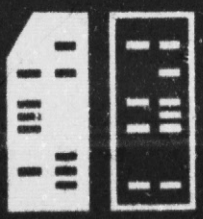
MONTE CARLO SIMULATION OF NONLINEAR RADIATION INDUCED PLASMAS

by

Benjamin Shaw-hu Wang

May, 1972



DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

MONTE CARLO SIMULATION OF NONLINEAR RADIATION INDUCED PLASMAS

BY

BENJAMIN SHAW-hu WANG
B.S., National Taiwan University, 1960
M.S., University of Missouri (Rolla), 1964

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1972

Urbana, Illinois

MONTE CARLO SIMULATION OF NONLINEAR RADIATION INDUCED PLASMAS

Benjamin Shaw-hu Wang, Ph.D.
Department of Computer Science
University of Illinois at Urbana-Champaign, 1972

A Monte Carlo simulation model for radiation induced plasmas with nonlinear properties due to recombination has been developed, employing a piecewise linearized predict-correct iterative technique. Several important variance reduction techniques have been developed and incorporated into the model, including an antithetic variates technique (negatively correlated variates) which has proven to be most useful. This approach is especially efficient for plasma systems with inhomogeneous media, multidimensions, and irregular boundaries, where other analytic or numerical solutions are either unrealistic or impractical. This model is quite general in scope and should be applicable to a variety of similar particle transport and diffusion-type plasma problems.

The Monte Carlo code developed in this work has been applied to the determination of the electron energy distribution function and related parameters for a noble gas plasma created by α-particle irradiation. The radiation induced plasma involved was characterized by the following features:

1) a continuous internal volume-source of high energy electrons

2) an applied electric field (optional)

3) secondary electron production via ionization processes

4) losses via recombination and leakage across boundaries

The calculations were specifically carried out for a helium gas medium with an electron source rate from 10^{14} to 10^{22} (#/cm^3-sec), initial electron energies from 70 to 1000 eV, pressures from 10 to 760 torrs and E/P (electric field/pressure) ratios from 1 to 10 V/cm-torr.

For the lower source rates, it is observed that, in the zero electric-field case, the low-energy portion of the distribution function is quite close to a standard Maxwellian distribution. However, the high-energy portion (above the ionization potential) of the distrib..ions is a rapidly decaying parabolic-shaped tail which can be crudely represented by a 1/(Energy) distribution. The addition of the electric field causes the distribution function in the low-energy region to shift towards Smit's (or Druyvesteyn's) distribution while the high-energy protion essentially retains its zero field shape. As the source rate increases beyond certain value (roughly 10^{18} cm^{-3}-sec^{-1} in the present case), the low-energy region begins to depart from the standard Maxwellian or Smit distributions and this departure can become quite large.

A major emphasis in this development was placed on computation efficiency. The present computation scheme with its variance and accelerated convergence techniques is thought to be quite efficient compared to alternate approaches.

ACKNOWLEDGEMENT

The author wishes to express his sincere appreciation and gratitude to Professor George H. Miley for his patient guidance and invaluable help throughout the course of this work, without which this study would have been impossible. The author wishes to also thank Mr. Ronnie Lo for his most valuable and helpful discussions.

The author would also like to thank Professor J. N. Snyder, the head of Computer Science Department, and the Nuclear Engineering Program for their financial support. This work was supported in part by the Physics Branch of the AEC Research Division under Grant No. AT(11-1)-2007 and by the NASA-AEC Space Power Office under Grant No. NGR 14-005-183.

Finally, the author wishes to thank Mrs. Sharon Taylor for her excellent typing of the final thesis manuscript.

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

A. Statement of the Problem and Objectives

Two major goals were set for this study. The first was to develop an efficient Monte Carlo simulation technique for solving radiation-induced plasma problems. This required adapting several variance reduction techniques in order to increase the Monte Carlo efficiency. The second goal was to apply the model developed herein to study the characteristics of a noble-gas plasma created by alpha-particle irradiation.

The major point of interest is the determination of the electron energy distribution in the nuclear radiation induced plasma. This distribution is of direct importance to current experimental studies at the University of Illinois where the possibility of directly pumping* gas lasers with nuclear radiation is under study (6). Recent results from these studies also show that nuclear radiation can be used to enhance the output from some type of electrically pumped lasers (5).

B. The Importance of Nuclear Radiation Produced Plasmas and the Physical Model

Nuclear radiation can be used in two ways: to directly pump the laser, or alternately, where an electric discharge

---

*"Pumping" of a laser is defined as the introduction of energy into the lasing medium required for population inversion. The electrical current which creates the discharge in the conventional gas laser is the pump in that case. Flash lamps are used for many solid state lasers.

serves as the main pump, to enhance the operation of lasers. The possibility of efficient coupling of lasers and nuclear reactors has generated an interest in the use of high-energy ion beams to pump lasers. To avoid radiation damage problems, gas lasers are generally thought of in this connection.

There are several important reasons for searching for more direct methods of coupling energy from reactor sources into lasers (8). One is the possibility that the resulting laser may offer unique characteristics, e.g., new frequencies and new methods of modulation and control. Simplified coupling could also offer weight reductions and improved reliability, which may be of advantage in space applications. Also, it may eventually become feasible to simultaneously extract several energy forms, e.g., laser and electrical power from a single nuclear station.

In most gas lasers the excited states, and consequently the inversion of population necessary for laser operation, are formed directly or indirectly by electronic collisions in a plasma generated by an electrical discharge. However, an alternate way to accomplish this goal is via nuclear radiation which can produce ionization and excitation through both direct interactions and interactions due to secondary-electron production. Given the high neutron fluxes available from nuclear reactors, the particles from neutron-induced nuclear reactions (e.g., α-particles from a Boron coating on the wall of a laser tube) can furnish the nuclear radiation

source. The description of the transport of both the primary radiation and the secondary electrons, from their point of birth throughout their path in the gas, is a key problem in the quantitative evaluation of the plasma. Therefore the development of nuclear-radiation pumping requires a thorough understanding of the radiation-induced plasma.

The following physical model is applied to the situation of a radiation-induced plasma. Pertinent quantities of interest include: electron energy distribution spectrum; average electron energy; and rate of energy loss.

The heavy-charged particles (α-particles) are born in coating and enter an adjacent gas. In the present calculations, the cylindrical geometry of the actual laser tube is approximated by a slab geometry (cf. Figure 1) with two parallel plane boron coatings separated by a gaseous medium. The α-particle radiation continuously creates high energy electrons throughout the gas volume (with energy E_o). The resulting plasma is a weakly ionized, three-component plasma (electrons, positive ions, and neutral gas atoms). It is assumed that the alpha-particle trajectory can be treated analytically using methods such as developed in (7). Further, this reference shows how to obtain the primary secondary-electron (δ-ray) energy spectrum. The difficult problem, however, is to follow the thermalization and further secondary production of these primary electrons.

Thus, the Monte Carlo simulation starts off by following these electrons on their life time histories, i.e., by simulating the course of their thermalization. Electrons go through various processes of collisions and scattering, resulting in a complex process of diffusion, energy degradation and secondary production, but finally,they recombine with positive ions or escape from the system,ending their histories.

With the radiation rates of interest here, the plasma is only weakly ionized and electrons mostly thermalize by collisions with the neutral background-gas atoms. In other words, collisions with ions and other free electrons can be neglected. In this sense, the problem is simplified in that it remains linear. However, some non-linearity enters in the following way. The recombination process depends on the ion density and spatial distribution. However, the ions are originally formed by ionizations which occur as the electrons slow down.

Another important characteristic is that the plasma is highly non-equilibrium. The thermalized electrons and ions form an approximately Maxwellian background, but superimposed on this is the high energy tail due to the electrons in the process of slowing. Also, the alpha particles themselves represent a non-equilibrium component.

In summary, the special features of the plasma system are:

1) Continuous internal volume source.

2) An applied electric field may or may not be present.

3) Finite cylindrical geometry--approximated here by slab geometry.

4) Electron-ion pair production and recombination processes.

5) The energy range of the electrons range from kilovolts to thermal (eV.).

6) The system represents a non-equilibrium plasma, but in a steady-state condition (or stationary).

7) Due to relatively small densities of electrons and ions compared with that of gas atoms, direct interactions between the electrons themselves or electron-ion collisions are neglected.

C. General Approaches and Methods of Solution

Possible approaches toward solutions of these types of problems (particle transport and diffusion) can be categorized as follows:

1) Analytical solution of the governing Boltzmann transport equation.

2) Numerical solution of the Boltzmann transport equation.

3) Direct analog simulation (Monte Carlo).

4) Mixtures of the above approaches.

The governing equation for this type of problem is the Boltzmann equation, and most classical methods have involved approximate analytic solutions of it. Solutions are possible only after various approximations and simplifications are made, and such solutions are often very restrictive and/or difficult to extend to actual situations of interest. More

or less the same drawbacks apply to numerical solutions. However, in this case approximations are necessary or the computation time becomes too long, especially as the dimensionality and the complexity of the problem increases. In fact, the computation time increases roughly exponentially with the dimensionality of the problem. Usually the governing partial differential and integral equations (coupled Boltzmann equations) are reduced to finite difference equations, and the requirements for stability and convergence severely limit the maximum step sizes of the independent variables. Further difficulties with the practical application of such approaches arise because it is often difficult to include realistic boundary conditions.

Direct simulation by the application of Monte Carlo techniques seems to be the better approach; sometimes the only approach for such problems. It by-passes the direct solution of the partial differential-integral equation, and generally no approximations and/or simplifications are necessary. Complicated boundary conditions can be handled easily, and the computation time increases at the most linearly with the dimensionality of the problem. However, computer time still may be a problem so that judicious selection of numerical schemes and variance reduction techniques are required.

Many methods have been devised in the past which involve a mixture of analytic, numerical, and Monte Carlo approaches.

In such situations, Monte Carlo is usually applied to the most difficult part of the problem. For instance, in the evaluation of the non-linear, five-dimensional collision integral in the Boltzmann equation, straight forward numerical quadrature would require months of computation time even on the present day fast computers. The Monte Carlo (statistical sampling techniques which closely resemble the actual collision phenomena) techniques can produce results within reasonable computation time (24). The Monte Carlo method may be defined as a numerical device (numerical experimentation) for studying an artificial stochastic model of a physical or mathematical process. This study may be approached by one of two methods. The first is the process of simulation (direct analog simulation) in which the particles - electrons and ions in the case of plasma systems, or photons in the case of radiative transfer - are represented by a game of chance in the computer. In the second method, which is non-simulation, the Monte Carlo technique is strictly used as a numerical stochastic device for the solution of a given integral or partial differential equations (24,25).

