphi(x) \geq 0$, then u^2 must satisfy the condition

$$u^2 \geq \varphi(x_c) \tag{33}$$

When equation (33) is substituted into equation (32), θ^* is defined by

$$\cos^2 \theta^* = \frac{\varphi(x_c)}{u_o^2 + V_o^2 + \varphi(x_c)}$$

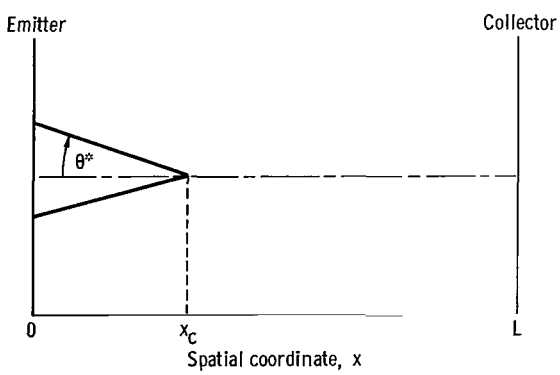


Figure 5. - Cone of capture at emitter.

or

$$\cos^2 \theta^* = \frac{1}{1 + \dfrac{u_o^2 + V_o^2}{\varphi(x_c)}} \tag{34}$$

Equation (34) shows that an increase in potential $\varphi(x_c)$ increases $\cos^2 \theta^*$ and reduces θ^*.

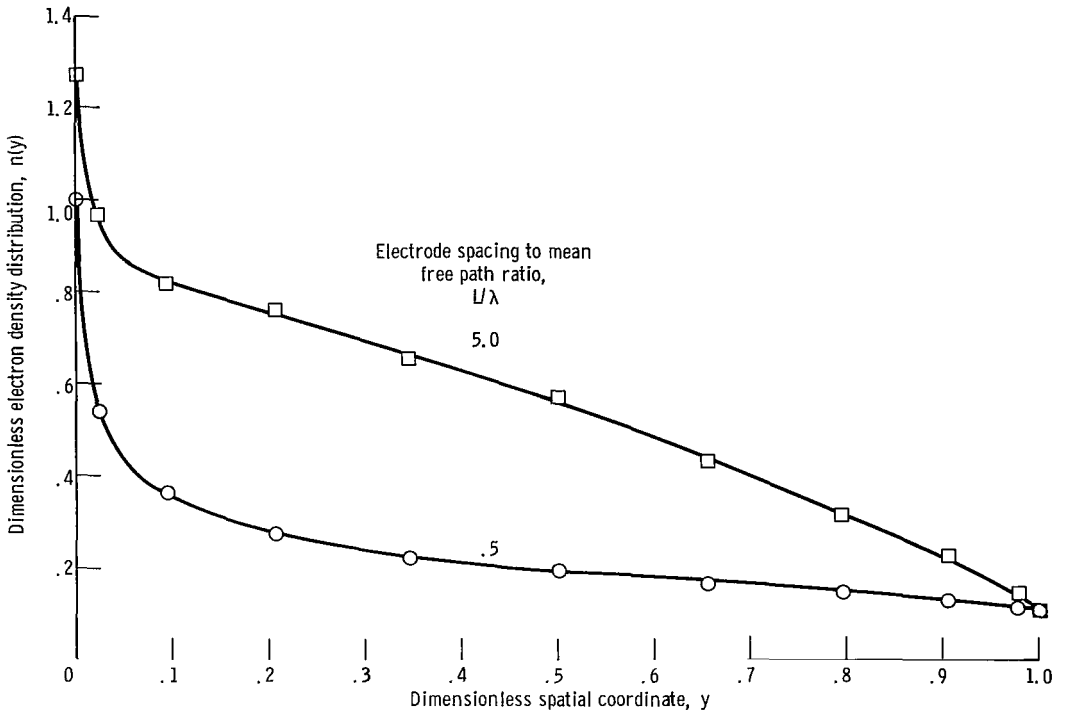


Figure 6. - Effect of mean free path on electron density distribution for thermionic emission. Dimensionless constant C = 50; dimensionless collector potential $\varphi(1) = 32$.

Thus, the higher the potential field, the smaller is the cone of capture at the emitter.

This same phenomenon accounts for the crossover in the curves of figure 4 away from the emitter. Since fewer of the backscattered electrons in the higher electric field case reach the emitter, this implies that more are turned about by the field. The presence of turning points in the electron trajectories affects the charge density in two ways. Since the u-component of velocity is zero at a turning point, the contribution to the charge density there is exceptionally high; in addition, the path length of an electron in the neighborhood of a turning point is much greater than the distance traveled normal to an electrode surface, these electrons suffer more collisions, and, hence, contribute more strongly to the charge density. This latter point is vividly illustrated by comparing the typical number of collisions per iteration for the two cases of figure 4 (items 7 and 8, table I, p. 14). In the low potential case ($\varphi(1) = 12$) over 8000 collisions were observed in one iteration, while for the high potential case ($\varphi(1) = 32$) over 16 000 collisions were observed. The increase in number of collisions accounts for the crossover in the two curves of figure 4 and the higher density for $y > 0.2$ in the case $\varphi(1) = 32$.

The effect of mean free path on the density and potential distributions for constant collector potential are shown in figures 6 and 7, respectively. As expected, the effect of collisions is to increase the charge density and, therefore, decrease the potential in the interelectrode space.

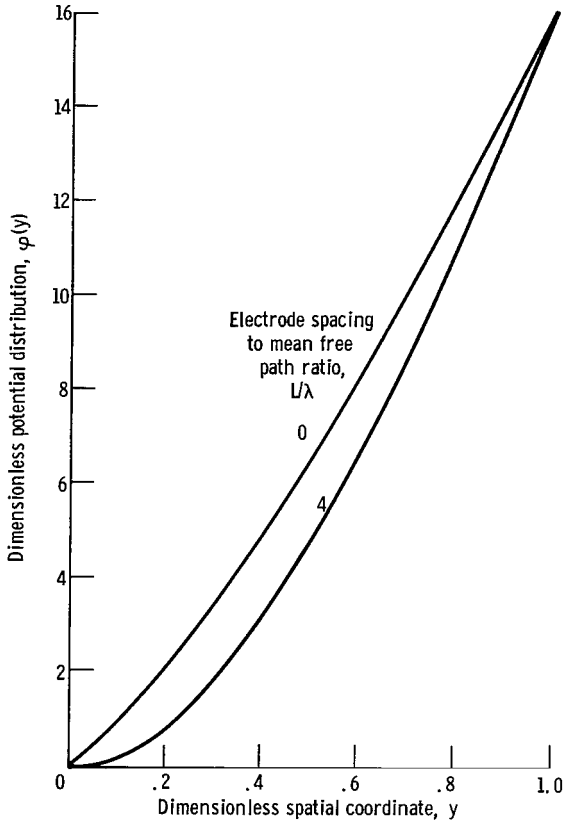


Figure 7. - Effect of mean free path on potential distribution for thermionic emission. Dimensionless constant C = 50.

Monoenergetic Emission

The corresponding diode characteristics for monoenergetic emission are shown in figures 8 to 11. The author has, at present, no hypothesis regarding the inflections observed in the current-voltage characteristics (fig. 8) for $L/\lambda = 0.5$ and 1.0. The points calculated are reproducible, and each point, as plotted, spans at least plus or minus two standard deviations about the mean J/J_o. The solid lines represent independent solutions of the Boltzmann equation for this problem in the limit of one collision (ref. 14).

Another noteworthy feature of the monoenergetic emission characteristics is the buildup of charge density in the interelectrode region as the potential is decreased (fig. 9). This increase in charge density is considerably enhanced by the appearance of a potential minimum (upper curve in fig. 9). The potential minimum causes more turning points to occur in the trajectories of the scattered

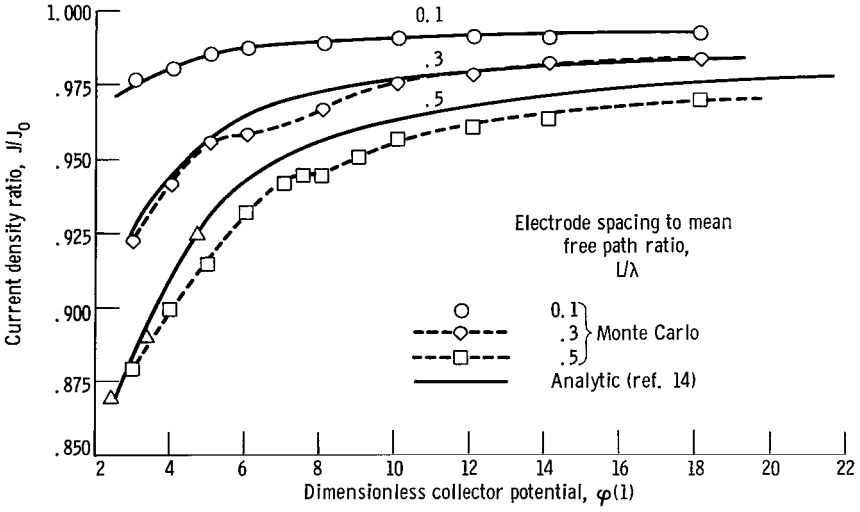


Figure 8. - Effect of mean free path on current-voltage characteristics for monoenergetic emission. Dimensionless constant C = 10.

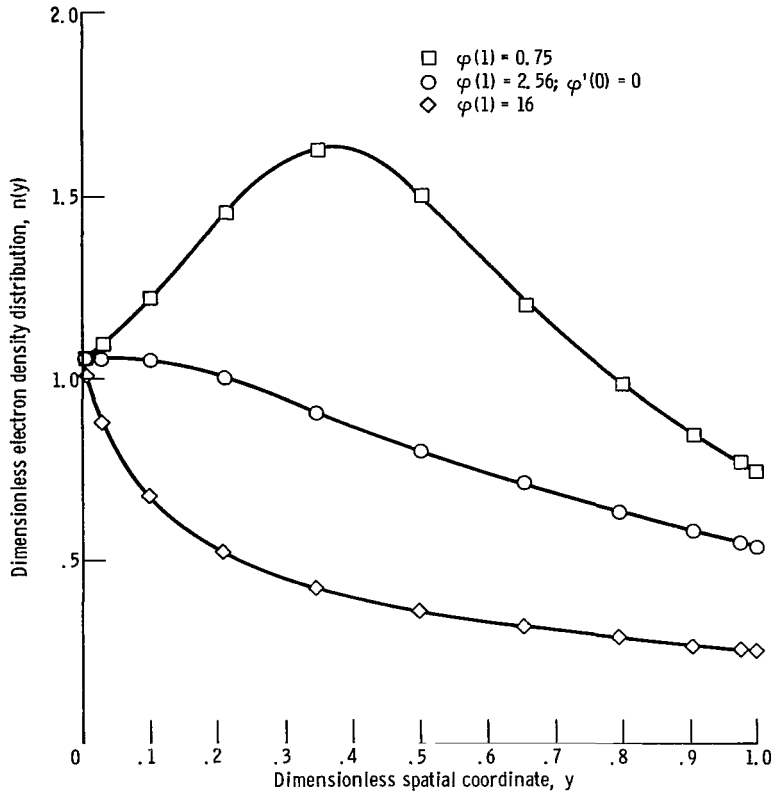


Figure 9. - Effect of anode potential on electron density distribution for electron beam. Dimensionless constant $C = 10/\sqrt{\pi}$; electrode spacing to mean free path ratio $L/\lambda = 0.1$.

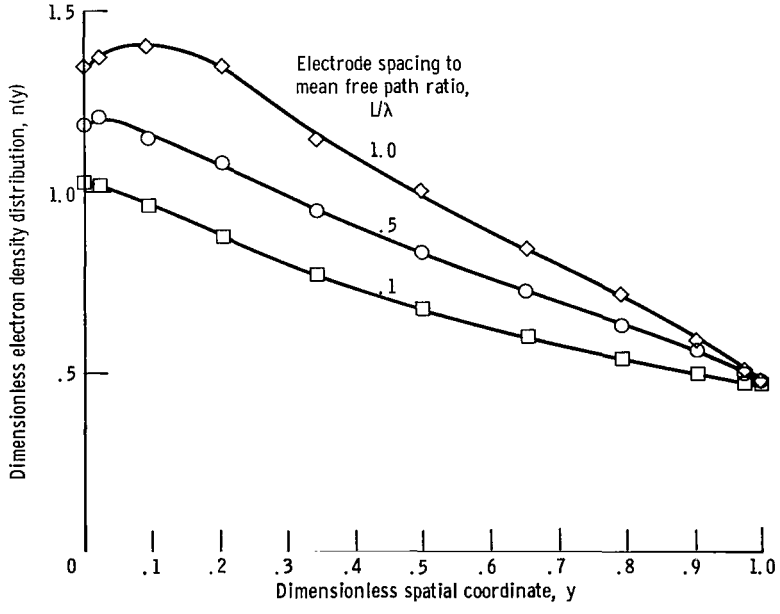


Figure 10. - Effect of mean free path on electron density distribution for electron beam. Dimensionless constant $C = 10/\sqrt{\pi}$; dimensionless collector potential $\varphi(1) = 4$.

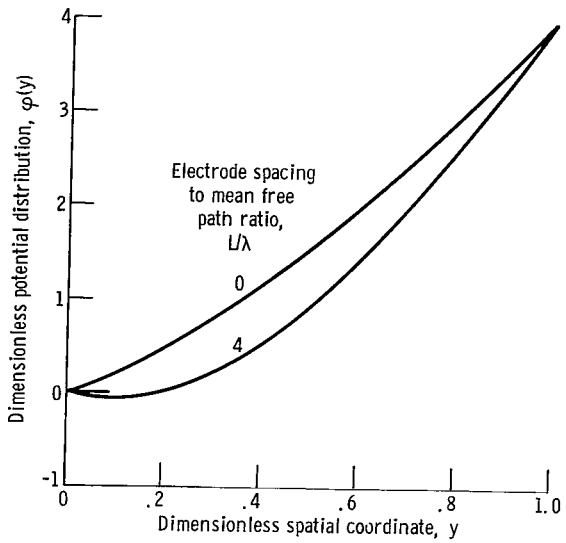


Figure 11. - Effect of mean free path on potential distribution for monoenergetic emission. Dimensionless constant $C = 10/\sqrt{\pi}$.

electrons. Since the u-component of velocity becomes zero at a turning point, the contribution to the charge density of electrons undergoing reflections in the potential field is exceptionally high.