CHAPTER II

THE MATHEMATICAL MODEL: NUMERICAL SCHEME AND PHYSICAL PROBLEM

A. General Description of the Physics of the Problem

The plasma under study is a weakly-ionized three-component plasma, and is non-equilibrium as described in Chapter I. The α-particle radiation continuously creates high-energy electrons throughout the gas volume. The Monte Carlo simulation is to trace these electrons' life histories and their complex processes of diffusion and energy degradation. Since the diffusion process of the electrons through the medium is affected by the density of the positive ions and the other electrons present (background electrons), the resulting problem is non-linear.

1. Physical Processes Involved

The mechanisms by which particles interact with the atoms of the medium are elastic and inelastic collisions, by which the incident particle is slowed and dissipates some or all of its kinetic energy to the medium by the combined action of the above mentioned processes. The particle can also experience so called superelastic collision by which the particle collides with the excited atoms gaining some energy. Finally, the particle can collide and recombine with positive ions or can leak out of the system thus ending its life

history.

As described in Chapter I, the type of interaction which is involved in any particular collision is governed by the laws of chance. For large numbers of interactions, the frequency of occurrence of any given interaction is determined by the relative magnitudes of the interaction cross sections. Secondary knock-on electrons due to the ionization process may be capable of further interacting with the atoms and producing additional secondary electrons.

As the system reaches steady-state, a balance is reached such that the birth rate for new particles, i.e., the rate at which primary or source electrons enter the system plus the production rate of secondary electrons due to ionization should approximately equal the loss rate due to leakage and absorption or recombination.

Since the transport of radiation through a medium affects its absorption properties, the transport of individual particles is indirectly affected by the density of the other particles present, and this introduces some non-linearities into the system. We shall adopt a piecewise predict-correct technique to approximate the nonlinear system with a linear model.

We are concerned with the simulation, by random sampling of the scattering of the charged particles. The direct simulation of the physical scattering processes in such a way that the diffusion process is imitated by letting

the particles carry out a random walk, each step of which takes into account the combined effect of many collisions. The mechanisms by which electrons interact with the atoms of the medium are elastic, inelastic, and superelastic collisions as well as recombination with positive ions. The incident source electron is slowed and dissipates some or all of its energy to the medium by the combined action of the various scattering processes, until it escapes, or recombines with an ion (radiative energy losses are neglected). The probability that an interaction occurs is determined by the total collision cross-section. The type of interaction is then determined by the relative magnitudes of the individual cross-sections.

2. Relation to the Transport Equation

Although no direct use will be made of the governing Boltzmann transport equation in the sequel, we shall write it down briefly, in order to indicate the mathematical problems to be solved implicitly by the Monte Carlo method. It is a nonlinear integro-differential equation of the form

$$\frac{\partial f}{\partial t} + \bar{u} \cdot \nabla_r f - \frac{|e|}{m} (\bar{E} + \bar{u} \times \bar{B}) \cdot \nabla_u f = \left(\frac{\partial f}{\partial t}\right)_{\text{collisions}} \quad (1)$$

where f: number density distribution function

$\bar{B}$: magnetic field

$\bar{u}$: velocity of the particles

$\bar{E}$: electric field strength

t: time

r: spatial variable

In our situation $\overline{B}=0$ since the magnetic field does not exist. Further, if we assume an infinite medium and uniform source, $\nabla_r f=0$, the distribution function f depends only on energy. Then the above equation reduces to

$$\frac{\partial f}{\partial t} - \frac{|e|}{m}\vec{E} \cdot \nabla_u f = \left(\frac{\partial f}{\partial t}\right)_{\text{collisions}} \qquad (2)$$

The right hand side term is the collision term, which is the most difficult part of the equation. The effects of all relevant particle interactions must be included in this term. For the case of electron collisions with stationary gas atoms of uniform density, we can write the collision term as:

$$\left(\frac{\partial f}{\partial t}\right)_{\text{collision}} = \left(\frac{d N_{in} - d N_{out}}{dV \cdot dt}\right) \qquad (3)$$

where dN_{in} gives the number of phase space points (electrons) that are scattered into the phase-space volume dV by collision in time dt. Similarly, dN_{out} represents the number of points scattered out of dV by collisions in time dt. This Eq. 2 states that the change in the distribution function f are caused by collisions of various kinds and the external force due to the electric field.

The assumptions that enter into the derivation of the transport equation which carry over to the Monte Carlo

simulation include:

a) the scattering centers (background atoms and ions) are distributed at random, i.e., possible correlations between the positions of various atoms and ions are not taken into account.

b) In the absence of an electric field, the trajectory of the particle is idealized as a zig-zag path, consisting of flights interrupted by sudden collisions in which the energy and direction of the particle changed. The quantum-mechamical interference (electron diffration) resulting from the coherent scattering by several centers are neglected, and a particle, in the course of traversing the medium, interacts with one scattering center at a time.

The diffusion process in terms of the transport equation is analogous to the use of Eulerian coordinates in hydrodynamics, in that one asks about the flux at given points in space. By contrast, the Monte Carlo method uses Lagrangian coordinates; one attaches a label to a particular bit of fluid, i.e., a diffusing particle, and follows its history. Then, by sampling many histories, one is able, at least in principle, to solve any diffusion problem.

B. The Stochastic Model - Numerical Scheme

The essential approach to the nonlinear problem adopted here is the use of piecewise linearized steps. In each such step (an amount of time Δt along the time axis), the

dominate parameters, which characterize the nonlinear properties, are predicted and considered to be fixed with respect to the independent variables. During the time Δt, the system can be considered as more or less linear, so the conventional Monte Carlo simulation technique is used. The quantities pertaining to the given system are computed for each of a reasonably large number of independent particle histories. The desired parameter values are obtained by averaging the results over all of the histories. At the end of the step, the dominate parameters are corrected before beginning the next step. In each step (time interval Δt, or cycle), instead of computing the trajectory and angular deflection of an electron after each interaction, we perform a random walk calculation. Since each step of the random walk takes into account the combined effects of many individual interactions, we form a so-called condensed history by this approach. Let us describe the model in some detail, step by step.

1. The Geometry and the Coordinate System

The initial spatial geometry in this problem is a one-dimensional, finite slab (from z=0 to z=D) which is subdivided into uniform intervals called zones as shown in Figure 1. The energy range is from 0 to E_o (=1keV) which is also subdivided into three regions: $0 \sim 1$ (thermal energy region), 1 to ED2, and ED2 to E_0. Each of these

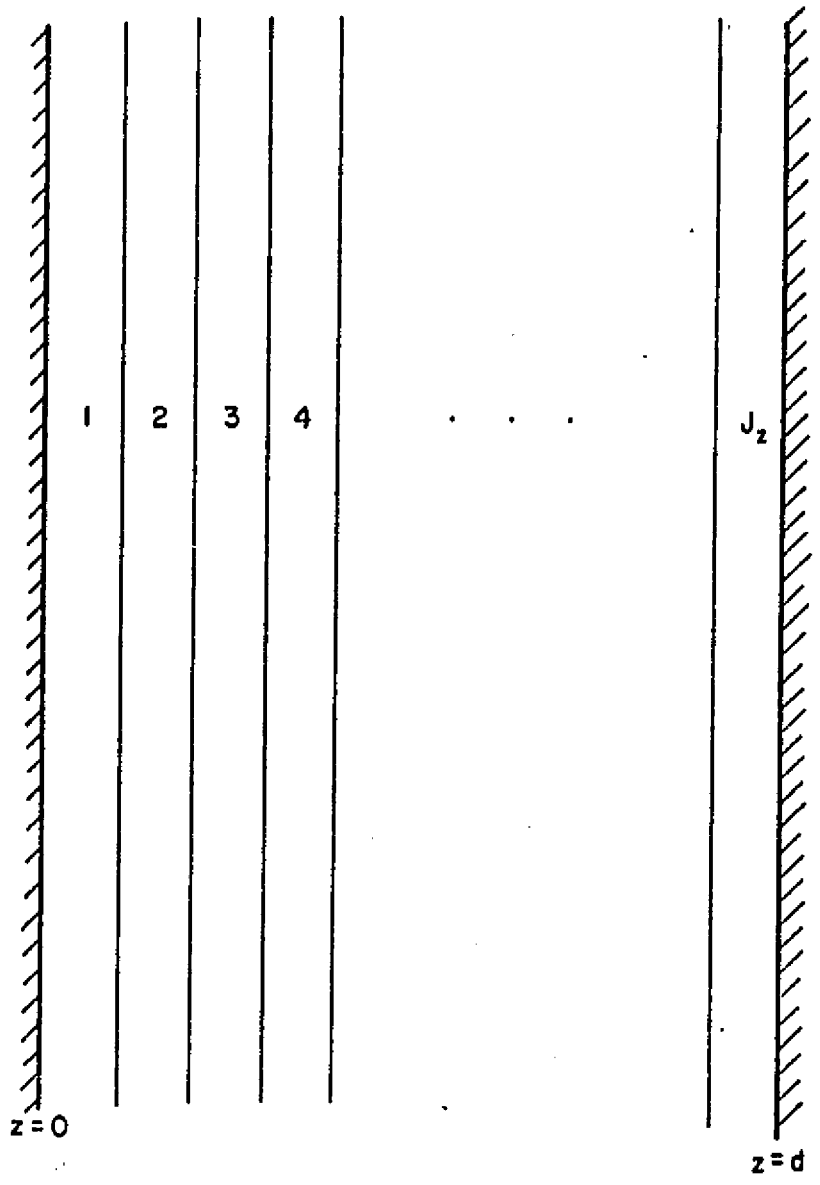


Figure 1. Slab geometry of the medium with subdivided zones.

regions is further divided into smaller uniform or logarithmic spacing (intervals). The particle number density distribution f(E,z) is a function of energy E and spatial position z and is generated such that particles with energy and spatial position falling into the same intervals are grouped together and normalized by the total particles processed. Although the system is stationary, we deliberately introduce the time variable for the convenience of recording output information and carrying out the calculation steps. The time serves as a clock, or we can imagine that the system begins in a transient state. Then, as time goes on it will eventually reach steady state. The storage of the timing information for individual particles is not necessary. Instead of following a single particle from the beginning to the end of its history, we follow a large number of particles, say N (which represent the sample particles of the whole system population) through a small time interval Δt (called the cycle time). Within this time interval, the system is considered to be linear, and the particles are processed more or less independently as in a conventional linear Monte Carlo model. At the end of Δt, called census time, we record data information for each particle (record particle state information). During this time interval (or present time cycle) some particles may experience collisions of some kind, some may not, and new particles (source particles, secondary particles produced

by ionization) enter the system. Some particles may be absorbed or escape from the system; their histories are therefore ended. This represents a random walk calculation, and each step of the random walk takes into account the combined effect of many individual interactions. The trajectory of a particle which is computed by this approach is called a condensed history. The residue particles (ionization produced secondary particles) are also set into motion, and their histories are followed separately. They are like all the other source particles in all respects. The complete description of particle histories is approximated by the condensed histories, or we take sequences of snapshots of all the particles in the system at various times to provide a moving picture of the history that can be used to estimate the quantities of interest. The path lengths travelled by the particles are also used as a clock to measure time, or equivalently, the variable t may be replaced by $S(t) = \int_{t_o}^{t} v(x)dx$. The trajectories of the particles (the condensed histories) are described by the arrays as indicated in Figured 2. The superscript denotes the cycle time number; the subscript denotes particle number where the index 0 refers to the initial state of the particle. Other notation includes E: the particle energy, z: spatial position, and U: the direction, and wt: the weight. When the particle has traveled a path-length DCEN (in the same duration Δt), a condensed history

i \ t	t_o	t_1		t_j		t_m
	E_1^0	E_1^1	. . .	E_1^j	. . .	E_1^m
	Z_1^0	Z_1^1		Z_1^j		Z_1^m
1	U_1^0	U_1^1		U_1^j		U_1^m
	wt_1^0	wt_1^1		wt_1^j		wt_1^m
	E_2^0	E_2^1	. . .	E_2^j	. . .	E_2^m
	Z_2^0	Z_2^1		Z_2^j		Z_2^m
2	U_2^0	U_2^1		U_2^j		U_2^m
	wt_2^0	wt_2^1		wt_2^j		wt_2^m
.	.	.		.		.
.	.	.		.		.
.	.	.		.		.
	E_n^0	E_n^1		E_n^j		E_n^m
	Z_n^0	Z_n^1		Z_n^j		Z_n^m
N	U_n^0	U_n^1		U_n^j		U_n^m
	wt_n^0	wt_n^1		wt_n^j		wt_n^m

Figure 2. Snapshops of the particles' condensed histories.

is sampled. Letting the particle carry out a random walk in each step (from state M to state M+1) takes into account the combined effect of many collisions. The sizes of the steps of the random walk, i.e., the pathlength intervals DCEN (equivalent to Δt) are chosen such that the total number of steps should be kept as small as possible. Too large a step size (Δt) may cause inaccuracies in the approximation. However, the length of the Monte Carlo calculation is directly proportional to the total number of steps. Thus, a compromise must be reached such that we choose as big a step size as possible without introducing intolerable inaccuracies. A reasonable initial value is chosen such that the particle with the average energy of the time cycle travels one mean free path, i.e., $\Delta t = D_{avg}/V_{avg}$. Different time cycles have different E_{avg} (V_{avg}) and D_{avg} so Δt may vary, but as steady-state is approached, Δt will not change much. By this technique we not only randomize and average the different particle samples, ensemblewise, but we also do the same (on the particle ensemble sets) along the time axis (averaging independent but equivalent time cycles). Trial and error results indicate that the optimum values of Δt (or DCEN) lie between 0.5 (D_{avg}/V_{avg}) 2.0 (D_{avg}/V_{avg}).

The maximum number of particles (N) that can be processed in each cycle is limited by the amount of storage available in the computer system. For the IBM 360/75 system

at the Illinois Computer Center, the storage of the data for 5,000 particles is easily accomplished. By playing Russian roulette, splitting, or similar techniques, the total number of artificial particles (N) can be increased by an order of magnitude. The end of one time cycle t_i is the start of the next time cycle t_{i+1}. While keeping track of the particles in a current time cycle at one time, we simultaneously record the data information after each particle reaches the census and extract the output parameter values after all the particles have been processed. In this way the storage requirements are reduced to a minimum level possible.

The limitation on N by the storage requirement may be further relaxed by the following technique. Instead of recording and tallying the output information and clearing or reinitializing all parameter values at the end of every cycle, we record the accumulated information for K continuous, and similar but statistically independent time cycles, then renormalize and reinitialize. In this way the total number of particles processed is KN, and if processing of a large number of particles is desired, we can use a larger value of K($K>1$) at the expense of more processing time. The reason for this is based on the fact that the system being simulated is time independent. We have introduced the time variable artificially so that ergodicity holds, namely ensemble averaging is the same as time averaging.

2. Tracing Procedure

The simulation is initialized by introducing N particles with energy E_{in} [sampled from an initial guess for the number density distribution vs. energy], with spatial position z_i^0 (uniformally distributed throughout the spatial region), with direction u_i^0 ($u=\cos\theta$; θ: angle of flight with z axis) which is uniformly distributed between -1 and +1 and with weight wt_i^0. Volume particles are emitted uniformly throughout the medium and isotropically in direction; namely, at $t=t_o$:

$$E_i^0 = E_{in} \tag{4a}$$

$$Z_i^0 = r_i \cdot D \tag{4b}$$

$$u_i^0 = 2r_i - 1 \tag{4c}$$

where r_i: random number, uniformly distributed between 0, 1, or $r_i \in (0, 1)$

d: slab width or cylinder diameter

wt_i, E_{in}: weight and particle energy determined from initial distribution.

Once the particle data are determined, the geometry routine is entered. This includes the determination of the distance to boundary d_B, the distance to collision d_{col}, and the distance to census time d_{cen} (clock distance equivalent to Δt). These distances are defined by:

$$d_B = [(j+1)\Delta Z - Z]/u \quad \text{if } u > 0 \tag{5}$$

$$= (Z - j\Delta Z)/|u| \quad \text{if } u < 0$$

$$= \text{DLB (a large number DLB} \gg \Delta Z) \text{ if } u \approx 0$$

$$d_{col} = \lambda(E) \left|\ln r\right| = \left|\ln r\right| / \Sigma_t(E) \tag{6}$$

$$d_{cen} = D_{avg} \cdot DI = V_{avg} \cdot \Delta t \cdot DI \tag{7}$$

where r: random number uniformly distributed

j: zone number

Δz: spatial interval size

Σ_t: the total cross-section evaluated at the particles' energy

D_{avg}: the average value of D_{col} of N particles of the previous cycle. It is approximately one mean free path corresponding to E_{avg}: the average energy of the cycle.

DI: input parameter for controlling the cycle time interval.

The next step depends on which of these distances is the smallest.

a. <u>The System Without Electric Field (E_f=0)</u>

i) If the smallest distance is d_{cen}: the particle is advanced in position and time to census time.

$$z' = z + u.d_{cen} \tag{8a}$$

$$u' = u \tag{8b}$$

$$E' = E \tag{8c}$$

The particle data is stored, the energy is tallied, and

the density counter is incremented. Then we go on to process the next particle.

ii) If the smallest distance is d_B, the particle is advanced to the boundary such that:

$$z' = z + (d_B + 0.001 \cdot \Delta z) \cdot u \tag{9a}$$

$$j = \text{INT}\,(z'/\Delta z) + 1 \tag{9b}$$

$$d'_{col} = d_{col} - d_B \tag{9c}$$

$$d'_{cen} = d_{cen} - d_B \tag{9d}$$

where j is the zone number. If j>JZ or j<1, then the particle is considered lost or has escaped from the system, and its history is ended. Otherwise the calculation and selection is repeated until the particle reaches the census time.

iii) If d_{col} is the smallest, the particle is first advanced to collision point and

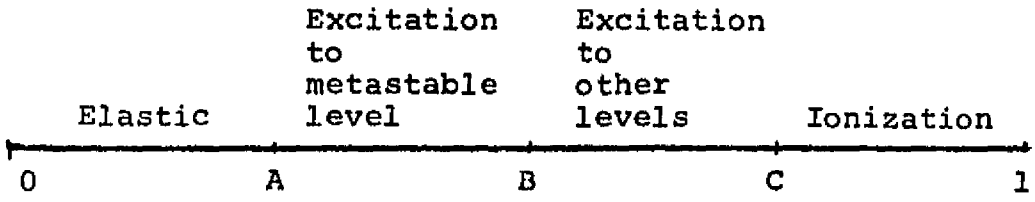
$$z' = z + u \cdot d_{col}$$

$$d'_{cen} = d_{cen} - d_{col} \tag{10}$$

Then the type of collision is specified. For specifying the collisional processes, we divide the whole energy range into two regions. Region I: energy range 0 to ED1 eV. Region II; energy range ED1 eV to E_o eV. Ed1 and E_o are specified by input parameters and are referred to as the energy dividing line and source particle energy, respectively.

If the particle's energy is in the range of Region I, three kinds of collisions are possible: elastic collisions, electron-ion recombination, and superelastic collisions (where electrons hitting an excited atom pick-up the excitation energy). If the particle's energy falls in Region II, it can experience four kinds of collisions; namely, elastic collisions and three types of inelastic collisions--excitation to metastable level, excitation to other levels, and ionization. For ionization, the secondary electron so produced shares energy with the primary electron, and the energy of primary electron before collision less the ionization potential is shared between the two electrons. The partition of the shared energy could be determined according to Goodrich's experimental distribution curve (31). But based upon Thomas' (1) calculation experience, the way in which the energy is partitioned does not have a strong influence on any of the calculated mean properties. Thus, for computational simplicity, as Thomas did in all his calculations, we choose the partition ratio 9:1. The secondary electrons produced via ionization are considered as new particles introduced into the system. Thus they are followed from their point of birth just like other particles. This is equivalent to the generation method for processing branched trees of samples.