DISCUSSION OF RESULTS

The agreement of the solution obtained by the method proposed in this report and the independent results of Sockol (ref. 2 and fig. 3, p. 15) and Goldstein and Goldstein (ref. 14 and fig. 8) is very gratifying indeed. Most encouraging, with regard to the extension of this method to other problems, are the statistics presented in table I (p. 14). These statistics show that the execution times needed to obtain reasonable standard deviations σ_J need not be excessive. In turn, they illustrate the effect of the consistent-field constraint (Poisson's equation) on the number of histories needed for good statistics (tens and hundreds of thousands of histories are generally required in other problems where this constraint is absent). It must be emphasized that the execution times illustrated in table I are not the minimum attainable, since no attempt has yet been made to incorporate any of the variance-reducing techniques discussed in the literature (ref. 15).

The execution times in the problems treated herein could most directly be decreased by a more extensive use of tabulated values (eliminating the Gaussian quadratures - hence, obviating completely the need to evaluate $\varphi(y)$ during an iteration) and by optimizing the number of tabulations needed (one may not need 1025 tabular values). In addition, for larger values of L/λ, it would be more appropriate to step along each trajectory from the emitter instead of first ascertaining if a collision has occurred in the interelectrode space as is done in the present case.

CONCLUDING REMARKS

A general method for the calculation of transport properties in a low-density ionized gas has been presented. This method has been applied to two cases of electron transport in a perfect Lorentzian gas. Excellent agreement has been demonstrated by two other independent investigations.

Although the particular applications of the method presented herein employ a hard-

sphere collision model, the great advantage in the Monte Carlo method lies in its inherent ability to provide similar solutions for any given collision model, theoretical or experimental. This includes inelastic, charge exchange, and ionizing collisions. This method is limited, however, to those cases where avalanche ionization does not occur; even in this latter case, however, the Monte Carlo method should be capable of providing the source intensities for the collision term in the Boltzmann equation for arbitrary cross sections, and, therefore, allow a numerical solution of the same.

This method should also be of value in the solution of plasma sheath problems, which are, in reality, just generalizations of the diode problem with different boundary conditions at the emitter and/or collector.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, May 14, 1965.

APPENDIX A

RANDOM VARIABLES

In this section there is interest only in continuous probabilities for which there exists a continuous function f(x), called the probability distribution function (hereinafter p. d. f.), such that

$$P[a \leq X \leq b] = \int_a^b f(x)dx$$

$$P[-\infty < X < \infty] = \int_{-\infty}^{+\infty} f(x)dx = 1 \tag{A1}$$

where

$$f(x) \geq 0 \qquad -\infty < X < \infty$$

To speak of a "random variable" X (instead of x) is really to define a mathematical point of view. This unambiguous point of view maintains no interest in the exact value of X but instead is only interested in inquiring about the probability of finding X in a certain region (of x-space).

For example, the case is considered where the probability density function is the nondimensionalized Maxwellian distribution of the x-component of flux (eq. (11) integrated over V):

$$\begin{aligned} f(u)du &= 2ue^{-u^2}du \qquad u \geq 0 \\ &= 0 \qquad\quad u < 0 \end{aligned} \tag{A2}$$

In the present analysis, the concern is not for a knowledge of a particular value of u, but rather to determine just what the probability is that a random variable U lies in the range u, u + Δu.

In the study of a random variable X the function $F_X(x)$ is of great importance:

$$F_X(x) = P[X \leq x] \equiv \int_{-\infty}^{x} f(x')dx' \tag{A3}$$

This $F_X(x)$, or simply $F(x)$, is called the cumulative distribution function (hereinafter c. d. f.) of the random variable X. This function shall be used subsequently.

A concept basic to the discussion of random variables is the expectation value $E[\ ]$ of a function of a random variable $g(X)$:

$$E[g(X)] \equiv \int_{-\infty}^{+\infty} g(x)f(x)dx \tag{A4}$$

In particular, the expectation of a random variable itself

$$E[X] = \int_{-\infty}^{+\infty} xf(x)dx \tag{A5}$$

is the familiar mean or average value of X, $-\infty < X < \infty$. Of interest in the text is the expectation of the function $1/U$ of the random variable U distributed as $f(u)$ (eq. (A2)).

$$E\left[\frac{1}{U}\right] = \int_0^{+\infty} \frac{1}{u} 2ue^{-u^2} du$$

$$= 2\int_0^{\infty} e^{-u^2} du$$

hence,

$$E\left[\frac{1}{U}\right] = \sqrt{\pi} \tag{A6}$$

Choosing from a Distribution

It is first necessary to define what is meant by choosing a sequence of random numbers X_k from a distribution $f(x)$ (or equivalently, choosing X_k distributed as $f(x)$). It is assumed, for the sake of illustration, that the p. d. f. $f(x)$ is nonzero only in the interval $0 \leq x \leq 1$. This interval is then subdivided into 10 equal subintervals. Then, if the sequence of N random numbers X_k is distributed as $f(x)$, a plot of N_i/N against the midpoint of the i^{th} interval (where N_i is the number of X_k's in the i^{th} interval) should approximate $f(x)$. Of course, the larger the number N, and/or the smaller the subdivision used, the better will be the approximation.

How is a sequence of random numbers, say U_k distributed as f(u) (eq. (A2)), chosen on a digital computer? In practice, this sequence is not obtained; instead, a sequence of random (pseudo-random) numbers R_k is obtained, distributed as the uniform distribution

$$\left.\begin{aligned} p(r) &= 0 \qquad r < 0 \\ &= 1 \qquad 0 \leq r \leq 1 \\ &= 0 \qquad r > 1 \end{aligned}\right\} \tag{A7}$$

Hence the immediate problem then becomes, given a sequence of random numbers R_k distributed as p(r) (eq. (A7)), how to obtain, even indirectly, a sequence of random numbers U_k distributed as f(u) (eq. (A2)).

Consider two random variables X and Y related by the monotonic increasing function

$$Y = h(X) \tag{A8}$$

where X has a known p.d.f. f(x). Then if x and y are corresponding values related by equation (A8),

$$P[Y < y] = P[X < x] \tag{A9}$$

and

$$\left.\begin{aligned} P[Y < y] = G_Y(y) &= \int_{-\infty}^{y} g(y')dy' \\ P[X < x] = F_X(x) &= \int_{-\infty}^{x} f(x')dx' \end{aligned}\right\} \tag{A10}$$

or

$$\int_{-\infty}^{y} g(y')dy' = \int_{-\infty}^{x} f(x')dx' \tag{A11}$$

The inverse problem can now be considered. If g(y) and f(x) are given, the functional relationship between y and x (eq. (A8)), such that equation (A9) is still valid,

must be determined. This relation can be easily obtained providing both integrals of equation (A11) can be solved in closed form. For example, if the p.d.f.'s p(r) (eq. (A7)) and f(u) (eq. (A2)) are employed,

$$\int_0^R p(r)dr = \int_0^U f(u)du \tag{A12}$$

$$\int_0^R 1\, dr = \int_0^U 2ue^{-u^2}\, du \tag{A13}$$

$$R = -e^{-U^2} + 1 \tag{A14}$$

or

$$U^2 = -\ln(1 - R)$$

but since R is a random number between 0 and 1, 1 - R is also a random number between 0 and 1; hence,

$$U^2 = -\ln R \tag{A15}$$

is the required functional relationship between U and R. Therefore, only a sequence of random numbers R_k from the uniform distribution p(r) (eq. (A7)) need be obtained, and then equation (A15) can be used to obtain a sequence of random numbers U_k distributed as f(u) (eq. (A2)).

Generalizing the previous procedure to obtain a sequence of random numbers X_k distributed as f(x), and given a sequence of random numbers R_k from the uniform distribution (eq. (A7)), it is only necessary to solve the equation

$$R_k = F(X_k) \tag{A16}$$

where F(x) is the c.d.f. (eq. (A3)) of X.

This method becomes unwieldy, however, whenever F(x) cannot be expressed in closed form as in the preceding example. There do exist techniques for choosing from distributions in this case (e.g., the rejection method, ref. 14), but they need not be discussed here.

Another important aspect of random sampling from a given distribution is the result of summing the random numbers, or a function of the random numbers, obtained. For

instance, if a sequence of random numbers X_k is chosen and distributed as f(x), there is obtained upon summing

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} X_k = E[X] \equiv \int_{-\infty}^{+\infty} xf(x)\,dx \tag{A17}$$

This is readily extended to

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} g(X_k) = E\left[g(X_k)\right] \equiv \int_{-\infty}^{+\infty} g(x)f(x)dx \tag{A18}$$

For a particular case of interest in the text

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \frac{1}{U_k} = E\left[\frac{1}{U_k}\right] = \sqrt{\pi} \tag{A19}$$

from equation (A6), where U is distributed as f(u) (eq. (A2)). A sample mean is defined as

$$\bar{g}_N = \frac{1}{N} \sum_{k=1}^{N} g(X_k) \tag{A20}$$

for finite N.

Standard Deviation

If the random variable X is distributed as f(x) and g(x) is an integrable function of x, then

$$E[g] \equiv \int_{-\infty}^{+\infty} g(x)f(x)dx$$

$$E[g^2] \equiv \int_{-\infty}^{+\infty} g(x)^2 f(x)dx$$

and the standard deviation of $g(x)$ is defined as

$$\hat{\sigma}_g^2 \equiv E\left[(g - E[g])^2\right] \equiv \int_{-\infty}^{+\infty} \{g(x) - E[x]\}^2 f(x)dx = E[g^2] - E[g]^2 \tag{A21}$$

It is noted that this definition of $\hat{\sigma}_g$ is based on a knowledge of the p. d. f. of X. It can be shown that an unbiased estimate of $\hat{\sigma}_g$ can be obtained (ref. 10, p. 370, exercise 4. 6) from a random sequence $\left\{g(X_k)\right\}$ by the formula

$$\sigma_g^2 = \frac{1}{N - 1} \sum_{k=1}^{N} \left[g(X_k) - \bar{g}\right]^2 \tag{A22}$$

Equation (A22) represents the computation performed in the text to obtain σ_J (see table I, p. 14).

Central Limit Theorem

This theorem (ref. 4, p. 362) is central to all Monte Carlo problems. It is based on the fact that regardless of the distribution of X, the sample means $\bar{g}$ (eq. (A20)) are distributed approximately as a normal distribution.

The central limit theorem can then be stated as

$$\lim_{N \to \infty} P\left[E(g) + \frac{\alpha\sigma_g}{\sqrt{N}} < \bar{g}_N < E(g) + \frac{\beta\sigma_g}{\sqrt{N}}\right] = \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\beta} e^{-t^2/2}\, dt \tag{A23}$$

For $\alpha = -1$ and $\beta = 1$, this theorem shows that the probability of the sample mean lying within $\pm\sigma_g/\sqrt{N}$ of the true value is approximately 0. 95.

APPENDIX B

COMPUTER PROGRAMS

by Harold E. Renkel

Curve Fitting Program

Subroutine CHEBY listing is a program for determining a finite approximation $f_N(x)$ in the least squares sense to data y_α obtained at the arguments x_α where

$$f_N(x) = \sum_{k=0}^{N} a_k x^k \tag{B1}$$

In the present problem the advantage is being able to choose the arguments before taking the data. This permits the application of Chebyshev polynomials as described by Lanczos (ref. 16). This method is both very powerful and very efficient. The coefficients a_k in equation (B1) are obtained without the need of inverting a matrix as is usual in the ordinary method of least squares curve fitting.

The arguments x_α are found from

$$x_\alpha = \frac{1}{2}\left[\cos\theta_\alpha + 1\right] \tag{B2}$$

where

$$\theta_\alpha = \alpha T/N$$

Then an expansion for $f_N(x)$ in terms of the shifted Chebyshev polynomial (ref. 16) $T_k(x)$ is obtained:

$$f_N(x) = \frac{1}{2} b_o + \sum_{k=1}^{N} b_k T_k(x) \tag{B3}$$

The coefficients b_k are obtained from

$$b_k = \frac{1}{2} y_o + \sum_{\alpha=1}^{N-1} y_\alpha T_k(x_\alpha) + \frac{1}{2} y_N T_k(x_N) \tag{B4}$$

where

$$T_k(x_\alpha) \equiv \cos(k\theta_\alpha) = \cos\left(k \frac{\alpha\pi}{N}\right) \tag{B5}$$

and y_α are the data obtained at x_α.

Each Chebyshev polynomial, however, can be expressed as a power series in x with integral coefficients:

$$T_k(x) = \sum_{j=0}^{N} C_{kj} x^j \tag{B6}$$

where the C_{kj} can be obtained from the following recursion relations:

$$\left.\begin{aligned} T_{k+1}(x) &= 2(2x - 1)T_k(x) - T_{k-1}(x) \\ T_o(x) &= 1 \\ T_1(x) &= -1 + 2x \end{aligned}\right\} \tag{B7}$$

Substituting equation (B6) into equation (B3) yields the coefficients a_k (eq. (B1)):

$$a_k = \frac{1}{2} b_o C_{ok} + \sum_{j=1}^{N} b_j C_{jk} \tag{B8}$$

In addition, subroutine CHEBY makes use of the symmetry of the trigonometric functions (eq. (B5)) as discussed in reference 17 to reduce the number of multiplications needed.

Gaussian Quadrature

The use of Monte Carlo techniques and variables based on random numbers for

numerically solving problems often demands that a large sampling of data be analyzed to obtain the necessary accuracy of the solution. Such large samplings may require many minutes and even hours of computing time if time saving methods are not employed. Subroutine QUAD is a Fortran IV program that numerically integrates a function f(x) over the range x_i to x_o. It is based on the method of Gaussian quadrature (ref. 18) which states that

$$\int_{x_i}^{x_o} f(x)dx = \sum_{j=1}^{n} H_j f(a_j) + E_n$$

where the H_j's are a sequence of weight coefficients and the a_j's are the associated abscissas that have been determined as the roots of certain orthogonal polynomials. The well-known error term E_n based on the $2n^{th}$ derivative of f(x) is not considered to be of such magnitude as to affect present calculations and therefore has been omitted from subroutine QUAD. In comparison to other more popular methods of numerical integration such as the trapezoidal formula and Simpson's rule, which require that the integrand be evaluated at many points over the range of integration $[x_i, x_o]$, Gaussian quadrature will produce the same accuracy with comparatively fewer evaluations of the integrand, which results in a considerable savings of computing time especially if the integrand f(x) contains trigonometric functions, logarithms, or square roots.

The subroutine in present form includes the weight coefficients and abscissas for n = 3 through 16. To apply subroutine QUAD, the function f(x) to be integrated, the upper and lower limits of the integral x_i and x_o, and n the number of points of evaluation must all be specified. The program converts the abscissas from the range [-1, 1] to the range $[x_i, x_o]$ by the algorithm

$$x_j(a_j) = X_i + \frac{1}{2}(x_o - x_i)(a_j + 1)$$

evaluates the integrand f(x) and x_j, and calculates the sum of the products $H_j f(a_j)$. The final sum is then multiplied by the correction factor $\frac{1}{2}(x_o - x_i)$ to compensate for the change in the range of the variable of integration.

When analyzing a function to be integrated by this method one must be careful to note any discontinuities in the range of integration $[x_i, x_o]$. If any should exist, then it becomes necessary to divide the region of integration into smaller intervals, choosing the new limits of integration so that comparatively small regions are established in the neighborhood of the discontinuity. This causes the integrand to be evaluated more often

in the neighborhood and results in a more accurate solution. The total integral for the interval $[x_i, x_o]$ is the usual sum of the integrals of each of the subdivisions.

APPENDIX C

FLOW CHARTS AND PROGRAM LISTINGS

The symbols used in the flow charts (figs. 12 to 16) are as follows:

DELX	size of subdivisions
EN	number of subdivisions
FPATH	distance to collision
IC	location number corresponding to XC
IFF	location number of bound to region containing XC
IMIN	location number corresponding to XMIN
IO	location number corresponding to XO
ITP	location number corresponding to XTP
KI	number of iterations
N	number of histories per iteration
PHIMIN	magnitude of potential minimum
S	path length along trajectory from XO to XTP or XC
TPHIX(I)	tabulated values of $\varphi(y)$
USQ	u^2
USQO	u_o^2
VSQ	V^2
XC	location of collision
XMIN	location of potential minimum
XO	location of scatter
XTP	location of turning point

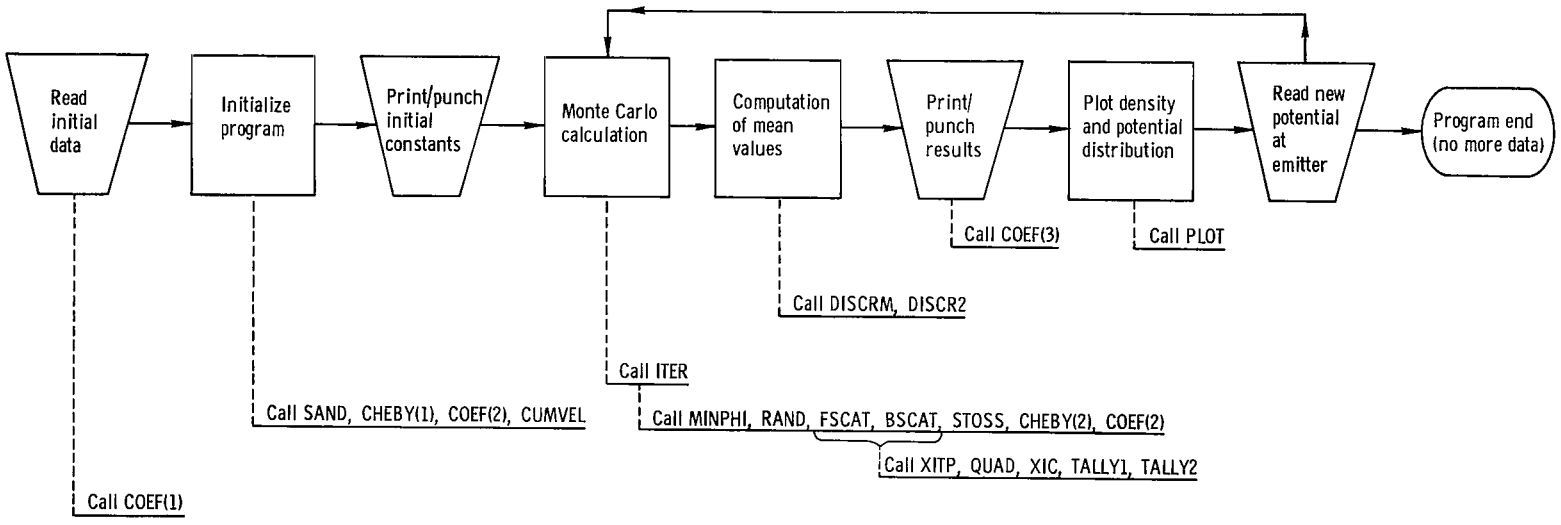


Figure 12. - System and MAIN program flow chart.

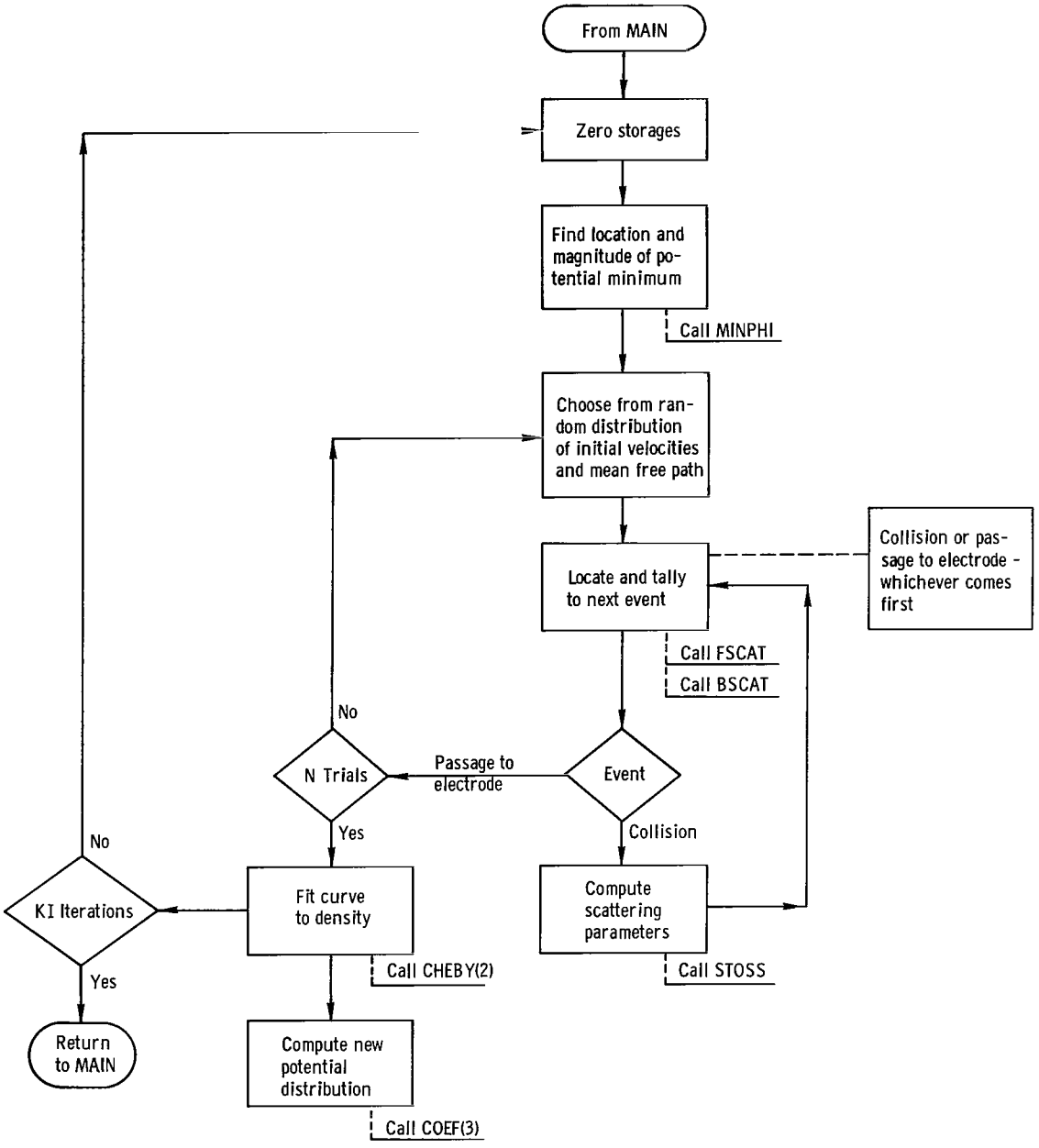


Figure 13. - Subroutine ITER flow chart.

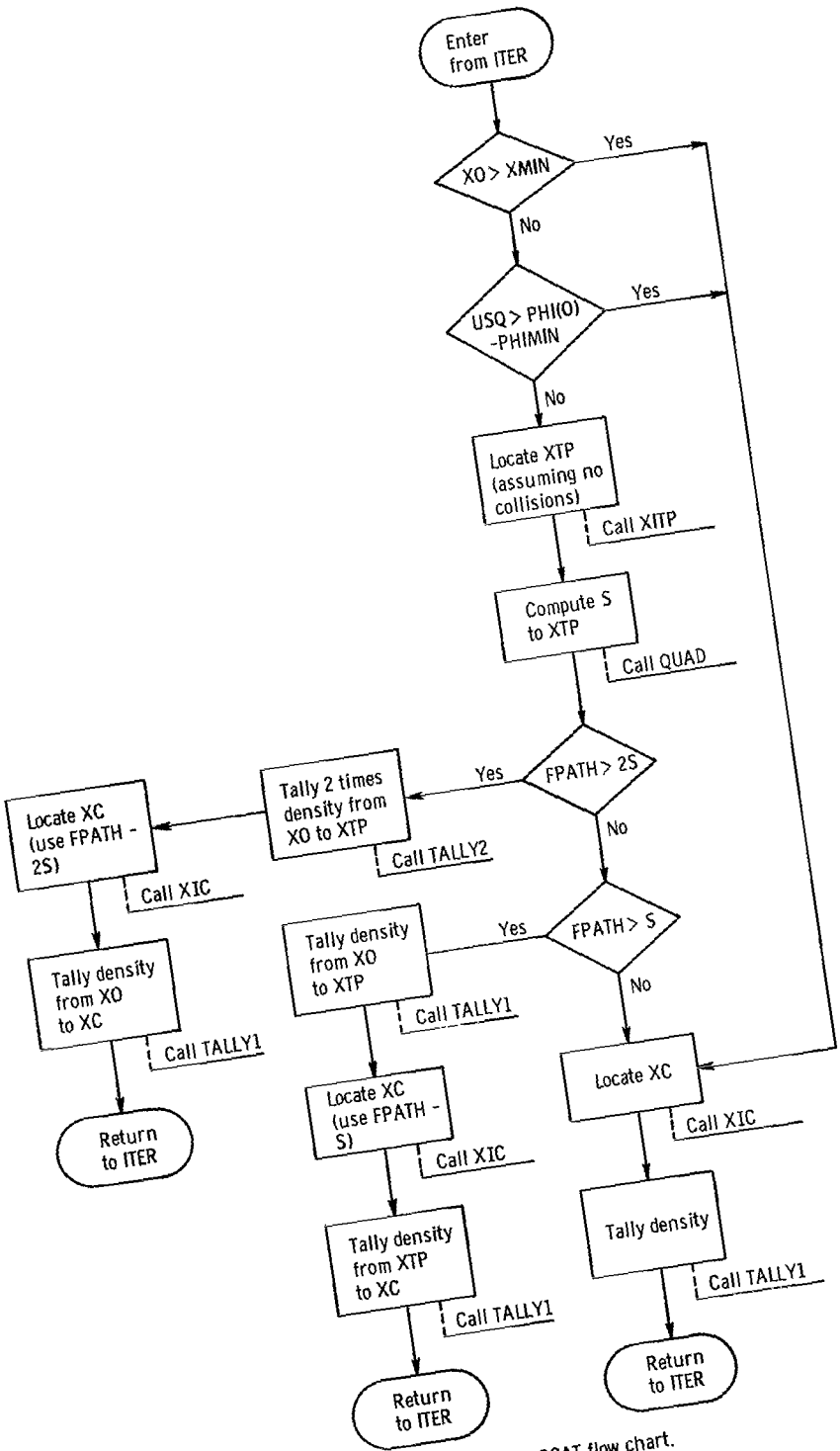


Figure 14. - Subroutine FSCAT flow chart.