The type of collision is determined according to Figure 3 using a random number (uniformly distributed



$A = P_{el}$

$B = P_{el} + P_{met}$

$C = P_{el} + P_{met} + P_{exc}$

$P_{el} + P_{met} + P_{exc} + P_{ion} = 1$

Figure 3. The probability model for determining the types of collision.

between 0 and 1) to play the law of chances.

b. With Electric Field Present

The coordinate axis z is defined as antiparallel to the electric field E_f. In the presence of electric field E_f, the state of motion of an electron is specified by its speed V and the direction cosine cos θ, where θ is the angle made by the line of flight with the z-axis at the time t. If an electron is in a state [E(V), u(cosθ), z, t], it will assume (after a flight along the orbit by Δt) the new state [E'(V'), u'(cosθ'), z+Δz, t+Δt]. These parameters are connected by the following relations:

$$V' \cdot \sin\theta' = V \cdot \sin\theta \tag{11}$$

$$mV'\cos\theta' = mV\cos\theta + eE_f \cdot \Delta t \tag{12}$$

$$1/2\ mV'^2 = 1/2\ mV^2 + eE_f \cdot \Delta z \tag{13}$$

$$E' = E + eE_f \cdot \Delta z \tag{14}$$

where m and -e are the mass and the electric charge of the electron, respectively. Here Δz is the projection of the path length L on the z axis. It can be obtained as follows;

$$\Delta z = \int_0^{\Delta z} V_z dt = \int_0^{\Delta t} (V \cdot u + \frac{E_f}{m} \Delta t)\, dt$$

$$= V \cdot u \cdot \Delta t + 1/2 \cdot \frac{E_f}{m} \cdot \Delta t^2 \tag{15}$$

In carrying out the geometry routine, it is suitable to use time directly as the clock. The determination of the distance to collision d_{col}(DCOL) and the distance to census

d_{cen}(DCEN) is accomplished as follows:

$$DCOL = \lambda \cdot |\ln r| \tag{16}$$

$$= 2/(1/\lambda_{(E)} + 1/\lambda_{(E')}) \tag{17}$$

$$DCEN = \sqrt{V_y^2 + V_z^2} \cdot \Delta t \tag{18}$$

$$V_y = V \cdot \sin\theta \tag{19}$$

$$V_z = V \cdot \cos\theta + \frac{E_f}{2m} \Delta t \tag{20}$$

where E: the initial energy of the particle

E': the energy of the particle after flight time Δt.

The next step is to determine which of the distances is smaller. If DCEN<DCOL, then the particle is advanced to the new state

$$E' = E + E_f \cdot \Delta z$$

$$z' = z + \Delta z$$

$$u' = \sqrt{E/E'} \cdot u + eE_f \cdot \Delta t \tag{21}$$

If $E' \leq 0$ or $z' > D$, $z' < 0$ then the particle is considered to have escaped from the system. If DCEN $\geq$ DCOL, the particle is advanced to collision point.

$$\Delta t' = \Delta t - DCOL \cdot \Delta t/DCEN$$

$$DCEN = DCEN - DCOL$$

$$z' = z + \Delta z \cdot DCOL/DCEN$$

$$E' = E + \Delta z \cdot E_f \cdot DCOL/DCEN \tag{22}$$

then the specific collision type must be determined.

c. Types of Collisions and Cross Sections

For the elastic collisions, the particle scatters off with angle of deflection θ and energy loss ΔE, the following formula is used (36).

$$\Delta E = \frac{2m}{M} (E-E_n) (1-\cos\theta_s) \qquad (23)$$

where θ_s: scattering angle (angle of deflection)

E: particle energy

E_n: target gas atom energy

m: electron mass

M: gas atom mass

E_n is selected according to a Maxwellian distribution, and the ΔE so formed may have either positive or negative values, indicating the loss or gain of energy respectively. For calculational simplicity, instead of selecting the deflection from the elaborate distributions, we use $\pi/4$ or $\pi/2$ deflection approximations, namely $\theta_s = \pi/4$ or $\pi/2$, then $\cos\theta_s = 0$ or 0.70711. Another reasonable alternative is to assume that the direction after collision is equally probably in all directions (isotropic scattering). We select this direction at random with uniform distribution, and based on Eq. 23 calculate $\cos\theta_s$ and ΔE, respectively, where

$$\cos\theta = 2r_i - 1 \qquad (24)$$

$$\theta_s = \theta - \theta_i$$

where θ_i: incident angle

θ: outgoing angle after collision

For the three kinds of inelastic collision cross sections we use approximate formula based on experimental data as correlated by Itoh and Musha (2). The formulas are expressed in the form:

$$\sum(E) = \frac{B(E-E_i)}{A+(E-E_i)^2} + C \qquad (25)$$

where the constants A, B, C, and E_i depend on the kind of collision and scattering angle. For the elastic cross section in helium, we use a single approximate formula for whole effective energy range based on Heylen and Lewis's paper (10), namely:

$$\sum_{el}(E) = 26\exp(-0.04E) \ (cm^{-1}) \qquad (26)$$

For the recombination cross section for helium (35), we assume the simple form:

$$\sum_{RCB}(E) = \mu(E)\cdot N^+$$
$$\mu(E) = 10^{-9}/V(E) \qquad (27)$$

where N^+: ion density

V: electron velocity

the superelastic cross section is

$$\sum_{Sel}(E) = \frac{0.109(19.8-E)}{0.25 + (E-19.8)^2} \cdot \frac{N^+}{3\cdot D_n} \qquad (28)$$

where N^+: positive ion density

D_n: gas atom density

E: electron's energy

3. The Prediction and Correction of Dominant System Parameters

There exists a single parameter, the ion density distribution $N^+(E,z)$, which completely characterizes the nonlinearity of the system. This simplifies the problem and suggests the following predict-and-correct scheme. We assume that the average mean free path (or Δt) and the ion density distribution $N^+(E,z)$ in the present time cycle also applies to the next time cycle, In other words, in each time cycle we use the predicted values of Δt and $N^+(E,z)$ from previous time cycle. At the end of present time cycle, we use the up to date information to renormalize and correct the value to be used for the next time cycle. As steady state is reached, the predicted and corrected values of the parameters converge. From the continuity equation

$$\nabla \cdot J = S - L \tag{29}$$

where J: the particle current flow in or out of the system

S: the source

L: the sum of losses

To start with, we assume that $\nabla \cdot J = 0$, so that

$$S = L = \alpha(E)\ N^+(E,z)\ N^-(E,z) \tag{30}$$

Here $N^-(E,z)$ is the electron number density, the key quantity we are calculating. To initialize the run we have to predict $N^+(E,z)$. It is reasonable to initially assume this distribution is uniform in z, and a maxwellian distribution in E, i.e., $N^+(E,z) = N_0 M(E)$. We also assume that $N^-(E,z)$ is the same as $N^+(E,z)$. Then we obtain

$$\begin{aligned} S = L &= \alpha(E)\ N^+(E,z)\ N^-(E,z) \\ &= \alpha(E)\ [N^+(E,z)]^2 \end{aligned} \tag{31}$$

$$\text{or} \quad S_0\ \delta(E-E_0) = \alpha(E)\ N_0^2\ M^2(E)$$

Integration of both sides gives

$$S_0 = N_0^2 \int_0^{E_0} \alpha(E)\ M^2(E)\ dE$$

$$\therefore\ N_0 = [S_0 / \int_0^{E_0} \alpha(E)\ M^2(E)\ dE]^{1/2} \tag{32}$$

This gives a starting value for the calculations. In later cycles the renormalization process involves integration of Eq. 29, with $\nabla \cdot J \neq 0$, i.e.,

$$\iint \frac{dJ(E,z)}{dz}\, dzdE = \iint S(E,z)dzdE$$

$$- \iint \alpha(E)\ \ N^+(E,z)\ N^-(E,z)dzdE - J_{esc} + J_s$$

which reduces to

$$J(Z=d) - J(z=0) = S_0\ d - \iint \alpha(E)\ N^+(E,z)\ N^-(E,z)dzdE$$

$$- J_{esc} + J_s$$

$$N_i^2 = \frac{S_0 d + J_s - [J(z=d) - J(z=0)] - J_{esc}}{\iint \alpha(E)\, M(E)\, \bar{n}(E,z)\, dz dE} \qquad (33)$$

where i refers to ith time cycle

J_{esc}: current due to escaped electrons

J_s: current due to secondary electrons produced by ionizations

$M(E)$: Maxwellian distribution (normalized)

$\bar{n}(E)$: electron number density distribution (normalized)

S_0: source rate

N_i: ion density distribution (normalized)

The value of N_i replaces N_0 in equation (33) at later time cycles. The value of N_i (N^+) enters the formula for the recombination cross section and superelastic cross section, thus it directly affects the rates of recombination and superelastic collision. Indirectly, it affects the balance of the system, or equivalently the rate of convergence of the Monte Carlo calculation. An adaptive correction procedure is used here. It is based on the value of the "balance factor" BF, defined as the ratio of the particles introduced into the system by source and ionization processes to the particles removed from the system due to recombination and leakage. We adopt a over-relaxation type of correction technique, the over-correction is used to speed up the over-all convergence rate. As defined before, if BF $>$ 1, the system has more particles introduced

into than eliminated, but if BF < 1, the opposite is true. The over-correction factor is proportional to $(BF)^k$, i.e., to the kth power of the balance factor. The value of k initially chosen was k=2, but an optimum value can be obtained by trial and error.

CHAPTER III

VARIANCE REDUCTION TECHNIQUES

A. General Background

The Monte Carlo method (or method of statistical trials) consists of solving various problems of computational mathematics by the construction of some random process for each such problem, with the parameters of the process set equal to the required quantities of the problem. These quantities are then approximated by observations of the random process and the computation of its statistical characteristics, which are approximately equal to required parameters [from Schreider (15)]. Every Monte Carlo computation that leads to quantitative results may be regarded as estimating the value of a multiple integral. Suppose we have M random numbers in the computation system. The results will be a vector valued function $R(\xi_1, \xi_2, \ldots \xi_M)$ involving the sequence of random numbers $\xi_1, \xi_2, \ldots \xi_M$. This is an unbiased estimator of $\int_0^1 \ldots \int_0^1 R(x_1, x_2, \ldots x_M)$ $dx_1 \ldots dx_M$. For the sake of simplicity, we take the one-dimensional integral as a standard example. Let

$$\theta = \int_0^1 f(x)dx \tag{34}$$

where $f \in L^2(0,1)$ or $\int_0^1 [f(x)]^2 dx$ exists.
Define the relative efficiency of two Monte Carlo methods as follows. Let the methods call for n_1 and n_2 units of computing time, respectively, and let the resulting

estimates of θ have variances σ_1^2 and σ_2^2. Then the efficiency of method 2 relative to method 1 is defined as

$$\frac{n_1 \cdot \sigma_1^2}{n_2 \cdot \sigma_2^2} \tag{35}$$

Equation 35 is the product of two terms, the variance ratio σ_1^2/σ_2^2 and the labor ratio n_1/n_2. If $\eta_1, \eta_2, \ldots \eta_N$ are independent random numbers with uniform distribution between 0, 1 or $\eta_i \in (0,1)$; then the quantities $f_i=f(\eta_i)$ are the independent random variates with expectation θ. Therefore

$$\bar{f} = \frac{1}{N} \sum_{i=1}^{N} f_i \tag{36}$$

is an unbiased estimator of θ, and its variance is

$$\frac{1}{n} \int_0^1 (f(x)-\theta)^2 dx = \frac{\sigma^2}{n} \tag{37}$$

Accordingly the quantity $\bar{f}$, which has been determined by observation of the random process, is approximately equal to the required quantity θ (or in other words, it is an unbiased estimator of θ) with a probability which can be made as close as required to unity if a large number of trials is practical. We refer to the estimator $\bar{f}$ as the crude Monte Carlo estimator of θ (commonly referred to as Monte Carlo estimation).

Let us introduce another even less efficient method, namely hit or miss Monte Carlo. Suppose that $0 \leq f(x) \leq 1$

when $0 \leq x \leq 1$. We may draw the curve $y = f(x)$ in the unit square $0 \leq x, y \leq 1$, and θ is the proportion of the area of the square beneath the curve. Stating this formally, we write

$$f(x) = \int_0^1 g(x,y)dy \tag{38}$$

where $g(x,y) = 0$ if $f(x) < y$

$= 1$ if $f(x) \geq y$

Then

$$\theta = \int_0^1 \int_0^1 g(x,y)\, dxdy \tag{39}$$

The estimator

$$\bar{g} = \frac{1}{n} \sum_{i=1}^{n} g(\xi_{2i-1}, \xi_{2i}) = \frac{n^*}{n} \tag{40}$$

n^* is the number of occasions where $f(\xi_{2i-1}) \geq \xi_{2i}$
In other words, we take n trials at random in the unit square, and count the proportion of them which lie below the curve $y = f(x)$.

Historically, a hit or miss method was first propounded in the explanation of Monte Carlo techniques. It is the easiest to understand but one of the least efficient techniques. The rate of convergence and the variance (statistical error) is in general proportional to $1/\sqrt{N}$ where N is the number of samples. It is obviously impractical to gain a significant improvement of accuracy by merely increasing N. Therefore more efficient techniques for reducing the variance and/or increasing the efficiency

are very important. Monte Carlo experimentalists need wide experience and background in both the mathematics and the physics of the problem, and they have to exercise considerable ingenuity in distorting and modifying problems in the pursuit of variance reduction techniques. Statistical and inferential procedures are also important in order to extract the most reliable conclusions from the observational data.

The basic requirement of the Monte Carlo calculations is to establish how random numbers may be used to sample a function that describes, in a probabilistic fashion, a physical event. The procedure by which this is done follows from the fundamental principle of Monte Carlo (16), which may be stated as follows. If $p(x)dx$ is the probability of x lying betweer x and x+dx, with $a \leq x \leq b$

$$\int_a^b p(x)dx = 1 \tag{41}$$

Then $$r = P(x) = \int_a^x p(y)dy \tag{42}$$

determines x uniquely as a function of the random number r which is uniformly distributed on range (0, 1). The quantities $p(x)$, $P(x)$ are the probability density and the probability distribution functions, respectively.

With the development of large computers, the use of the Monte Carlo method in a wide range of problems has

increased rapidly. In particular, the method is applied to varicus physical problems such as plasmas (15), radiative transfer problems (21), neutron transport and nuclear reacter physics (29), etc.

The random walk type simulation can be briefly described in the following manner. One first introduces the basic physics of the problem into the computer in a probabilistic fashion. A system of coordinates and boundaries are defined and then, as a computer experiment, particles are released from the source. These particles are traced as they diffuse through a prescribed medium following the probabilistic interaction laws. The particles are followed until they escape from the medium or are absorbed or are converted to the thermal field.

The parameters pertinent to the evaluation of the desired quantities are recorded. One continues processing additional particles until adequate statistical estimates of the quantities of interest have been obtained.

When doing a Monte Carlo problem one focusses attention on three main topics. They are:

1. The development of the analogy for the probability processes (simulation).
2. The generation of sample values of the random variables on a given computing machine.
3. The design and use of variance reduction techniques.

The variance reduction methods are often strongly dependent on the probabilistic model, thus the greatest gains are often made by exploiting specific details of the problem, rather than by routine application of general principles.

B. Importance Sampling

The general idea of importance sampling is to draw samples from a distribution other than the one suggested by the problem. Then an appropriate weighting factor is introduced to correct the biasing caused by changing the original distribution. If done correctly the final results are essentially unbiased. The object is to concentrate the distribution of sample points in the parts of the region that are most "important" instead of spreading them out evenly. Thus, the probability of sampling from an interesting region is increased; the probability of sampling from unimportant or less interesting region is correspondingly decreased. Stating this formally, we have

$$\theta = \int_0^1 f(x)dx = \int_0^1 \frac{f(x)}{g(x)}\, g(x)dx = \int_0^1 \frac{f(x)}{g(x)}\, dG(x) \qquad (43)$$

for any functions g and G satisfying

$$G(x) = \int_0^x g(y)dy \qquad (44)$$

where g must be a positive valued function such that

$$G(1) = \int_0^1 g(y)dy \tag{45}$$

if η is a random number sampled from the distribution G, then $f(\eta)/g(\eta)$ has the expectation θ and variance

$$\sigma^2_{f/g} = \int_0^1 \left(\frac{f(x)}{g(x)} - \theta\right)^2 dG(x) \tag{46}$$

We notice that if $g(x) = cf(x)$, or if g is proportional to f and if $c = \frac{1}{\theta}$ then $\sigma^2_{f/g} = 0$. This perfect situation does not exist, since we do not know θ. We always obtain an unbiased estimate for positive function g. Our object is to select g to reduce the standard error of our estimate.

Because the estimate is the average of observed values of f/g, we choose g such that f/g is as constant as possible in order to achieve a small sampling variance. While we intend g to mimic f, we generally restrict our choice of g to functions simple enough to allow analytic integration.

For the present plasma problem, the importance sampling technique is developed as follows: during the particle's (electron's) random walk, it suffers a large number of collisions with gas atoms, and these collisions may be elastic or inelastic or of other kinds. The probabilities of their occurrence depend on their cross section Σ_i. We have

$$\Sigma_t = \Sigma_1 + \Sigma_2 + \ldots \Sigma_M \tag{47}$$

where Σ_i: the cross section for ith process

Σ_t: total cross section or the summations of the cross sections of all the possible processes that the particle may experience.

Dividing both sides of equation 47 by Σ_t, we obtain:

$$1 = p_1 + p_2 + \cdots p_M \tag{48}$$

where $p_i = \Sigma_i / \Sigma_t$ the probability of ith process occurring as the particle experiences a collision. If we introduce weights w_i into equation 47 we obtain

$$\Sigma_t' = w_1 \cdot \Sigma_1 + w_2 \cdot \Sigma_2 + \cdots w_M \cdot \Sigma_M \tag{49}$$

Similarly we have

$$1 = p_1' + p_2' + \cdots p_M' \tag{50}$$

After the application of the weights, we put more weights on the interesting or more important events. Then their artificial probabilities are greater, therefore we can get better estimates of the important parameters. Coincidently, such a linear weighting scheme can only be applied to the elastic collision process in our system. The other collision processes involve appreciable changes in the particle's energy (catastrophic collision). Also cross sections are energy dependent, these nonlinear effects prevent the application of above mentioned weighting scheme to inelastic collision processes. By looking at the cross section values, in the low energy region, the elastic cross section is by

far the largest by order of magnitude. Others have shown that the elastic process does not strongly influence the estimated parameters that we are interested in (1, 2). As an electron makes an elastic collision, it loses only a small amount of energy while making a small angle deflection. In fact, most other workers simply neglect elastic loses if an electric field is present (1, 2, 10). In our case, with E_f=0, elastic effects must be included. In the low energy region ($\approx$ 0 eV to 20 eV), the energy of a majority of the particles lies below 1 eV and over 90% collisions are elastic collisions, and only a few collisions are inelastic.

The collision process subroutines are a time-consuming part of the simulation calculations. If we can supress the elastic processes by a factor of k_{el} the chances of all other processes are increased by approximately the same factor. Therefore we convert more computing time to processes which contribute more strongly to the quantities of interest. The following formulation provides the theoretical support for this. In lieu of equations 47-50, the p_i are the original probabilities (undistorted probabilities) while the p_i' are the weighted probabilities. Then the correction factors are intrcduced as follows; at energy E, we have

$$C_1 = p_1/p_1', \; C_2 = p_2/p_2' \; . \; . \; . \; C_M = p_M/p_M' \qquad (51)$$

$$\text{or} \quad C_i = p_i/p_i' = \frac{\Sigma_i \cdot \Sigma_t'}{\Sigma_t \cdot \Sigma_i w_i} = \frac{\Sigma_t'}{\Sigma_t} \frac{1}{w_i} \tag{51a}$$

If we let $w_1 = \frac{1}{k_{el}}$, $w_2 = w_3 = \ldots w_M = 1$

then from (51) we have

$$C_1 = \frac{\Sigma_1 \cdot \Sigma_t'}{\Sigma_t \Sigma_1 \cdot w_1} = \frac{\Sigma_t'}{\Sigma_t} \frac{1}{w_1}, \quad C_i = \frac{\Sigma_t'}{\Sigma_t} = 1, \quad i \neq 1 \tag{52}$$

let $\Sigma_1 = \Sigma_{el}$ namely we introduce weight on elastic cross section only. Since $\Sigma_{el}(E) >> \Sigma_i(E)$, $i > 1$, $\forall_i$ then

$$\frac{\Sigma_t'(E)}{\Sigma_t(E)} \simeq \frac{\Sigma_{el}'(E)}{\Sigma_{el}(E)} \qquad w_1 = \frac{1}{k_{el}} \tag{53}$$

$$\text{and} \quad C_1 \simeq w_1 \frac{1}{w_1} = 1, \; C_i \simeq \frac{1}{k_{el}} \quad (i \neq 1) \tag{53a}$$

Equation 53 holds for low energy region, which is approximately independent of energy.

C. Russian Roulette and Splitting

In general, the sampling is done such that we examine the sample and classify it as being in some sense "interesting" or "uninteresting". We are willing to spend more than average amount of work on the interesting samples and on the contrary, we want to spend less effort on the uninteresting ones. This can be done by splitting the interesting samples into independent branches, thus resulting in more of them, and by killing off some of the uninteresting ones. The first process is splitting and the second Russian Roulette.

The "killing off" is done by a supplementary game of chance. If this game is lost, the sample is killed; if it is won, the sample is counted with an extra weight to make up for the fact that some other samples have been "killed". The game has a certain similarity to the Russian game of chance played with revolvers - whence the name.

For the present plasma system we divide the whole energy spectrum into two regions, the low energy region I (0 eV to ED1 eV) and the high energy region II (ED1 to E_0). Here E_0 is energy of source particles, which is the highest energy the particle can have. ED1 is the energy region dividing line, which can be chosen as any value between 5 eV and 100 eV depending on the initial guessed energy distribution. In region I, the number density is high, and the elastic process is the dominant process. As we mentioned earlier, the elastic process does not contribute much, so Region I is classified as uninteresting. Each particle in this region (particles with energy less than ED1) is assigned a weight k1($k1 \geq 1$). In other words, each particle plays the role of k1 particles.