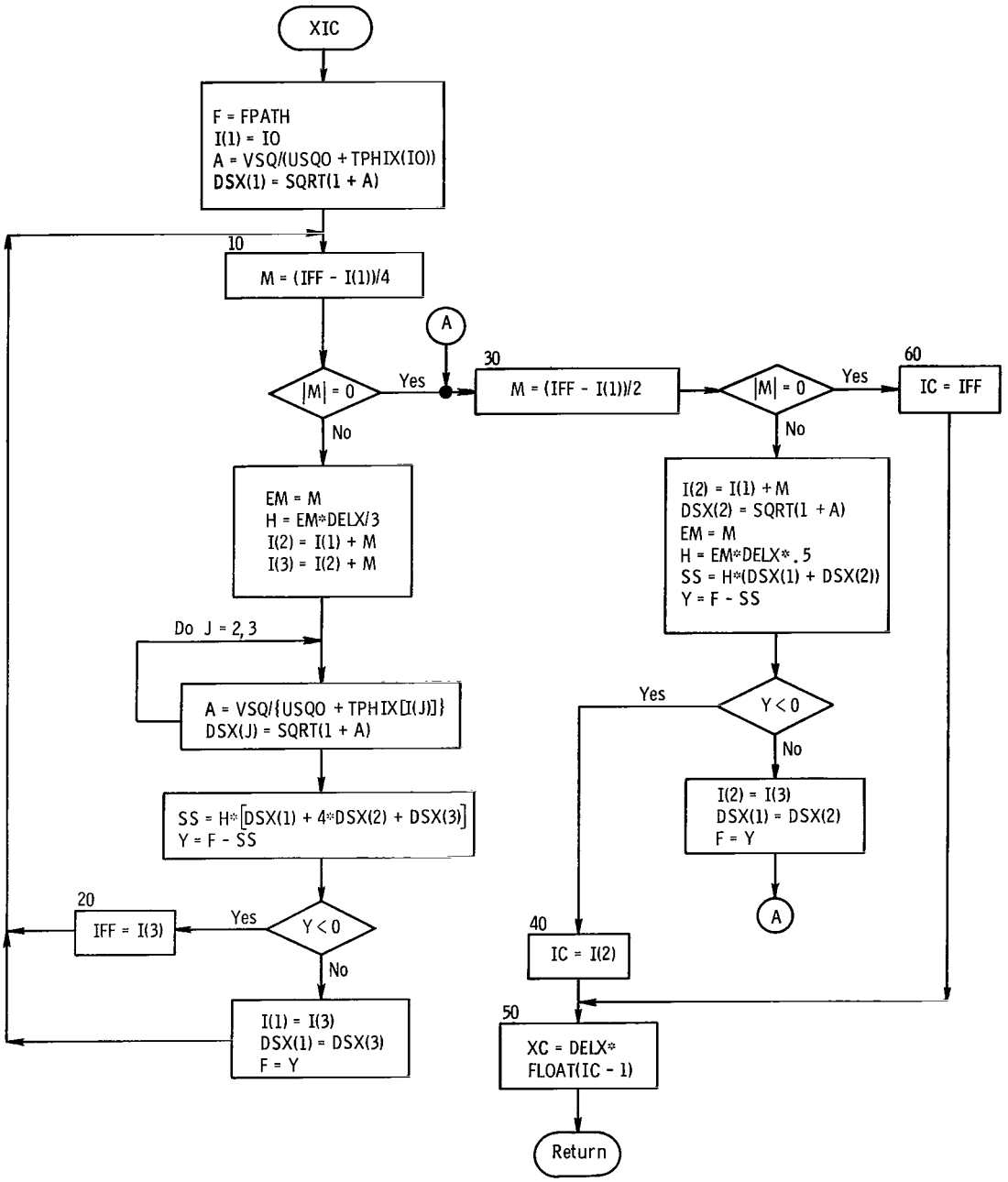


Figure 15. - Subroutine XIC flow chart.

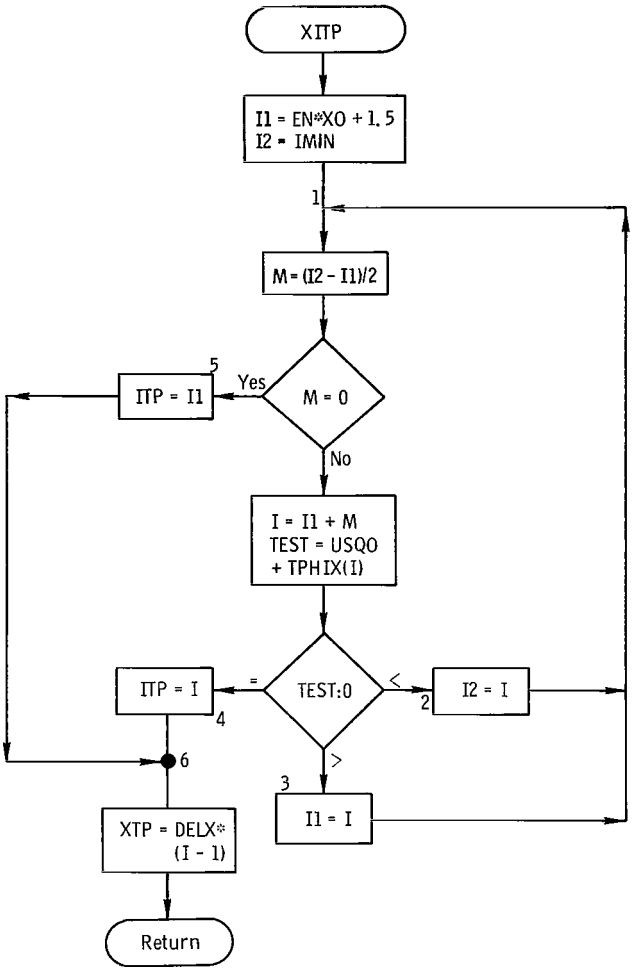


Figure 16. - Subroutine XITP flow chart.