In region II, classified as interesting region, each particle carries a weight w_2 ($w_2 = {}^1/k_2$, $w_2 \leq 1$). Namely, each particle in this region is split into k_2 particles with weight w_2.

As particles cross over to other regions, care must be taken to account for killing off and splitting effects.

For instance, a particle in region I may acquire enough energy, either from electric field or by superelastic collision with the excited atoms, to cross over to region II. Then we should generate k_3 particles ($k_3 = k_1 \cdot k_2$) in region II with the same energy.

Particles from the high-energy region may lose part of their energy by inelastic collision such that they cross over to the low-energy region. Then killing off should be effective. For every k_3 such particles that cross over from region II to region I, only one particle survives. The values of k_1 and k_2 can be chosen according to the total number of particles in the system to be processed, the source rate, and related accuracy requirements. They a : input parameters.

D. Initial Guess

Monte Carlo methods comprise that branch of experimental mathematics which is concerned with experiments on random numbers. The user, like the experimental physicist needs theory and knowledge to give structure and purpose to his experiments, and as experimental work provides growing insight into the nature of a problem and suggest appropriate theory. Good Monte Carlo practice should keep this relationship as a general maxim.

The basic procedure of the Monte Carlo method is the manipulation of random numbers, but these numbers should not

be employed prodigally. Each random number is a potential source of added uncertainty in the final result. Thus it will usually pay to scrutinize each part of a Monte Carlo experiment to see whether that part can not be replaced by an exact theoretical analysis contributing no uncertainty. In other words, exact analysis should replace Monte Carlo sampling wherever possible. Sometimes reliable intuition would aid in increasing the efficiency of Monte Carlo calculations. For the plasma system we are simulating, the straight-forward approach would be to follow N source particles on their life time history, until they lose energy sufficient by collisions to reach a thermal equilibrium. Over 50 inelastic collisions generally occur, and the probability (inelastic cross sections) for inelastic p.ocesses is low. Consequently, the particles travel a relatively large distance which consumes a good amount of time. Thus a large number of time cycles would be required before the system reaches steady state. To avoid this, we use our intuition, to start off with a guessed initial distribution. The electron energy distribution f(E) (or electron number density distribution) is a function of the electron's energy E. Initial particle data is generated using this distribution, and this can save considerable irrelevant computations. This idea is simple, yet very useful and effective. Without an initial distribution, the problem would require hundreds of time cycles to

converge. However, with a reasonable guess, good results have been obtained in as little as tens of time cycles. Of course, the better the initial guess, the faster the convergence.

E. Antithetic Variates

When we estimate an unknown parameter θ by means of an estimator t, we may seek another estimator t' having the same (unknown) expectation θ as t but a strong negative correlation with t. Then 1/2(t + t') will be an unbiased estimator of θ, and its variance is

$$\mathrm{Var}[\tfrac{1}{2}(t+t')] = \tfrac{1}{4}\mathrm{var}(t) + \tfrac{1}{4}\mathrm{var}(t') + \tfrac{1}{2}\mathrm{cov}(t,t') \qquad (54)$$

where cov(t,t') is negative, and

$$\mathrm{var}(t) = \mathcal{E}[(t-u)^2]$$

$$\mathrm{cov}(t,t') = \mathcal{E}[(t-u)(t'-u')]$$

u,u' are the mean of t,t' respectively, and $\mathcal{E}(x)$ is the expectation of the random variable x.

It is possible to make var(t+t') smaller than var(t) by suitably selection t'. For example, 1-r is uniformly distributed whenever r is, and if f(r) is the unbiased estimator of θ, so is f(1-r). When f is a monotone function, f(r) and f(1-r) will be negatively correlated. Thus we could take

$$\tfrac{1}{2}(t+t') = \tfrac{1}{2}\,t(r) + \tfrac{1}{2}\,f(1-r) \qquad (55)$$

as an estimator of θ.

The name and idea of antithetic variates was first introduced by Hammersley and Morton (13, 14) during attempts to improve Buffon's needle experiment for estimating the value of π (3.1415926).

The main idea of antithetic variates is based on the underlining theorem [from Hammersley and Morton (12)] which can be stated informally as follows: "Whenever we have an estimator consisting of a sum of random variables, it is possible to arrange for a strict functional dependence between them, such that the estimator remains unbiased, while its variance comes arbitrarily close to the smallest that can be attained with these variables". As the name implies, antithetic variates are the set of estimators which mutually compensate each other's variations. Essentially, we rearrange the random variables by permuting finite subintervals in order to make the sum of the rearranged functions as nearly constant as possible. From the practical viewpoint, the mathematical conditions imposed by antithetic variates to the Monte Carlo calculation are quite loose and flexible. It is relatively easy to find a negatively correlated unbiased estimator to produce an efficient variance reduction scheme.

A system of antithetic variates is obtained by simple transformations based on a stratification of the interval (12). The transformations are constructed in such a way

that the efficiency of the method will increase as a higher power of M (the number of antithetic variates taken). The transformations considered are linear combinations of the values of the function f at a number of points. Thus a correlated stratified sampling technique is developed, and its efficiency generally exceeds that of crude sampling by a factor M^3.

The following useful transformations are obtained in this manner:

$$t_1 = \frac{1}{n}\sum_{i=1}^{n} \{\alpha f(\alpha r_i) + (1-\alpha)\, f[\alpha+(1-\alpha)r_i]\} \qquad (56)$$

$$= \frac{1}{n}\sum_{i=1}^{n} G_\alpha f(r_i)$$

$$t_2 = \frac{1}{n}\sum_{i=1}^{n} \{\alpha f(\alpha r_i) + (1-\alpha)\, f[1-(1-\alpha)r_i]\} \qquad (57$$

$$= \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_\alpha f(r_i)$$

The value of α is between 0 and 1, the variance $\mathrm{var}(t_j)$ has a minimum at some value of α. An adequate rule of thumb is to choose α by finding the root of

$$f(\alpha) = (1-\alpha)\, f(1) + \alpha f(0) \qquad (58)$$

Another useful transformation is

$$t_3 = \frac{1}{m}\sum_{j=0}^{m} f\left(\frac{r+j}{m}\right) = U_m f(r) \qquad (59)$$

In general, the variance of crude Monte Carlo is $var(t) = 0(m^{-1})$ while the variance of t_3 is $var(t_3) = 0(m^{-2M})$, $(M > 1)$. The simplest form of antithetic transformations have proved quite successful in neutron transport calculations.

The general notion and a simple form of antithetic variates are applied in our present calculations. High efficiency gain can be obtained easily if it is used sparingly and at judicious points in the computation. We have used the random number $r_i (r_i \in (0,1))$ most of the time for playing the law of chances, and $1-r_i$ is another random number of the same kind which is the image of r_i. The two are used together in good many places where repeated use of r_i is required. This serves the purpose of two antithetic variates and cuts the labor in half (same random number r_i being used twice). Equations 55 and 59 have been used in the initial data preparation program and source routine where initial particles states (E_i, u_i, z_i, wt_i), E_i are generated according to input initial distribution. For instance:

$$
\begin{aligned}
&u_i = 2r_i - 1, \quad u_{i+1} = 2(1-r_i) - 1 \\
&z_i = r_i \cdot d, \quad z_{i+1} = (1-r_i) \cdot d \\
&E_i = E_k + r_i \cdot \Delta E_k, \quad E_{i+1} = E_k + (1-r_i) \cdot \Delta E_k \qquad (60) \\
&E_i = E_0 \text{ for source particles.}
\end{aligned}
$$

Here k is an energy index, or kth energy interval. Wherever r_i appears, r_i and $1-r_i$ are used interchangeably.

Thus we are generating source particles from diametrically opposite directions, or from symmetric locations.

F. Double History (Double Processing) Technique

In the particle tracing routine, using the general idea of antithetic variate technique, we have adopted a "so-called" double history processing technique which is described as follows;

a) In the course of tracing the particle's history, in each time cycle, we use the transformed random number $\lambda \cdot |\ln r|$ to decide the distance to collision $d_{col\ 1}$ and go on until the particle reaches census time. However, we process the same particle twice under the identical initial condition except that we use $\lambda \cdot |\ln(1-r)|$ as the distance to collision $d_{col\ 2}$. Both sets of history are ta lied but only one set of particle data is'stored. This way each particle is processed twice independently but correlated in such a way that they compensate each other's variation (variance).

b) Another alternative for double history processing is, to play a dual direction random walk instead of dual collision distances. We process the same particle twice in such a way it looks like there are two identical particles emerging into the system in two diametrically opposite directions. Again both histories are tallied, but only the original particle's data is stored.

The two versions of the double history processing technique increase the labor (computation time) by a factor of 1.2 to 1.5. The efficiency gain is difficult to specify. At least it is certain that the gain is more than double the total number of the particles in the system.

The purpose of dual processing is to compensate each particle's variation and minimize the variance (statistical errors). One set of particle data in dual processing is thrown away, while the other set is stored just like normal processing. Which one of the two resulting sets is saved is completely random.

CHAPTER IV

MONTE CARLO SIMULATION CODE

A. Introduction

This chapter presents the complete Monte Carlo simulation program, including an input, output, main flow chart, and the background theory and techniques incorporated. This program implements the condensed history approach that was described in Chapter II together with various variance reduction techniques described in Chapter III.

The basic Monte Carlo calculation can be described most easily by reference to the simplified flow chart shown in Figure 4. By means of this flow chart we shall follow the particle tracing process, making relatively brief comments on the various steps implied by each box, on each routine, and on the main program variables. The program is written in FORTRAN IV and is designed to be run on IBM 360/75 under the HASP-MVT system.