A listing of the FORTRAN IV programs used to calculate the transport properties in a low ionized gas follows.

```
$IBFTC MAIN    DEBUG,DECK                                                    1 00031
       COMMON/BITER/NO,KI                                                    1 00040
       COMMON/BSTOSS/USQ,VSQ,COSN,ALPHA                                      1 00050
       COMMON/BPHI/NPHI,APHI(20),NDPHI,ADPHI(20),NDEN                        1 00060
       COMMON/BMAIN/ CONST,VOLT(20),CURRNT(20)                               1 00070
       COMMON/BMIN/XMIN,PHIMIN,IMIN,TPHID(33),TPHIX(1026),A,B,C,DELX,EN      1   075
       COMMON/BCHEB/N1,X(21),Y(21),COEFS(21),ERROR                           1 00080
       COMMON/BCHEB2/N2                                                      1 00081
       COMMON/BVEL/VEL(1024)                                                 1 00085
       DIMENSION DATE(2)                                                     1   095
       COMMON KNTR,N(20)
C    NO = NUMBER OF TRIALS PER ITERATION                                     1   110
C    N1 = NUMBER OF POINTS WHERE DENSITY IS SAMPLED                          1   115
C    KI = NUMBER OF ITERATIONS                                               1   116
C    ALPHA = RATIO OF MFP TO ELECTRODE SPACING                               1   117
C    CONST = CONSTANT PRECEEDING DENSITY IN POISSONS EQUATION                1   118
C                                                                            1   119
C    READ IN INITIAL DATA                                                    1   120
       READ(5,1) NO,N1,N2,KI,ALPHA,CONST                                     1  0130
     1 FORMAT(4I5,2E10.0)                                                    1   140
       CALL COEF(1)                                                          1   150
       READ(5,4) DATE                                                        1   151
     4 FORMAT(2A6)                                                           1   152
C                                                                            1   155
C    INITIALIZE PROGRAM                                                      1   160
       CALL SAND(RO)                                                         1   170
       CALL CHEBY(1)                                                         1   180
       CALL COEF(2)                                                          1   190
       DEBUG (COEFS(I),I=1,11)                                               1 00192
       DEBUG(APHI(I),I=1,13)                                                 1 00192
       CALL CUMVEL                                                           1 00194
C                                                                            1   205
C    COMPUTE COLLECTOR VOLTAGE AND CURRENT FOR EACH VALUE OF APHI(2)         1   210
     5 READ(5,2) VOLT(20),ALPHA                                              1   220
     2 FORMAT(2E5.1)                                                         1   230
     3 CALL TIME1(T1)                                                        1   233
C                                                                            1   234
C    COMPUTE MEANS AND PRINT / PUNCH OUT RESULTS                             1   235
       WRITE(6,100) VOLT(20),CONST,ALPHA,NO,KI,N1,N2                         1   236
   100 FORMAT(1H1,     39H              ANODE POTENTIAL  IS       ,F6.2/     1   237
      1       1X,      39HCONSTANT IN POISSONS EQUATION,      C =,F6.2/     1   238
      1 1X,39HDIMENSIONLESS MEAN FREE PATH     ALPHA=,1PE10.1/               1 00239
      1       1X,      39HTRIALS PER ITERATION,            NO =,I5  /       1   240
      1 1X,39HNUMBER OF ITERATIONS              KI =, I2 /                   1 00243
      1       1X,      39HNUMBERS OF SAMPLE POINTS,        N1 =,I2  /       1   244
      1       1X,      39HNO. OF TERMS IN DENSITY FIT      N2 =,I2  )       1   245
       CALL ITER                                                             1   250
       CALL DISCRM                                                           1   270
       CALL DISCR2(CURRNT,CM,CSTD)                                           1 00290
       CALL CHEBY(2)                                                         1   292
       CALL COEF(2)                                                          1   293
       CALL COEF(3)                                                          1   295
       CALL TIME1(T2)                                                        1   296
       TIME=(T2-T1)/3600.                                                    1 00297
       WRITE(6,203) KNTR                                                     1   298
   203 FORMAT(1HO,28HTOTAL NUMBER OF COLLISIONS =,I5/                        1   299
      11HO,36HNUMBER OF ENTRIES AT EACH DATA POINT  )                        1   300
       WRITE(6,204) (N(I),I=1,N1)                                            1   301
   204 FORMAT(1H ,11I10)                                                     1   302
       WRITE(6,200) VOLT(20),APHI(2),CM,CSTD,XMIN,PHIMIN,TIME                1   303
   200 FORMAT(1HO,17HANODE POTENTIAL =,F15.6,6X,28HPOTENTIAL SLOPE AT EMI    1   304
      1TTER =,F10.4/1H ,15HANODE CURRENT =,F15.6,6X,9HSTD.DEV.= ,F15.6/      1   305
      11HO,6HXMIN =,F15.6,6X,8HPHIMIN =,F15.6/1HO,6HTIME =,F6.3,             1   306
      18H MINUTES )                                                          1   307
       ELMFP=1./ALPHA                                                        1   308
       WRITE(6,202) DATE,CONST,ELMFP,VOLT(20),CM,CSTD,APHI(2),NO,KI          1   309
   202 FORMAT(1H$,2A6,F7.0,F6.2,F9.3,F9.4,F10.5,F8.2,I7,I4)                  1   310
C                                                                            1   340
C   PLOT DENSITY AND POTENTIAL DISTRIBUTION                                  1   350
       CALL PLOT                                                             1   360
       GO TO 5                                                               1   370
       STOP                                                                  1   380
       END                                                                   1   390
```

```
$IBFTC ITER    DEBUG,DECK                                                    2 00741
       SUBROUTINE ITER                                                      2 00750
C                                                                           2 00755
C MONTE CARLO CALCULATION OF DENSITY  AND ITERATION ON POTENTIAL -           2 00756
C   DISTRIBUTION                                                            2 00757
C                                                                           2 00758
       COMMON/BMAIN/ CONST,VOLT(20),CURRNT(20)                              2 00760
       COMMON/BITER/NO,KI                                                   2 00770
       COMMON/BNXF/XO,XF,XC,XTP,FPATH,S,NQUAD ,K,II                         2 00780
       COMMON/BPHI/NPHI,APHI(20),NDPHI,ADPHI(20),NDDPHI                     2 00790
       COMMON/BCHEB/N1,X(21),Y(21),COEFS(21),ERROR                          2 00800
       COMMON/BMIN/XMIN,PHIMIN,IMIN,TPHID(33),TPHIX(1026),A,B,C,DELX,EN     2 00810
       COMMON/BNIF/IO,IC,ITP,PHIO,USQO                                      2 00815
       COMMON/BSTOSS/USQ,VSQ,COSN,ALPHA                                     2 00820
       COMMON/BVEL/ VEL(1024)                                               2 00825
       COMMON/BTALLY/ ICM1,DEN(20,20)                                       2 00830
       COMMON KNTR,N(11)                                                    2 00834
       DATA SQRTPI/1.77245385/                                              2 00835
       INTEGER A                                                            2  0836
       EQUIVALENCE(A,KNTR2)                                                 2  0837
       DO 35 K=1,KI                                                         2 00840
       CALL TIME1(T1)                                                         2
       NTHRU = 0.                                                           2 00850
       DO 10 I=1,N1                                                         2 00860
       N(I)=0                                                               2 00865
   10  DEN(I,K)= 0.                                                         2 00870
C                                                                           2 00875
C DETERMINATION OF LOCATION AND MAGNITUDE  OF POTENTIAL MINIMUM             2 00876
       CALL MINPHI                                                          2 00880
       DEBUG(TPHID(I),I=1,11)                                               2 00882
       KNTR=0                                                               2 00885
       KNTR2=0                                                              2  0886
       DO 33 II=1,NO                                                        2 00900
       CALL RAND(R)                                                         2 00910
       J = IFIX(1024.*R)+1                                                  2 00920
       USQ = VEL(J)                                                         2 00925
       CALL RAND(R)                                                         2 00930
       J = IFIX(1024.*R) +1                                                 2 00940
       VSQ = VEL(J)                                                         2 00945
       CALL RAND(R)                                                         2 00950
       J = IFIX(1024.*R) +1                                                 2 00960
       FPATH =  ALPHA*VEL(J)                                                2 00965
       XO=0.                                                                2 00980
       ICM1=0                                                               2 00990
       CALL FSCAT                                                           2 01000
   20  IF(ICM1.EQ.N1) GO TO 32                                              2 01010
       IF(ICM1.EQ.0) GO TO 33                                               2 01020
       CALL STOSS                                                           2 01030
       IF(COSN.LT.0.) GO TO 25                                              2 01040
       CALL FSCAT                                                           2 01050
       GO TO 20                                                             2 01070
   25  CALL BSCAT                                                           2 01080
       GO TO 20                                                             2 01100
   32  NTHRU = NTHRU+1                                                      2 01110
   33  CONTINUE                                                             2 01120
       DO 34 I=1,N1                                                         2 01130
       DEN(I,K) = DEN(I,K)/(FLOAT(NO)*SQRTPI)                               2 01140
   34  Y(I)= DEN(I,K)                                                       2 01150
       DEBUG(Y(I),I=1,N1)                                                   2 01152
C                                                                           2 01154
C CURVE FIT OF DENSITY AND COMPUTATION OF  PHI(X) AND DPHI(X)               2 01155
       CALL CHEBY(2)                                                        2 01160
C                                                                           2 01175
       CALL COEF(2)                                                         2 01176
       VOLT(K)=TPHID(N1)                                                    2 01180
       CURRNT(K) = FLOAT(NTHRU)/FLOAT(NO)                                   2 01190
       DEBUG(N(I),I=1,11)                                                   2 01192
       DEBUG KNTR                                                           2 01192
       DEBUG VOLT(K), CURRNT(K)                                             2 01194
       CALL TIME1(T2)                                                         2
       TIME=(T2-T1)/3600.                                                     2
       DEBUG TIME                                                             2
   35  CONTINUE                                                             2 01195
       RETURN                                                               2 01200
       END                                                                  2 01210
```

```
$IBFTC FSCAT   DECK
      SUBROUTINE FSCAT                                                  3 01230
      COMMON/BMIN/XMIN,PHIMIN,IMIN,TPHID(33),TPHIX(1026),A,B,C,DELX,EN  3  1240
      COMMON/BNXF/XO,XF,XC,XTP,FPATH,S,NQUAD ,K,II                      3 01250
      COMMON/BSTOSS/USQ,VSQ,COSN,ALPHA                                  3 01260
      COMMON/BNIF/IO,IC,ITP,PHIO,USQO                                   3 01270
      IO=EN*XO+1.5                                                      3  1270
      PHIO=TPHIX(IO)                                                    3 01275
      USQO=USQ-PHIO                                                     3 01276
      IF(XO.GE.XMIN) GO TO 6                                            3 01280
      IF(USQO.GT.-PHIMIN) GO TO 4                                       3 01290
      CALL XITP                                                         3 01300
      IF(IO.EQ.ITP) GO TO 10                                            3  1305
      XF=XTP                                                            3 01310
      CALL QUADGM                                                       3 01340
      DIF (K.EQ.1.AND.II.LE.5) DEBUG XTP,S                              3 01352
      IF(FPATH.GE. 2.*S) GO TO 3                                        3 01360
      IF(FPATH.GE.S) GO TO 2                                            3 01370
    1 CALL XIC                                                          3 01380
      DIF (K.EQ.1.AND.II.LE.5) DEBUG XC                                 3 01392
      CALL TALLY1                                                       3 01400
      XO=XC                                                             3 01405
      RETURN                                                            3 01410
    2 XC=XTP                                                            3 01420
      CALL TALLY1                                                       3 01430
      FPATH=2.*S-FPATH                                                  3  1460
      GO TO 1                                                           3  1470
    3 CALL TALLY2                                                       3 01510
      FPATH = FPATH-2.*S                                                3  1514
   10 IF(XO.EQ.0.) RETURN                                               3 01515
      XF=0.                                                             3 01520
      NQUAD=5                                                           3 01525
      CALL QUAD                                                         3 01540
      DIF (K.EQ.1.AND.II.LE.5) DEBUG FPATH,S                            3 01552
      IF(FPATH.LT.S) GO TO 1                                            3  1560
      XC=XF                                                             3 01570
      CALL TALLY1                                                       3 01580
      RETURN                                                            3 01590
    6 SMAXSQ=(1.+VSQ/USQ)*(1.-XO)**2                                    3 01600
      IF(FPATH*FPATH.GE.SMAXSQ) GO TO 5                                 3 01603
    4 NQUAD=3                                                           3 01610
      IF(XO.LT..2)NQUAD=5                                               3 01620
      IF(XO.LT..2.AND.USQ.LT..01) NQUAD =9                              3 01630
      XF=1.                                                             3 01640
      CALL QUAD                                                         3 01650
      DIF(K.EQ.1.AND.II.LE.5) DEBUG NQUAD,S                             3 01652
      IF(FPATH.GE.S) GO TO 5                                            3 01660
    9 CALL XIC                                                          3  1675
      DIF (K.EQ.1.AND.II.LE.5) DEBUG S,XC                               3 01682
      CALL TALLY1                                                       3 01690
      XO=XC                                                             3 01695
      RETURN                                                            3 01700
    5 XC=1.                                                             3 01710
      CALL TALLY1                                                       3 01720
      RETURN                                                            3 01730
      END                                                               3 01740
```

```
$IBFTC BSCAT   DECK
      SUBROUTINE BSCAT                                                  4 01760
      COMMON/BMIN/XMIN,PHIMIN,IMIN,TPHID(33),TPHIX(1026),A,B,C,DELX,EN  4 01770
      COMMON/BNIF/IO,IC,ITP,PHIO,USQO                                   4 01775
      COMMON/BNXF/XO,XF,XC,XTP,FPATH,S,NQUAD ,K,II                      4 01780
      COMMON/BSTOSS/USQ,VSQ,COSN,ALPHA                                  4 01790
      IO=EN*XO+1.5                                                      4  1800
      PHIO=TPHIX(IO)                                                    4 01805
      USQO=USQ-PHIO                                                     4 01806
      IF(XO.LT.XMIN)GO TO 4                                             4 01810
      IF(USQO.GT.-PHIMIN) GO TO 4                                       4 01820
      CALL XITP                                                         4 01830
      IF(IO.EQ.ITP) GO TO 10                                            4  1835
      XF=XTP                                                            4  1840
      CALL QUADGM                                                       4 0186
      DIF (K.EQ.1.AND.II.LE.5) DEBUG XTP,S                              4 01872
      IF(FPATH.GE. 2.*S) GO TO 3                                        4 01880
      IF(FPATH.GE.S) GO TO 2                                            4 01890
    1 CALL XIC                                                          4 01900
      DIF (K.EQ.1.AND.II.LE.5) DEBUG XC                                 4 01912
      CALL TALLY1                                                       4 01920
      XO=XC                                                             4  1925
      RETURN                                                            4 01930
    2 XC=XTP                                                            4 01940
      CALL TALLY1                                                       4 01950
      FPATH=2.*S-FPATH                                                  4  1980
      GO TO 1                                                           4  1990
    3 CALL TALLY2                                                       4 02030
      FPATH = FPATH-2.*S                                                4 02050
   10 SMAXSQ=(1.+VSQ/USQ)*(1.-XO)**2                                    4 02060
      IF(FPATH*FPATH.GE.SMAXSQ) GO TO 6                                 4 02065
      XF=1.                                                             4  2066
      NQUAD=5                                                           4 02068
      CALL QUAD                                                         4 02070
      DIF (K.EQ.1.AND.II.LE.5) DEBUG FPATH,S                            4 02082
      IF(FPATH.LT.S) GO TO 1                                            4 02090
    6 XC=1                                                              4  2100
      CALL TALLY1                                                       4 02110
      RETURN                                                            4 02120
    4 NQUAD=5                                                           4 02130
      IF(USQ.LT..01) NQUAD=9                                            4 02140
      XF=0.                                                             4 02150
      CALL QUAD                                                         4 02160
      IF(FPATH.GE.S) GO TO 5                                            4 02170
    9 CALL XIC                                                          4  2180
      DIF (K.EQ.1.AND.II.LE.5) DEBUG S,XC                               4 02192
      CALL TALLY1                                                       4 02200
      XO=XC                                                             4 02205
      RETURN                                                            4 02210
    5 XC=0.                                                             4 02220
      CALL TALLY1                                                       4 02230
      RETURN                                                            4  2320
      END                                                               4  2330
```

```
$IBFTC STOSS   DECK
      SUBROUTINE  STOSS                                                 5 02270
C                                                                       5 02274
C COMPUTATION OF COLLISION PARAMETERS  FOR ELECTRON-NEUTRAL  SCATTERING  5 02275
C                                                                       5 02276
      COMMON/BSTOSS/USQ,VSQ,COSN,ALPHA                                  5 02280
      COMMON/BNXF/XO,XF,XC,XTP,FPATH,S,NQUAD ,K,II                      5 02290
      COMMON/BVEL/VEL(1024)                                             5 02295
      COMMON/BNIF/IO,IC,ITP,PHIO,USQO                                   5  2294
      COMMON/BMIN/XMIN,PHIMIN,IMIN,TPHID(33),TPHIX(1026),A,B,C,DELX,EN  5 02296
      WSQ=USQO+VSQ+TPHIX(IC)                                            5 02300
      CALL RAND(R)                                                      5 02310
      COSN=1.-2.*R                                                      5 02320
      USQ=WSQ*COSN*COSN                                                 5 02330
      VSQ=WSQ-USQ                                                       5 02340
      CALL RAND(R)                                                      5 02350
      J=IFIX(1024.*R)+1                                                 5 02360
      FPATH=ALPHA*VEL(J)                                                5 02365
      DIF (K.EQ.1.AND.II.LE.5) DEBUG USQ,VSQ,COSN,FPATH                 5 02372
      RETURN                                                            5 02380
      END                                                               5 02390
```

```
$IBFTC TALLY1  DECK                                                          6 02401
      SUBROUTINE TALLY1                                                      6 02410
      COMMON/BSTOSS/USQ,VSQ,COSN,ALPHA                                       6 02420
      COMMON/BNXF/XO,XF,XC,XTP,FPATH,S,NQUAD ,K,II                           6 02430
      COMMON/BCHEB/N1,X(21),Y(21),COEFS(21),ERROR                            6 02440
      COMMON/BTALLY/ ICM1,DEN(20,20)                                         6 02450
      COMMON/BMIN/XMIN,PHIMIN,IMIN,TPHID(33),TPHIX(1026),A,B,C,DELX,EN       6 02460
      COMMON/BNIF/IO,IC,ITP,PHIO,USQO                                        6 02465
      COMMON KNTR,N(11)                                                      6 02470
      IF(XO.GT.XC) GO TO 7                                                   6 02480
      I = ICM1                                                               6 02490
      GO TO 10                                                               6 02500
    1 DEN(I,K) = DEN(I,K) + 1./SQRT(TEST)                                    6 02510
      DIF (K.EQ.1.AND.II.LE.5) DEBUG DEN(I,K)                                6 02512
      N(I)=N(I)+1                                                            6 02515