B. Calculational Procedure and Program Description

To obtain an overall view of the calculations, we note that there are three basic steps involved in the simulation process. The first is the initialization of all parameter values, i.e., prepare the particle data and predict the unknown dominate parameter values. The second is the main loop of particle tracing which generates census information. The third uses the census information and takes tallies to estimate the histograms (electron energy or number density

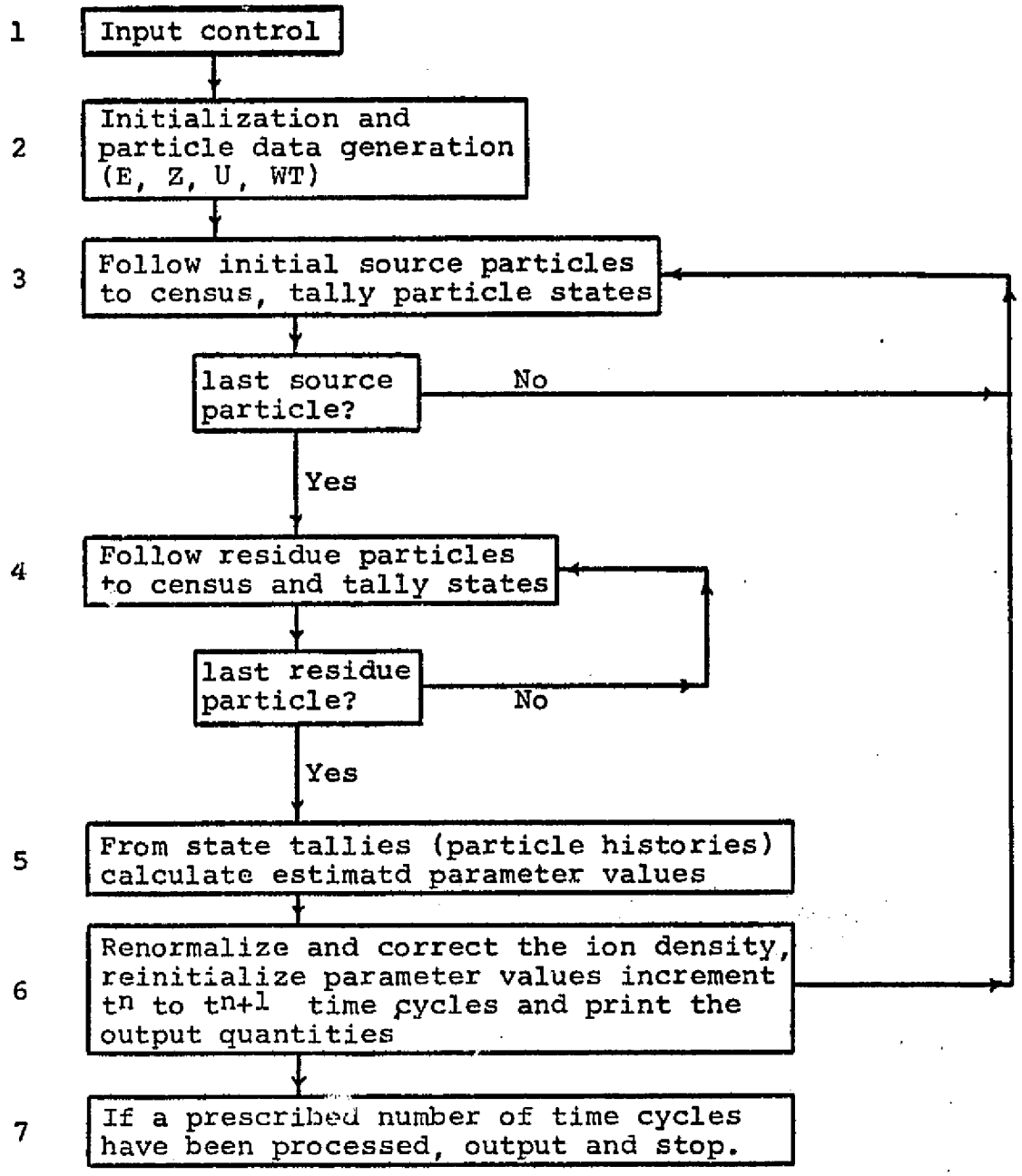


Figure 4. Flow chart of the Monte Carlo simulation code.

frequency distribution vs. energy), and parameter values. Finally the dominate parameter va ues are corrected and the program returns to the first step.

Let us follow the logic flow diagram (Figure 4) in the order of the number marked on the left hand side of the boxes, and describe their individual functions.

1. Input Control: The main program is usually compiled and the loaded version is then stored on disk storage, ready for execution. In order to offer a wide flexibility, the following input parameters are used to control the program runs.

   a) NDISK: program start control. The program can start from the beginning (cycle 0), or it can start from the stopping point of the previous run. This feature enables long runs to be broken up into several short runs. It also offers debugging flexibility and economy of computer time. While a long run may run up to 150 time cycles, we can always break it up into multiple numbers of fives and tens cycles. Thus we can adjust other parameter values on a cut and try basis without wasting much computing time. The input parameter for this control purpose is NDISK: input 0 for storing from very beginning; input 1 for starting from the latest tenth cycle of last

run; input 2 for starting from the latest fifth cycle.

b) K1, K2, and ED1: Regional weights and dividing energy. The Russian Roulette and splitting variance reduction techniques use three input parameters to control the amount of killing off and/or splitting of particles. This also determines where to divide energy regions for the Russian Roulette and splitting, assuming that the option to use Roulette is selected. The input parameters are K1: the weight of particles in region I ($K1 \geq 1$), K2: the inverse of the weight w2 of particles in region II where $w2 = 1/K2$ ($K2 \geq 1$), ED1: the energy dividing line, namely (0 to ED1) is region I, (ED1 to E_0) region II where the energies are in electron volts.

c) NTC: The total number of time cycles to be run.

DTC: The input parameter to control the length of census time Δt of the time cycles. In terms of distance travelled by the particles, Δt is equal to a fraction or multiple of one mean free path , corresponding to average energy of particles in the system.

d) NTP: The total number of particles that actually exist in the simulation system.

NTW: The artificial* total number of particles in the system during Russian Roulette and splitting games.

NS: The number of source particles injected into the system during each time cycle.

e) KELC: The input parameter that controls the importance sampling option of the program, i.e., the multiplying factor used to suppress the elastic cross section (reduces the probability of elastic process by a factor KELC).

f) EDP: The input value of the ratio of the electric field strength to the pressure.

PDPO: The input value for the reduced pressure of the system (p/p_0 where p_0 = 1 torr).

g) NIP: The input parameter used to control double processing: NIP = 2 for double processing, NIP = 1 for single processing.

IUSN: IUSN = 1 for dual-collision type double processing

IUSN = -1 for dual-direction type double processing

---

*The total number modified by scaling and weighting factors.

h) D: The input value of the width of the slab (diameter of the cylinder) which defines the external boundaries of the medium.

JZ: The number of zones the slab is divided into, or the number of internal zones.

i) NCUM: Inp t control to give cumulative results (time average) over the time cycles or a cumulative average over NCUM time cycles.

NAVP: Input constant to control the power of Δt, (or ΔD) namely Δt_i**NAVP is the weight of the cycle to be used for cumulative results. These two parameters are used to implement antithetic variates techniques along the time axis.

j) EO: The energy of source particles.

SR: The rate of the source particles entering the system.

k) FID: An array that contains the initial guess distribution (number density vs. energy).

l) NUSC: An input parameter used to control the upscattering elastic collision process; NUSC = 1: with upsacttering, NUSC = 0: without upscattering.

2. Data preparation: The initialization is entered only once during the calculation, where the time cycle is 0 (or at time t_0). After the input phase, all the pertinent

parameters are set to their proper permanent or temporary values. Before entering the next phase to trace the particle histories, several dominate parameter values must be predicted. The data preparation routine is called in to generate particle data (E_i: energy, u_i: direction of travel, Z_i: position, Wt_i: weight) based on the guess for the initial distribution (generate machine particles). Then the average energy of the system is calculated and is used to predict the census time (Δt or equivalently DCEN) as well as the number of ions(N^+).

3. Particle tracing: With the point of origin, direction of propagation, and energy (velocity) known for the initial source particles, the next phase follows the particles through a series of stochastic interactions until the census time is reached, and the particle data and parameters values are tallied and updated. During the course of tracing the particle histories, all the relevant events are recorded and later on can be interrogated to provide the output quantities of interest. This section of the program is the central part of the main program. It contains the main loop for tracing all the particles in the system in a self consistent way. Most of the variance reduction techniques are incorporated here, including double processing, Russian Roulette, and splitting, as well as the various versions of

importance sampling techniques.

4. Secondary particle tracing: The residue particles, i.e., particles generated by ionization collision process (the secondary electrons so produced) are treated in all respects like the regular source particles, except that their starting positions and time coordinates as well as directions are those associated with the particles that generated them. These residue particles may go further and produce other residue particles (secondary electrons) if their energy is high enough, thus forming a short production chain. The probability of such chain production is small however, since the average energy of the secondaries is quite low. The end of the main loop comes after all such residue particles are processed.
5. Data calculations: The details of the phase of the computations depend upon the specific goals and character of study. Our primary interest is the particle energy distribution (or number density vs. energy). This frequency distribution and other pertinent parameters are estimated (or calculated) using the stored data from various parameter counters in the main loop (tallied information) as well as tallied histories. The quantities of interest include local values (for the current time cycle only) and the cumulated values from several time cycles. After these calculations, the next phase is to prepare for the processing of new time

cycle.

6. Normalization and printout: Up to this stage, all the output quantities have been calculated. Next the output information is printed out, and at the same time, if the cycle number is the multiple of five, all the particle data and related parameter values are stored on the disk. Then the next run can be read off of the disk and the process can proceed from where we left off the last time. All the parameter values are again initialized, and new source particles are generated for the next time cycle. Particles that have escaped from the system or have been absorbed create holes in the particle data storage which are filled up by the new source particles and the residue particles produced during the time cycle. Before going on to the next cycle, the ion density is renormalized and corrected in a semiadaptive way according to a balance factor BF. The ion density distribution parameter N^+ is a dominate parameter since it affects several processes such as the rate of recombination and the rate of super-elastic collision (refer to Equations 27 and 28). It is corrected in such a way as to accelerate convergence, till the system reaches a balanced situation. Then increment time from t^n to t^{n+1}. The output quantities are the following:

   a) The kind of gas in the medium, source particle energy, pressure, slab thickness, and Δt the time interval of this cycle.

b) The number of time cycles, the time in seconds, the number of source particles generated during the cycle time, and the average energy of the particles in the system.

c) The number density distribution vs. energy, the single cycle distribution, the cumulated distribution, the number-density deviation, the average-energy deviation, the average energy, and the energy interval vs. energy.

d) The number density and the average energy vs. z (position zone).

e) The ion distribution (N_+), the number of ionizations (NIZN), the number of escaped particles (NES), the number of absorbed or recombined particles (NAS), the number of super-elastic collisions (NSE), the number of excitations to metastable levels (NMET), and the number of excitations to other levels (NEXC).

f) All the input control parameters as listed in section IV-1.

g) The electric field strength (EF), the source rate, the W-value, the average energy of escaped particles, and the average energy of absorbed particles.

h) The number check on the density disbributions.

All the above outputs include both current cycle values and the cumulated values over the past cycles.

7. Final checks: The last stage is to check whether the prescribed number of time cycles have been processed, adequate statistics have been collected, and all the useful data have been stored on the disk. If so, the program stops.