   10 I = I+1                                                                6 02520
      TEST=USQO+TPHID(I)                                                     6 02525
      IF(TEST.LE.0.) GO TO 3                                                 6 02526
       IF(X(I).EQ.XC) GO TO 2                                                6 02530
      IF(X(I).LT.XC) GO TO 1                                                 6 02535
      GO TO 3                                                                6 02536
    2 DEN(I,K) = DEN(I,K) + 1./SQRT(TEST)                                    6 02540
      N(I)=N(I)+1                                                            6 02545
      DIF (K.EQ.1.AND.II.LE.5) DEBUG DEN(I,K)                                6 02552
      ICM1=I                                                                 6 02560
      RETURN                                                                 6 02570
    7 I = ICM1+1                                                             6 02580
      GO TO 11                                                               6 02585
    4 DEN(I,K) = DEN(I,K) + 1./SQRT(TEST)                                    6 02590
      N(I)=N(I)+1                                                            6 02595
      DIF (K.EQ.1.AND.II.LE.5) DEBUG DEN(I,K)                                6 02602
   11 I = I-1                                                                6 02610
      TEST=USQO+TPHID(I)                                                     6 02615
      IF(TEST.LE.0.) GO TO 6                                                 6 02616
       IF(X(I).EQ.XC) GO TO 5                                                6 02620
      IF(X(I).GT.XC) GO TO 4                                                 6 02625
      GO TO 6                                                                6 02626
    5 DEN(I,K) = DEN(I,K) + 1./SQRT(TEST)                                    6 02630
      N(I)=N(I)+1                                                            6 02635
      DIF (K.EQ.1.AND.II.LE.5) DEBUG DEN(I,K)                                6 02642
      ICM1=I-1                                                               6 02650
      RETURN                                                                 6 02660
    3 ICM1=I-1                                                               6 02670
      RETURN                                                                 6 02680
    6 ICM1=I                                                                 6 02690
      RETURN                                                                 6 02700
      END                                                                    6 02710
```

```
$IBFTC TALLY2  DECK                                                          7 02721
      SUBROUTINE TALLY2                                                      7 02730
      COMMON/BTALLY/ ICM1,DEN(20,20)                                         7 02740
      COMMON/BSTOSS/USQ,VSQ,COSN,ALPHA                                       7 02750
      COMMON/BNXF/XO,XF,XC,XTP,FPATH,S,NQUAD ,K,II                           7 02760
      COMMON/BCHEB/N1,X(21),Y(21),COEFS(21),ERROR                            7 02770
      COMMON/BMIN/XMIN,PHIMIN,IMIN,TPHID(33),TPHIX(1026),A,B,C,DELX,EN       7 02780
      COMMON/BNIF/IO,IC,ITP,PHIO,USQO                                        7 02785
      COMMON KNTR,N(11)                                                      7 02786
      IF(XO.GT.XTP) GO TO 2                                                  7 02800
      I=ICM1                                                                 7 02810
      GO TO 3                                                                7 02815
    1 DEN(I,K)=DEN(I,K)+2./SQRT(TEST)                                        7 02820
      N(I)=N(I)+1                                                            7 02825
      DIF (K.EQ.1.AND.II.LE.5) DEBUG DEN(I,K)                                7 02832
    3 I=I+1                                                                  7 02840
      TEST=USQO+TPHID(I)                                                     7 02845
      IF(TEST.GT.0..AND.X(I).LT.XTP ) GO TO 1                                7 02850
      RETURN                                                                 7 02860
    2 I=ICM1+1                                                               7 02870
      GO TO 4                                                                7 02875
    5 DEN(I,K)=DEN(I,K)+2./SQRT(TEST)                                        7 02880
      N(I)=N(I)+1                                                            7 02885
    4 I=I-1                                                                  7 02890
      TEST=USQO+TPHID(I)                                                     7 02900
      IF(TEST.GT.0..AND.X(I).GT.XTP ) GO TO 5                                7 02910
      DIF (K.EQ.1.AND.II.LE.5) DEBUG DEN(I,K)                                7 02912
      RETURN                                                                 7 02920
      END                                                                    7 02930
```

```
$IBFTC DISCRM  DECK                                                          8029051
      SUBROUTINE DISCRM                                                      8029060
      REAL MEAN1, MEAN2                                                      8029070
      COMMON/BITER/NO,K                                                      8 29080
      COMMON/BTALLY/ ICM1,DEN(20,20)                                         8029090
      COMMON/BCHEB/N1,X(21),Y(21),COEFS(21),ERROR                            8029100
      WRITE(6,100)                                                           8029110
  100 FORMAT(1HL, 47H MEAN DENSITIES BEFORE AND AFTER DISCRIMINATION /1H     8029120
     1K, 58H  I      XD       MEAN 1  STD.DEV.1   L   MEAN 2  STD.DEV.2 /)  8029130
      FKR = 1./FLOAT(K)                                                      8029140
      DO 9 I=1,N1                                                            8029150
      SUM1 = 0.                                                              8029160
      SUM2 = 0.                                                              8029170
      DO 13 J=1,K                                                            8029180
      SUM1 = SUM1 + DEN(I,J)                                                 8029190
   13 SUM2 = SUM2 + DEN(I,J)*DEN(I,J)                                        8029200
      MEAN1 =SUM1* FKR                                                       8029210
      STD1=SQRT((SUM2*FKR - MEAN1*MEAN1)/FLOAT(K-1))                         8 29220
      SUM1=0                                                                 8029230
      SUM2=0                                                                 8029240
      L=0                                                                    8029250
      DO 15 J=1,K                                                            8029260
      Q= DEN(I,J)-MEAN1                                                      8029270
      IF(ABS(Q).GT.3.*STD1) GO TO 15                                         8 29280
      SUM1 = SUM1 + DEN(I,J)                                                 8029290
      SUM2 = SUM2 + DEN(I,J)*DEN(I,J)                                        8029300
      L=L+1                                                                  8029310
   15 CONTINUE                                                               8029320
      FLR =1./FLOAT(L)                                                       8029330
      MEAN2 = SUM1*FLR                                                       8029340
      STD2=SQRT((SUM2*FLR - MEAN2*MEAN2)/FLOAT(L-1))                         8 29350
      WRITE(6,120)I,X(I),MEAN1,STD1,L,MEAN2,STD2                             8029360
  120 FORMAT(1H ,1X,I2,1X, 3F10.6,1X,I2,1X,2F10.6)                           8029370
      Y(I)=MEAN2                                                             8029380
    9 CONTINUE                                                               8029390
      RETURN                                                                 8029400
      END                                                                    8029410
```

```
$IBFTC DISCR2  DECK                                                          9029421
      SUBROUTINE DISCR2  (A,AMEAN,ADEV)                                      9029430
      COMMON/BITER/NO,K                                                      9029440
      DIMENSION A(20)                                                        9029450
      REAL MEAN1, MEAN2                                                      9029460
      FKR = 1./FLOAT(K)                                                      9029470
      SUM1 = 0.                                                              9029480
      SUM2 = 0.                                                              9029490
      DO 13 J=1,K                                                            9029510
      SUM1 = SUM1 + A(J)                                                     9029520
   13 SUM2 = SUM2 + A(J) *A(J)                                               9029530
      MEAN1 =SUM1* FKR                                                       9029540
      STD1 = SQRT((SUM2*FKR - MEAN1*MEAN1)/FLOAT(K-1) )                      9 29550
      SUM1=0                                                                 9029560
      SUM2=0                                                                 9029570
      L=0                                                                    9029580
      DO 15 J=1,K                                                            9029590
      X=A(J)-MEAN1                                                           9029600
      IF(ABS(X).GT.3.*STD1) GO TO 15                                         9 29610
      SUM1=SUM1+A(J)                                                         9029615
      SUM2 = SUM2 + A(J) *A(J)                                               9029620
      L=L+1                                                                  9029630
   15 CONTINUE                                                               9029640
      FLR =1./FLOAT(L)                                                       9029650
      MEAN2 = SUM1*FLR                                                       9029660
      STD2 = SQRT((SUM2*FLR - MEAN2*MEAN2)/FLOAT(L-1))                       9 29670
      AMEAN=MEAN2                                                            9029680
      ADEV=STD2                                                              9029690
      RETURN                                                                 9029700
      END                                                                    9029710
```

```
$IBFTC QUAD    DECK,DEBUG                                                                10029721
       SUBROUTINE QUAD                                                                   10029730
       COMMON/BNXF/XI,XO,XC,XTP,FPATH,S,NQUAD ,K,II                                      10029750
       COMMON/BSTOSS/WSQ,VSQ,COSN,ALPHA                                                  10029755
       COMMON/BNIF/IO,IC,ITP,PHID,USQD                                                   10029756
       DIMENSION A(70), H(70)                                                            10029790
       REAL INTGRL                                                                       10029800
       DATA (A(I),H(I),I=1,28)/                                                          10030050
      1 7.74596669241483E-01, 5.55555555 55556E-01,-0.              E-00,10030060
      1 8.88888888888889E-01, 8.61136311 94053E-01, 3.47854845137454E-01,10030070
      1 3.99981043584856E-01, 6.52145154 62546E-01, 9.06179845938664E-01,10030080
      1 2.36926885056189E-01, 5.38469310 05683E-01, 4.78628670499366E-01,10030090
      1-0.              E-00, 5.68888888 88889E-01, 9.32469514203152E-01,10030100
      1 1.71324492379170E-01, 6.61209386 66265E-01, 3.60761573048139E-01,10030110
      1 2.38619186083197E-01, 4.67913934 72691E-01, 9.49107912342759E-01,10030120
      1 1.29484966168870E-01, 7.41531185 99394E-01, 2.79705391489277E-01,10030130
      1 4.05845151377397E-01, 3.81830050 05119E-01,-0.              E-00,10030140
      1 4.17959183673469E-01, 9.60289856 97536E-01, 1.01228536290376E-01,10030150
      1 7.96666477413627E-01, 2.22381034 53374E-01, 5.25532409916329E-01,10030160
      1 3.13706645877887E-01, 1.83434642 95650E-01, 3.62683783378362E-01,10030170
      1 9.68160239507626E-01, 8.12743883 15740E-02, 8.36031107326636E-01,10030180
      1 1.80648160694857E-01, 6.13371432 00590E-01, 2.60610696402935E-01,10030190
      1 3.24253423403809E-01, 3.12347077 40003E-01,-0.              E-00,10030200
      1 3.30239355001260E-01, 9.73906528 17172E-01, 6.66713443086880E-02,10030210
      1 8.65063366688985E-01, 1.49451349 50581E-01, 6.79409568299024E-01,10030220
      1 2.19086362515982E-01, 4.33395394 29247E-01, 2.69266719309996E-01,10030230
      1 1.48874338981631E-01, 2.95524224 14753E-01/                                     10030240
       DATA (A(I),H(I),I=29,56)/                                                        10030250
      1 9.78228658146057E-01, 5.56685671 61740E-02, 8.87062599768095E-01,10030260
      1 1.25580369464905E-01, 7.30152005 74049E-01, 1.86290210927734E-01,10030270
      1 5.19096129206812E-01, 2.33193764 91990E-01, 2.69543155952345E-01,10030280
      1 2.62804544510247E-01,-0.              E-00, 2.72925086777901E-01,10030290
      1 9.81560634246719E-01, 4.71753363 65120E-02, 9.04117256370475E-01,10030300
      1 1.06939325995318E-01, 7.69902674 94305E-01, 1.60078328543346E-01,10030310
      1 5.87317954286617E-01, 2.03167426 23066E-01, 3.67831498998180E-01,10030320
      1 2.33492536538355E-01, 1.25233408 11469E-01, 2.49147045813403E-01,10030330
      1 9.84183054718588E-01, 4.04840047 53160E-02, 9.17598399222978E-01,10030340
      1 9.21214998377280E-02, 8.01578090 33310E-01, 1.38873510219787E-01,10030350
      1 6.42349339440340E-01, 1.78145980 61946E-01, 4.48492751036447E-01,10030360
      1 2.07816047536889E-01, 2.30458315 55135E-01, 2.26283180262897E-01,10030370
      1-0.              E-00, 2.32551553 30874E-01, 9.86283808696812E-01,10030380
      1 3.51194603317520E-02, 9.28434883 63574E-01, 8.01580871597600E-02,10030390
      1 8.27201315069765E-01, 1.21518570 87903E-01, 6.87292904811685E-01,10030400
      1 1.57203167158194E-01, 5.15248636 58154E-01, 1.85538397477938E-01,10030410
      1 3.19112368927890E-01, 2.05198463 21296E-01, 1.08054948707344E-01,10030420
      1 2.15263853463158E-01, 9.87992518 20485E-01, 3.07532419961170E-02,10030430
      1 9.37273892400706E-01, 7.03660474 81080E-02/                                     10030440
       DATA (A(I),H(I),I=57,70)/                                                        10030450
      1 8.48206583410427E-01, 1.07159220 67172E-01, 7.24417731360170E-01,10030460
      1 1.39570677926154E-01, 5.70972172 08539E-01, 1.66269205816994E-01,10030470
      1 3.94151347077563E-01, 1.86161000 15562E-01, 2.01194093997435E-01,10030480
      1 1.98431485327112E-01,-0.              E-00, 2.02578241925561E-01,10030490
      1 9.89400934991650E-01, 2.71524594 17540E-02, 9.44575023073233E-01,10030500
      1 6.22535239386480E-02, 8.65631202 87832E-01, 9.51585116824930E-02,10030510
      1 7.55404408355003E-01, 1.24628971 55534E-01, 6.17876244402644E-01,10030520
      1 1.49595988816577E-01, 4.58016777 57227E-01, 1.69156519395003E-01,10030530
      1 2.81603550779259E-01, 1.82603415 44924E-01, 9.50125098376370E-02,10030540
      1 1.89450610455068E-01/                                                           10030550
       EQUIVALENCE (N,NQUAD)                                                            10029810
       XOFA(A)=XI+(XO-XI)*(1.+A)*.5                                                     10029820
  200  INTGRL=0.0                                                                       10029830
       INDKT= MOD(N,2)+1                                                                10029840
C      INDKT = 1, N IS EVEN                                                             10029850
C      INDKT = 2, N IS ODD                                                              10029860
       GO TO (204,210),INDKT                                                            10029870
  204  MIN=(N*N)/4 -1                                                                   10029880
       MAX=(N*(N+2))/4 -2                                                               10029890
       GO TO 215                                                                        10029900
  210  MIN= (N*N-9)/4 +1                                                                10029910
       MAX= (N*(N+2)-11)/4                                                              10029920
  215  DO 220 I=1,2                                                                     10029930
       DO 220 J=MIN,MAX                                                                 10029940
       A(J)= -A(J)                                                                      10029950
       X=XOFA(A(J))                                                                     10029960
       TEST=USQD+PHI(X)                                                                 10029970
       IF(TEST.LE.0.) GO TO 1                                                           10029975
       F=SQRT(1.+VSQ/TEST)                                                              10029976
  220  INTGRL= INTGRL+H(J)*F                                                            10029980
       GO TO (250,225),INDKT                                                            10029990
  225  X=XOFA(A(MAX+1))                                                                 10030000
       TEST=USQD+PHI(X)                                                                 10030010
       IF(TEST.LE.0.) GO TO 1                                                           10030015
       F=SQRT(1.+VSQ/TEST)                                                              10030016
       INTGRL= INTGRL+H(MAX+1)*F                                                        10030020
  250  INTGRL=.5*(XO-XI)*INTGRL                                                         10030030
       S=ABS(INTGRL)                                                                    10030035
       RETURN                                                                           10030040
    1  CONTINUE                                                                         10
       S=0                                                                              10030056
       RETURN                                                                           10030057
       END                                                                              10030560
```

```
$IBFTC PHI      DECK                                                    11030571
       FUNCTION PHI(Z)                                                  11030580
       COMMON/BPHI/NDEG,    B(20),NDPHI,ADPHI(20),NDDPHI                11030590
       LI=NDEG+1                                                        11030600
       P=B(LI)*Z+B(NDEG)                                                11030610
       DO 100 I=2, NDEG                                                 11030620
       LJ=LI-I                                                          11030630
   100 P=B(LJ)+Z*P                                                      11030640
       PHI=P                                                            11030650
       RETURN                                                           11030660
       END                                                              11030670
```

```
$IBFTC DPHI     DECK                                                    12030681
       FUNCTION DPHI(Z)                                                 12030690
       COMMON/BPHI/NPHI,APHI(20),NDEG,      B(20),NDDPHI                12030700
       LI=NDEG+1                                                        12030710
       P=B(LI)*Z+B(NDEG)                                                12030720
       DO 100 I=2, NDEG                                                 12030730
       LJ=LI-I                                                          12030740
   100 P=B(LJ)+Z*P                                                      12030750
       DPHI=P                                                           12030760
       RETURN                                                           12030770
       END                                                              12030780
```

```
$IBFTC DENS     DECK                                                    13 30791
       FUNCTION DENS(Z)                                                 13 30800
       COMMON/BCHEB/X(43),B(21),ERROR                                   13030810
       COMMON/BCHEB2/LI                                                 13030815
       NDEG=LI-1                                                        13030820
       P=B(LI)                                                          13 30830
       DO 100 I=1,NDEG                                                  13 30840
       LJ=LI-I                                                          13030870
   100 P=B(LJ)+Z*P                                                      13030880
       DENS=P                                                           13 30890
       RETURN                                                           13030900
       END                                                              13030910
```

```
$IBFTC MINPHI  DEBUG,DECK                                              14 00010
       SUBROUTINE MINPHI                                                14 00020
       COMMON/BMIN/XMIN,PHIMIN,IMIN,TPHID(33),TPHIX(1026),A,B,C,DELX,EN 14 00030
       COMMON/BCHEB/N1,X(21),DUMMY(43)                                  14 00040
       DATA N/1024/                                                     14 00050
       PHIMIN = 0                                                       14 00060
       IMIN=1                                                           14  0065
       EN=N                                                             14 00070
       DELX = 1./FLOAT(N)                                               14 00080
       DO 1  I=1,N1                                                     14 00090
     1 TPHID(I) = PHI(X(I))                                             14 00100
       M = N+1                                                          14 00110
       TPHIX(1) = 0                                                     14 00120
       DO 2  I=2,M                                                      14 00130
       U=DELX*FLOAT(I-1)                                                14  0140
       TPHIX(I) = PHI(U)                                                14 00150
       IF (PHIMIN.LT.TPHIX(I)) GO TO 2                                  14 00160
       PHIMIN = TPHIX(I)                                                14 00170
       IMIN = I                                                         14 00180
     2 CONTINUE                                                         14 00190
       XMIN=DELX*FLOAT(IMIN-1)                                          14  0200
       DEBUG XMIN,PHIMIN                                                14 00270
       RETURN                                                           14 00320
       END                                                              14 00350
```

```
$IBFTC XIC      DECK                                                          15  0000
      SUBROUTINE XIC                                                        15  0010
      COMMON/BSTOSS/USQ,VSQ,COSN,ALPHA                                      15  0020
      COMMON/BNIF/IO,IC,ITP,PHIO,USQO                                       15  0030
      COMMON/BNXF/XO,XF,XC,XTP,FPATH,S,NQUAD ,K,II                          15  0040
      COMMON/BMIN/XMIN,PHIMIN,IMIN,TPHID(33),TPHIX(1026),A,B,C,DELX,EN      15  0050
      DIMENSION I(3),DSX(3)                                                 15  0055
      COMMON KNTR,N(11)                                                     15  0060
      KNTR=KNTR+1                                                           15  0100
      I(1)=IO                                                               15  0110
      IFF=EN*XF+1.5                                                         15  0120
      F=FPATH                                                               15  0130
      Q=VSQ/USQ                                                             15  0144
      DSX(1)=SQRT(1.+Q)                                                     15  0146
   10 M=(IFF-I(1))/4                                                        15  0150
      IF(IABS(M).EQ.0) GO TO 30                                             15  0160
      MM=ALOG10(Q)                                                          15  0162
      MN=2**(4-MM)                                                          15  0164
      IF(M.GT.MN) M=MN                                                      15  0166
      IF(MM.GT.4) M=1                                                       15  0167
      EM=M                                                                  15  0170
      H=ABS(EM*DELX/3.)                                                     15  0180
      DO 11 J=2,3                                                           15  0190
      I(J)=I(J-1)+M                                                         15  0200
      IJ=I(J)                                                               15  0205
      Q=USQO+TPHIX(IJ)                                                      15  0206
      Q=VSQ/Q                                                               15  0208
      IF(Q.LT.0.) GO TO 61                                                  15  0209
   11 DSX(J)=SQRT(1.+Q)                                                     15  0210
      SS=H*(DSX(1)+4.*DSX(2)+DSX(3))                                        15  0220
      Y=F-SS                                                                15  0230
      IF(Y.LT.0.) GO TO 20                                                  15  0240
      I(1)=I(3)                                                             15  0250
      DSX(1)=DSX(3)                                                         15  0260
      F=Y                                                                   15  0270
      GO TO 10                                                              15  0280
   20 IFF=I(3)                                                              15  0290
      GO TO 10                                                              15  0300
   30 M=(IFF-I(1))/2                                                        15  0310
      IF(IABS(M).EQ.0) GO TO 60                                             15  0320
      I(2)=I(1)+M                                                           15  0330
      I2=I(2)                                                               15  0335
      Q=USQO+TPHIX(I2)                                                      15  0340
      DSX(2)=SQRT(1.+VSQ/Q)                                                 15  0344
      EM=M                                                                  15  0350
      H=ABS(EM*DELX*.5)                                                     15  0360
      SS=H*(DSX(1)+DSX(2))                                                  15  0370
      Y=F-SS                                                                15  0380
      IF(Y.LT.0.)GO TO 40                                                   15  0390
      I(1)=I(2)                                                             15  0400
      DSX(1)=DSX(2)                                                         15  0410
      F=Y                                                                   15  0420
      GO TO 30                                                              15  0430
   40 IC=I(2)                                                               15  0440
   50 XC=DELX*FLOAT(IC-1)                                                   15  0450
      RETURN                                                                15  0560
   60 IC=IFF                                                                15  0565
      GO TO 50                                                              15  0570
   61 IC=IO                                                                 15  0680
      RETURN                                                                15  0690
      END                                                                   15  0700
```

```
$IBFTC XITP     DECK                                                        16  0000
      SUBROUTINE XITP                                                       16  0010
      COMMON/BSTOSS/USQ,VSQ,COSN,ALPHA                                      16  0020
      COMMON/BNXF/XO,XF,XC,XTP,FPATH,S,NQUAD,K,II                           16  0030
      COMMON/BMIN/XMIN,PHIMIN,IMIN,TPHID(33),TPHIX(1026),A,B,C,DELX,EN      16  0050
      COMMON/BNIF/IO,IC,ITP,PHIO,USQO                                       16  0040
      I1=EN*XO+1.5                                                          16  0100
      I2=IMIN                                                               16  0110
    1 M=(I2-I1)/2                                                           16  0120
      IF(M.EQ.0) GO TO 5                                                    16  0130
      I=I1+M                                                                16  0140
      TEST=USQO+TPHIX(I)                                                    16  0150
      IF(TEST)2,4,3                                                         16  0160
    2 I2=I                                                                  16  0170
      GO TO 1                                                               16  0180
    3 I1=I                                                                  16  0190
      GO TO 1                                                               16  0200
    4 ITP=I                                                                 16  0210
      GO TO 6                                                               16  0220
    5 ITP=I1                                                                16  0230
    6 XTP=DELX*FLOAT(I-1)                                                   16  0240
      DIF(K.EQ.1.AND.II.LE.5) DEBUG ITP,USQO,TPHIX(ITP-1),TPHIX(ITP),       16  0250
     1TPHIX(ITP+1)                                                          16  0260
      RETURN                                                                16  0270
      END                                                                   16  0280
```

```
$IBFTC CHEBY    DECK                                                          17031301
       SUBROUTINE CHEBY(MOVE)                                                17031310
C                                                                            17 31315
C  CHEBYSCHEV  POLYNOMIAL FIT TO SAMPLE DENSITIES FROM ITER SUBROUTINE       17 31316
C                                                                            17 31317
       DIMENSION TRNMAT(21,21),X(21),SUB(11,11),                             17031340
      2         UP(11),UPP(11),APP(11),AP(11),ASUBN(21),COEFS(21),Y(21)     17031350
       COMMON/BCHEB/ N1,X,Y,COEFS,EPS                                        17031360
       COMMON/BPHI/NPHI,    A(20),NDPHI,    AI(20),NDDPHI                    17031370
       COMMON/BCHEB2/N2                                                      17031370
       DOUBLE PRECISION PI,THETA,PION                                        17 31378
       DATA  PI/3.1415926535897932/                                          17 31379
       DATA  TRNMAT(1,1),TRNMAT(1,2),TRNMAT(2,2)/1.,-1.,2./                  17 31380
       EQUIVALENCE (NP1,N1)                                                  17031400
       GO TO (10,100),MOVE                                                   17031410
   10  N=N1-1                                                                17031420
       NPN = N+N                                                             17031430
       NPLUS2 = N+2                                                          17031440
       M = N/2                                                               17031450
       MP1 = M+1                                                             17031460
       MP2 = M+2                                                             17031470
       EN=N                                                                  17 31480
       PION=PI/EN                                                            17 31490
       DO 15  I=1,NP1                                                        17031500
       THETA = (1. - FLOAT(I-1)/EN)*PI                                       17 31510
       X(I)=DCOS(THETA)                                                      17 31520
   17  X(I)=(X(I)+1.)*.5                                                     17 31540
   15  CONTINUE                                                              17031550
   20  DO 25  I=3,NP1                                                        17031570
       TRNMAT (1,I) = -TRNMAT(1,I-1)                                         17031580
       DO 25  J=2,I                                                          17031590
   25  TRNMAT(J,I) = 4.*TRNMAT(J-1,I-1)-2.*TRNMAT(J,I-1)-TRNMAT(J,I-2)       17031600
   40  L=MP1                                                                 17031660
       DO 70 I=1,L                                                           17031670
       DO 70 J=1,L                                                           17031680
   70  SUB(I,J) = DCOS(FLOAT((J-1)*(I-1)) * PION)                            17 31690
       RETURN                                                                17 31700
  100  NP1 = NP1                                                             17031710
       UP(1) = Y(NP1)-Y(1)                                                   17031720
       UPP(1) = Y(NP1)+Y(1)                                                  17031730
       MOG=M-1                                                               17031740
       DO 104  I=1,MOG                                                       17031750
       NP1MI = NP1-I                                                         17031760
       UPP(I+1) = (Y(NP1MI)+Y(I+1))*2.0                                      17031770
  104  UP(I+1) = (Y(NP1MI)-Y(I+1))*2.0                                       17031780
       UPP(MP1) = 2.0*Y(MP1)                                                 17031790
       DO1 I=1,L,2                                                           17031810
       APP(I)=0.                                                             17031820
       AP(I)=0.                                                              17031830
       DO 2 J=1,L,2                                                          17031840
    2  APP(I)=APP(I)+SUB(I,J)*UPP(J)                                         17031850
       DO 1 J=2,L,2                                                          17031860
    1  AP(I) =AP(I) + SUB(I,J)*UPP(J)                                        17031870
       DO 3 I=2,L,2                                                          17031880
       AP(I)=0.                                                              17031890
       APP(I)=0.                                                             17031900
       DO 4 J=1,L,2                                                          17031910
    4  APP(I)=APP(I)+SUB(I,J)* UP(J)                                         17031920
       DO 3 J=2,L,2                                                          17031930
    3  AP(I) =AP(I) + SUB(I,J)* UP(J)                                        17031940
       DO 130  I=1,MP1                                                       17031970
  129  ASUBN(I) =(APP(I)+AP(I))/EN                                           17031980
       INDICE= NPLUS2-I                                                      17031990
  130  ASUBN(INDICE)=(APP(I)-AP(I))/EN                                       17032000
       ASUBN(1) = ASUBN(1)/2.0                                               17032010
       ASUBN(NP1) = ASUBN(NP1)/2.0                                           17032020
       DO 135 I=1,NP1                                                        17032030
       IF(ABS(ASUBN(I)      -EPS) 133,135,135                                17032050
  133  ASUBN(I)=0.0                                                          17032060
  135  CONTINUE                                                              17032070
       DO 140  I=1,NP1                                                       17032090
       COEFS(I) = 0.                                                         17032100
       DO 140  J=1,N2                                                        17032110
  140  COEFS(I) = COEFS(I)+TRNMAT(I,J)*ASUBN(J)                              17032120
       L=N1+1                                                                17032140
   11  L=L-1                                                                 17032150
       IF(COEFS(L).EQ.0.) GO TO 11                                           17032160
       NDDPHI=L-1                                                            17032170
 1000  RETURN                                                                17032260
       END                                                                   17032270
```

```
$IBFTC COEF     DECK                                                    19     11
       SUBROUTINE COEF(MODE)                                           19     20
       COMMON/BCHEB2/N2                                                19     24
       COMMON/BPHI/NPHI,APHI(20),NDPHI,ADPHI(20),NDEN                  19     25
       COMMON/BCHEB/N1,X(42),COEFS(22)                                 19     26
       COMMON/BMAIN/ CONST,VOLT(20),CURRNT(20)                         19 00027
       DATA KNTR/0/                                                    19 00028
       GO TO (1,2,3),MODE                                              19     30
C                                                                      19     40
C    MODE 1 = READ INITIAL APHI(2),N1,COEFS                            19     45
     1 CALL BCREAD(X(41),COEFS(11))                                    19     50
       APHI(2)= X(41)                                                  19     54
       NDPHI=X(42)                                                     19     55
       RETURN                                                          19     56
C                                                                      19     60
C    MODE 2 = COMPUTE COEFFICIENTS OF PHI AND DPHI                     19     65
     2 IF(KNTR.NE.0) NDPHI=N2                                          19 00070
       NPHI=NDPHI + 1                                                  19     75
       NDEN =NDPHI - 1                                                 19     80
       APHI(2)=VOLT(20)                                                U9     85
       DO 20 I=1,NDPHI                                                 19     90
       APHI(I+2) = CONST*COEFS(I)/FLOAT(I*(I+1))                       19     92
    20 APHI(2)=APHI(2)-APHI(I+2)                                       U9     94
       IF(KNTR.EQ.0) APHI(2)=A                                         19     96
       KNTR=1                                                          19     98
       DO 21 I=1,NPHI                                                  19    100
    21 ADPHI(I)= APHI(I+1)*FLOAT(I)                                    19    105
       RETURN                                                          19    110
C                                                                      19    120
C    MODE 3 = PRINT COEFS                                              19    125
C              PUNCH APHI(2),N1,COEFS                                  19    126
     3 WRITE(6,30)(COEFS(I),I=1,N1)                                    19    130
    30 FORMAT(1HO, 5HCOEFS/1H ,8F15.6/1H ,3F15.6)                      19 00135
       X(41)=APHI(2)                                                   19 00140
       X(42)=N2                                                        19 00145
       CALL BCDUMP(X(41),COEFS(11))                                    19 00146
       RETURN                                                          19    150
       END                                                             19    155
```

```
$IBFTC PLOT     DECK                                                   20 00011
       SUBROUTINE PLOT                                                 20 00020
C                                                                      20 00023
C  PLOTS OF FINAL DENSITY AND POTENTIAL DISTRIBUTIONS                  20 00024
       COMMON/BITER/NO,KI                                              20 00025
       COMMON/BCHEB/N1,U(21),V(21),COEFS(21),ERROR                     20 00030
       COMMON/BPHI/NPHI,APHI(20),NDPHI,AB(21)                          20 00040
       COMMON/BSTOSS/USQ,VSQ,COSN,ALPHA                                20 00046
       COMMON/BMAIN/ CONST,VOLT(20),CURRNT(20)                         20 00045
       DIMENSION PD(11),PP(11),X1(26),X2(26),D(26),PH(26)              20 00050
       DATA PD/26.,0.,5.,10.,0.,2.,-20000.,500.,4.,0.,1./              20 00060
       DATA PP/26.,0.,5.,10.,0.,5.,-200.,4.,4.,0.,1./                  20 00070
       DO 1 I=1,26                                                     20 00080
       X1(I) = .04 * FLOAT(I-1)                                        20 00090
       X2(I) = X1(I)                                                   20 00100
       D(I)=-DENS(X1(I))                                               20 00110
     1 PH(I)= - PHI(X1(I))                                             20 00120
       PP(7)= AINT(PH(26))-1.                                          20 00125
       PP(7)=10.*PP(7)                                                 20 00126
       CALL SORTXY(D,X1,26)                                            20 00130
       CALL SORTXY(PH,X2,26)                                           20 00140
       WRITE(6,2)                                                      20 00150
     2 FORMAT(2HPT, 50X,30HELECTRON DENSITY DISTRIBUTION  )            20 00160
       CALL PLOTXY(D,X1,118,PD)                                         20 0017
       WRITE(6,3)APHI(2),CONST,ALPHA,NO,N1,KI                          20 00180
     3 FORMAT(2HPL,20X, 29HAPHI(2),CONST,ALPHA,NO,N1,KI/ ,F5.2,1H,,F7.2, 20 00190
      11H,,E10.1,1H,, I6,1H,,I2,1H,,I2)                                20 00200
       WRITE(6,4)                                                      20 00210
     4 FORMAT(2HPT, 55X,22HPOTENTIAL DISTRIBUTION )                    20 00220
       CALL PLOTXY(PH,X2,118,PP)                                       20 00230
       WRITE(6,5)APHI(2),CONST,ALPHA,NO,N1,KI                          20 00240
     5 FORMAT(2HPL,20X, 29HAPHI(2),CONST,ALPHA,NO,N1,KI/ ,F5.2,1H,,F7.2, 20 00250
      11H,,E10.1,1H,, I6,1H,,I2,1H,,I2)                                20 00260
       RETURN                                                          20 00270
       END                                                             20 00280
```

```
$IBFTC CUMVEL  DECK                                                    21 00010
      SUBROUTINE CUMVEL                                                21 00020
      COMMON/BVEL/VEL(1024)                                            21 00030
      DATA NMC/1024/                                                   21 00035
      DELX=1./FLOAT(NMC)                                               21 00040
      DO 1 I=1,NMC                                                     21 00050
      X=DELX*(FLOAT(I-1)+.5)                                           21 00060
    1 VEL(I)=-ALOG(X)                                                  21 00070
      RETURN                                                           21 00080
      END                                                              21 00090
```

```
$IBFTC QUADGM  DECK
      SUBROUTINE QUADGM
C     MODIFIED GAUSS'MEHLER QUADRATURE                                 GMQU0030
C     NUMERICAL INTEGRATION OF FOFX(X)/SQRT(X-XO) FROM XO TO XF        GMQU0040
      COMMON/BNIF/IO,IC,ITP,PHIO,USQO
      COMMON/BNXF/XF,XI,XC,XO ,FPATH,S,NQUAD ,K,II
      COMMON/BSTOSS/USQ,VSQ,COSN,ALPHA
      DIMENSION Y(33),A(33)                                            GMQU0050
      REAL INTGRL                                                      GMQU0060
      DATA N/5/                                                        GMQU0062
      DATA (Y(I),A(I),I=1,33)/                                         GMQU0070
     1 0.56939116E-01, 0.93582787E-00, 0.43719785E-00, 0.72152315E-00, GMQU0080
     1 0.86949939E-00, 0.34264898E-00, 0.33648268E-01, 0.72536757E-00, GMQU0090
     1 0.27618431E-00, 0.62741329E-00, 0.63467748E-00, 0.44476207E-00, GMQU0100
     1 0.92215661E-00, 0.20245707E-00, 0.22163569E-01, 0.59104845E-00, GMQU0110
     1 0.18783157E-00, 0.53853344E-00, 0.46159736E-00, 0.43817273E-00, GMQU0120
     1 0.74833463E-00, 0.29890270E-00, 0.94849393E-00, 0.13334269E-00, GMQU0130
     1 0.15683407E-01, 0.49829409E-00, 0.13530001E-00, 0.46698507E-00, GMQU0140
     1 0.34494238E-00, 0.40633485E-00, 0.59275013E-00, 0.32015666E-00, GMQU0150
     1 0.81742801E-00, 0.21387865E-00, 0.96346128E-00, 0.94350673E-01, GMQU0160
     1 0.11675872E-01, 0.43052771E-00, 0.10183270E-00, 0.41039693E-00, GMQU0170
     1 0.26548116E-00, 0.37107680E-00, 0.47237154E-00, 0.31440633E-00, GMQU0180
     1 0.68426202E-00, 0.24303714E-00, 0.86199133E-00, 0.16031617E-00, GMQU0190
     1 0.97275575E-00, 0.70238921E-01, 0.90273770E-02, 0.37890122E-00, GMQU0200
     1 0.79300560E-01, 0.36520683E-00, 0.20977937E-00, 0.33831304E-00, GMQU0210
     1 0.38177105E-00, 0.29919198E-00, 0.57063582E-00, 0.24925794E-00, GMQU0220
     1 0.74931738E-00, 0.19031702E-00, 0.89222197E-00, 0.12450705E-00, GMQU0230
     1 0.97891421E-00, 0.54304919E-01/                                 GMQU0240
      FOFX(X,Y)=SQRT(ABS(X-XO) *(1.+VSQ/Y))                            GMQU0242
      XOFY(Y)=XO+(XF-XO)*Y                                             GMQU0250
      INTGRL=0.                                                        GMQU0260
      MIN=N*(N-1)/2 -2                                                 GMQU0270
      MAX=MIN+N-1                                                      GMQU0280
      DO 210 J=MIN,MAX                                                 GMQU0290
      X=XOFY(Y(J))                                                     GMQU0300
      Z=USQO+PHI(X)                                                    GM  0310
      IF(Z.LE.0.) GO TO 211                                            GM  0312
      F=FOFX(X,Z)                                                      GM  0314
  210 INTGRL=INTGRL+A(J)*F                                             GMQU0320
      INTGRL=SQRT(ABS(XF-XO))*INTGRL                                   GMQU0330
      S=ABS(INTGRL)                                                    GMQU0332
      RETURN                                                           GMQU0340
  211 S=0                                                              GMQU0342
      RETURN                                                           GMQU0344
      END                                                              GMQU0350
```

APPENDIX D

SYMBOLS

[All dimensioned variables in cgs-esu units.]

a_k	coefficients, eq. (24)
C	dimensionless constant, eqs. (6), (8), and (10)
c_j	coefficients, eq. (25)
E[ ]	expectation value of [ ], eq. (A4)
e	electronic charge, eq. (7)
$F_u(V), F_V(u)$	marginal distributions, eqs. (12) and (13)
f(u)	probability distribution function, eq. (A2)
f(u, V)	dimensionless velocity distribution function, eq. (4)
g(X)	function of random variable X, appendix A
$\bar{g}_N$	sample mean of g(x), eq. (A20)
J	electron current to collector, eq. (22)
J_o	electron emission current, eqs. (8) and (22)
k	Boltzmann's constant, eq. (5)
L	interelectrode separation, eq. (2)
ℓ	dimensionless path length, eq. (3)
ℓ_c	dimensionless path length for collision, eq. (2)
m	mass of electron, eq. (5)
N_c	number of electrons striking collector, eq. (22)
N_o	total number of histories, eq. (22)
n	dimensionless electron density, eq. (6)
$\hat{n}$	electron density, eq. (7)
n_o	electron density of emitted flux, eq. (7)
P[ ]	probability of [ ], appendix A
p(r)	uniform probability distribution function, eq. (A7)
R_k	uniformly distributed random numbers
s	path length, eq. (3)
T	emitter temperature, eq. (5)
U	random variable, appendix A
u	dimensionless x-component of velocity, eqs. (5) and (9)
u_o	initial velocity, eq. (20)
V	dimensionless velocity component transverse to the x-direction
$\mathscr{V}(x)$	potential distribution

v_o	initial velocity of monoenergetic emission, eq. (9)
v_x, v_y, v_z	components of velocity
X, Y	random variables, appendix A
x	spatial coordinate, eq. (7)
y	dimensionless spatial coordinate, eq. (7)
y_c	location of collision, eq. (29)
y_o	location of last event, eq. (29)
α	dimensionless reciprocal mean free path, eq. (3)
Γ_c	flux to collector
Γ_o	emitted flux
θ	scattering angle, eq. (1)
θ^*	capture angle, eq. (34)
λ	mean free path, eq. (3)
$\hat{\sigma}_g$	theoretical standard deviation, eq. (A20)
σ_g	sample deviation, appendix A
σ_J	standard deviation of current to collector
φ	dimensionless potential distribution: thermionic emission, eq. (7); monoenergetic emission, eq. (9)
Ω	solid angle, eq. (1)
[ ]	integral value
{ }	sequence of terms { }, appendix A

REFERENCES

1. Grad, H.: Theory of Rarefied Gases. Proc. Int. Symposium on Rarefied Gas Dynamics, Devienne, F. M., ed., Pergamon Press, 1959, p. 100.

2. Sockol, Peter M.: Flow of Electrons Through a Neutral Scattering Gas in a Thermionic Diode. Report on the Thermionic Conversion Specialist Conference, IEEE, Cleveland (Ohio), October 1964, pp. 170-177. (Also available from TIS, AIAA as A65-16980.)

3. Langmuir, I.: The Effect of Space Charge and Initial Velocities on the Potential Distribution and Thermionic Current Between Parallel Plane Electrodes. Phys. Rev., vol. 21, Apr. 1923, pp. 419-435. (See also The Collected Works of Irving Langmuir. Vol. 3. Pergamon Press, 1961, pp. 95-110.)

4. Langmuir, I.; and Jones, H. A.: Collisions Between Electrons and Gas Molecules. Phys. Rev., vol. 31, Mar. 1928, pp. 357-404. (See also The Collected Works of Irving Langmuir. Vol. 5. Pergamon Press, 1961, pp. 60-110.)

5. Meyer, H. A., ed.: Symposium on Monte Carlo Methods. John Wiley & Sons, Inc., 1956, p. vi.

6. Goertzel, Gerald; and Kalos, Malvin H.: Monte Carlo Method in Transport Problems. Prog. in Nuclear Energy, ser. 1, Phys. and Math., Vol. 2, D. J. Hughes, J. E. Sanders and J. Horowitz, eds., Pergamon Press, 1958, pp. 315-369.

7. Brown, G. W.: Monte Carlo Methods. Modern Mathematics for the Engineer, E. F. Beckenbach, ed., McGraw-Hill Book Co., Inc., 1956, pp. 279-303.

8. Kraft, R.; and Wensrich, C. J.: Monte Carlo Methods, A Bibliography Covering the Period 1949 to 1963. Rept. No. UCRL-7823, Lawrence Radiation Lab., Apr. 1, 1964.

9. Fleck, J. A., Jr.: The Calculation of Nonlinear Radiation Transport by a Monte Carlo Method. Methods in Computational Physics, Vol. 1, B. Alder, ed., Academic Press, 1963, pp. 43-65.

10. Burger, P.: The Opposite-Stream Plasma Diode. Rept. No. SEL-64-012, Stanford Electronics Lab., 1964, p. 41.

11. Itoh, T.; and Musha, R.: Monte Carlo Calculations of the Motions of Electrons in Helium. J. Appl. Phys., vol. 31, no. 4, Apr. 1960, pp. 744-745.

12. Kennard, E. H.: Kinetic Theory of Gases. McGraw-Hill Book Co., Inc., 1938.

13. Parzen, E.: Modern Probability Theory and Its Applications. John Wiley & Sons, Inc., 1960.

14. Goldstein, C. M.; and Goldstein, A. W.: Effect of Electron Neutral Elastic Scattering on Low-Pressure Diode Characteristics. Bull. Am. Phys. Soc., ser. 2, vol. 9, no. 4, 1964, p. 469.

15. Kahn, Herman: Applications of Monte Carlo. Rept. No. RM-1237-AEC, Rand Corp., Apr. 27, 1956, p. 91, ff.

16. Lanczos, Cornelius: Applied Analysis. Prentice Hall, Inc., 1956, p. 229, ff.

17. Danielson, G. C.; and Lanczos, C.: Some Improvements in Practical Fourier Analysis and Their Application to X-Ray Scattering From Liquids. J. Franklin Inst., vol. 233, no. 4, Apr. 1942, pp. 365-380.

18. Kopal, Zdenek: Numerical Analysis. Chapman & Hall, Ltd., 1961, p. 347, ff.

3/18/25

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Technical information generated in connection with a NASA contract or grant and released under NASA auspices.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

TECHNICAL REPRINTS: Information derived from NASA activities and initially published in the form of journal articles.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities but not necessarily reporting the results of individual NASA-programmed scientific efforts. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546