Further programming details are provided in Appendix

CHAPTER V

RESULTS

A. Convergence and Reliability

The present Monte Carlo Model is a predict-correct iterative scheme, with semiadaptive control ability. For such an iterative procedure, our main criteria for the evaluation and determination of convergence are based on the following inferences:

1. The balance factor (BF) as defined in Chapter II (Section II-B-3) signifies whether or not the system has reached steady state, i.e., converged to a final solution. At steady state, the value of BF should be approximately equal to unity. It may approach to unity in both directions, and due to semiadaptive control procedure used, it may oscillate around unity. Larger amplitude oscillations are observed at first, but they rapidly damp out.
2. The estimated values of the major parameters, e.g. the average energy and the W-value show a similar fluctuation, and then they converge to nearly constant values as steady state is approached.
3. The validity of the convergence has been checked using test problems where analytic results are available. A good comparison has been obtained, showing that the program and the model are working correctly, free of obscure errors.

B. Error Estimation and Analysis

1. Error Analysis

1. As mentioned in early chapters, Monte Carlo simulation is equivalent to performing mathematical experiments, the results being based on observations of such experiments. From this analogy we can derive the following procedures for the analysis of errors or accuracies of our calculations.

It is convenient to subdivide experimental errors into three broad types, namely: random errors, systematic errors, and blunders. In general, the experimental error is some additive function of all three, while blunders can hopefully be eliminated. We shall describe them in detail in following sections.

a. Random errors - such errors are of great concern in Monte Carlo type calculations. They generally involve statistical fluctuations or deviations, representing the difference between the singly measured value and the best value of a set of measurements, i.e., the difference between the arithmetic mean and the true mean. Sometimes such errors are referred to as elementary (or inherent) errors in the measurements. The commonly used measure of random errors is the variance or standard deviation.

b. Systematic error - a systematic error tends to have the same algebraic sign, i.e. it is either an additive or subtractive quantity introduced in the measurement

process. It is an unpleasant and insidious contribution which is not generally amenable to statistical treatment. Thus such errors seriously impair the reliability of the estimation. Typical examples include:

i) Incorrect assumptions or approximation in the representation of certain processes.

ii) Constructional faults and mistakes in the algorithms or subroutines.

iii) Inadequate regard of constancy of experimental conditions or inadequate sampling techniques (biased).

c. Blunders - these are outright mistakes which should be corrected by all means. Possible examples include:

i) Incorrect logic, misunderstanding of the problems.

ii) Errors in transcription of data.

iii) Mistakes in constants used.

iv) Confusion of units.

In any type of calculation, the systematic error as well as blunders should be removed. One way for detecting such errors is to compare the program against some known reliable solutions. Only the random errors are subject to reduction by the various treatments amenable to attack by variance reduction technqiues.

The precision in an estimated (mean) value is proportional to the reciprocal of the statistical error and is

high if the statistical error is small, i.e. the accuracy is high if the net systematic error is small. Usually, but not necessarily, high accuracy implies small statistical errors as well.

As discussed in later sections, our model and program has been checked against several test problems, and good agreement has been obtained. Therefore, it can be assumed that any serious systematic errors have been removed and that the program is free of blunders. Thus, we shall concentrate on the analysis of random errors.

No thorough analysis of variance has been made. Due to the intricate way that the histories calculated during a given time cycle depend on previous time cycles, a meaningful variance estimate beyond the first time cycle seems to be out of the question. However as the system reaches steady state, the deviations or variations (statistical fluctuations) of the estimated parameters can be estimated by means of standard deviation (or variance) over subsequent time cycles. Namely, we use the following formulation:

$$S_i = \left[\frac{\sum_{j=1}^{n} (f_{ij} - \bar{f}_i)^2}{n} \right]^{1/2} \quad \forall\, i \qquad (61)$$

where s_i: the standard deviation at the ith energy interval over n consecutive time cycles.

f_{ij}: the parameter value (density distribution function of ith energy interval, at jth time cycle.

$\bar{f}_i$: the average parameter value of ith interval over n time cycles.

A heuristic approach was adopted to measure local deviations or statistical fluctuations at each time cycle. This provides a relative magnitude of deviation or dispersion which can be used to compare the various variance reduction techniques. The approach is described as follows:

The estimated major frequency distribution (the electron energy density distribution function in our case) is obtained in the form of histogram. The abscissa of the histogram is divided into so-called class intervals (energy intervals in our calculations). In each such interval an erect rectangle (block) of heigh f_i and width E_i is formed. This block-area type of distribution is the fitted frequency distribution curve. We devise a numerical description of local deviations by defining:

i) the location index of the center of the erected rectangles in the histogram as i and with abscissa value E_i.

ii) the spread or dispersion around the center.

The total number of particles in each class interval (energy interval) is proportional to the area of the rectangular flock ($f_i \Delta E_i$), the particle's energy in this class interval falls in the range ($E_i - \Delta E_{i/2}$, $E_i + \Delta E_{i/2}$) with overall average energy $\overline{E}_i = E_i$. Then the following relationship holds (refer to Figure 5):

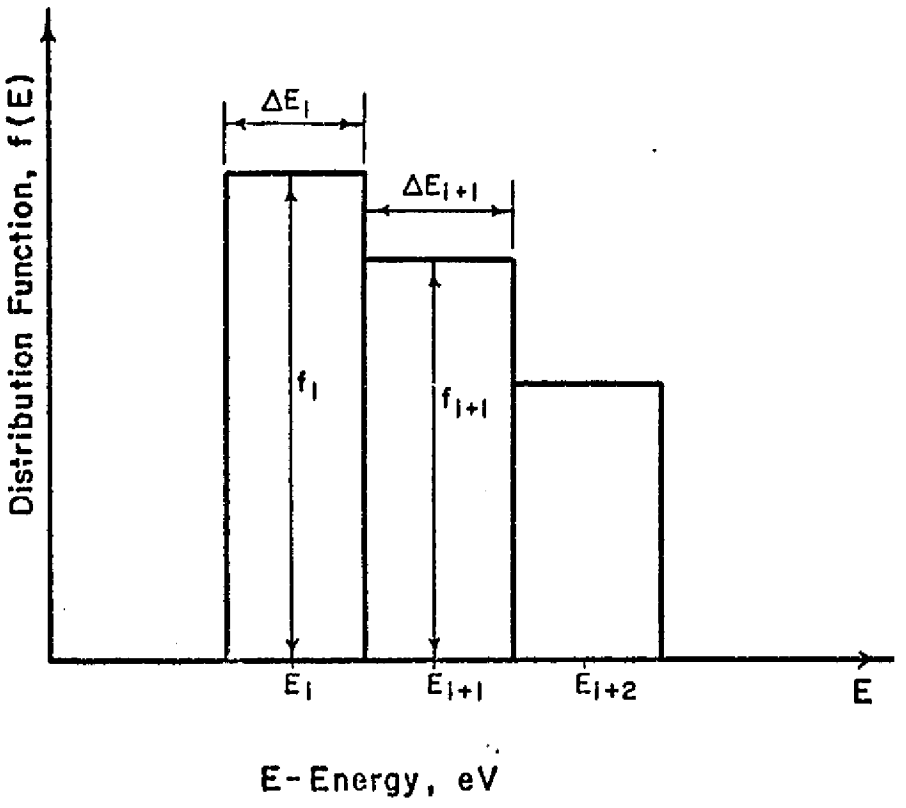


Figure 5. Rectangular block representation of the electron energy distribution histogram.

$$\frac{\Delta\overline{E}_i}{\Delta E_i} = \frac{\Delta f_i}{f_i}$$

or $$\Delta f_i = f_i \frac{\Delta\overline{E}_i}{\Delta E_i} \qquad (62)$$

where $\Delta\overline{E}_i$ is the deviation of the average energy $\overline{E}_i$ from E_i or $\Delta\overline{E}_i = \overline{E}_i - E_i$. Also Δf_i is the corresponding deviation of the frequency distribution at location index i.

Equation (62) provides a relative measure of the amplitude of the deviation of density distribution curve at energy E_i which can be used for comparing the efficiency of different variance reduction techniques as well as provide an heuristic clue to the modification of the fitted density distribution curve.

2. Data Filtering

The random errors (statistical fluctuations) result in bumps and ridges in the estimated frequency distribution function. It may be possible to alleviate many of these distortions by means of data smoothing techniques (data filtering).

Two types of data smoothing techniques commonly used in the processing of time series might be used here:

a. Moving averages - this technique operates by replacing each point of the frequency distribution function (height of the histogram at $E_i V_i$) at time cycle t_j with an average value of several subsequent points in the time cycle series. Thus if f_{ij} is the value

of frequency function at ith energy interval and jth time cycle, then f_{ij} is replaced by

$$\sum_{j=1}^{M} f_{ij} \cdot W_{j/M}$$

where W_j, the weighting factor for the jth time cycle, usually has the value of 1 or we can define it as a function of Δt_j. The value of M is the number of successive points of the time cycles to be included in each average.

b. Parabolic filtering - this is similar to the moving averages technique, but M is selected as an odd number, the point to be replaced is located in the center of these M points, and a parabola is fitted to these points by means of least squares. Thus it is replaced by the corresponding point f_i on the derived parabola. The process is repeated by shifting the center point to the next one in the time cycle series.

The technique of moving averages (a) is employed in the program, because of it's simplicity and the fact that the parabolic smoothing technique is too sensitive to the changes in the distribution function. Note that this smoothing (filtering) technique only becomes effective as the simulated system reaches steady state.

C. Solutions of the Problem

1. Preliminary Checks

In order to verify that the model is valid and correctly working, and the computer program is free of bugs, blunders, and systematic errors, we solved a specific problem first reported by J. A. Smit (11) and later extended by Heylen and Lewis (10). Their solutions for the electron energy distribution were obtained analytically based on the Boltzmann transport equation for the special case characterized as follows:

i) electric field present

ii) no external sources

iii) infinite size medium of background noble gases.

The comparisons are made for helium gas at $E_{f/p}$ = 4, 5, 10 V/cm^{-1}. ($E_{f/p}$: electric field strength to pressure ratio). The results are shown in Figures 6, 7 and 8. Both Smit and Heylen's results have taken into account all the collision processes including those due to elastic loss as well as inelastic losses. Heylen's results is more up-to-date, for low $E_{f/p}$ values only $E_{f/p}$ = 5 is given, which was chosen for the present comparison. In so far as possible, the same cross sections as Smit and Heylen and Lewis used were incorporated in the present calculations. The slight discrepancies apparant at the low-energy end is probably due to differences in treating the scattering process, and

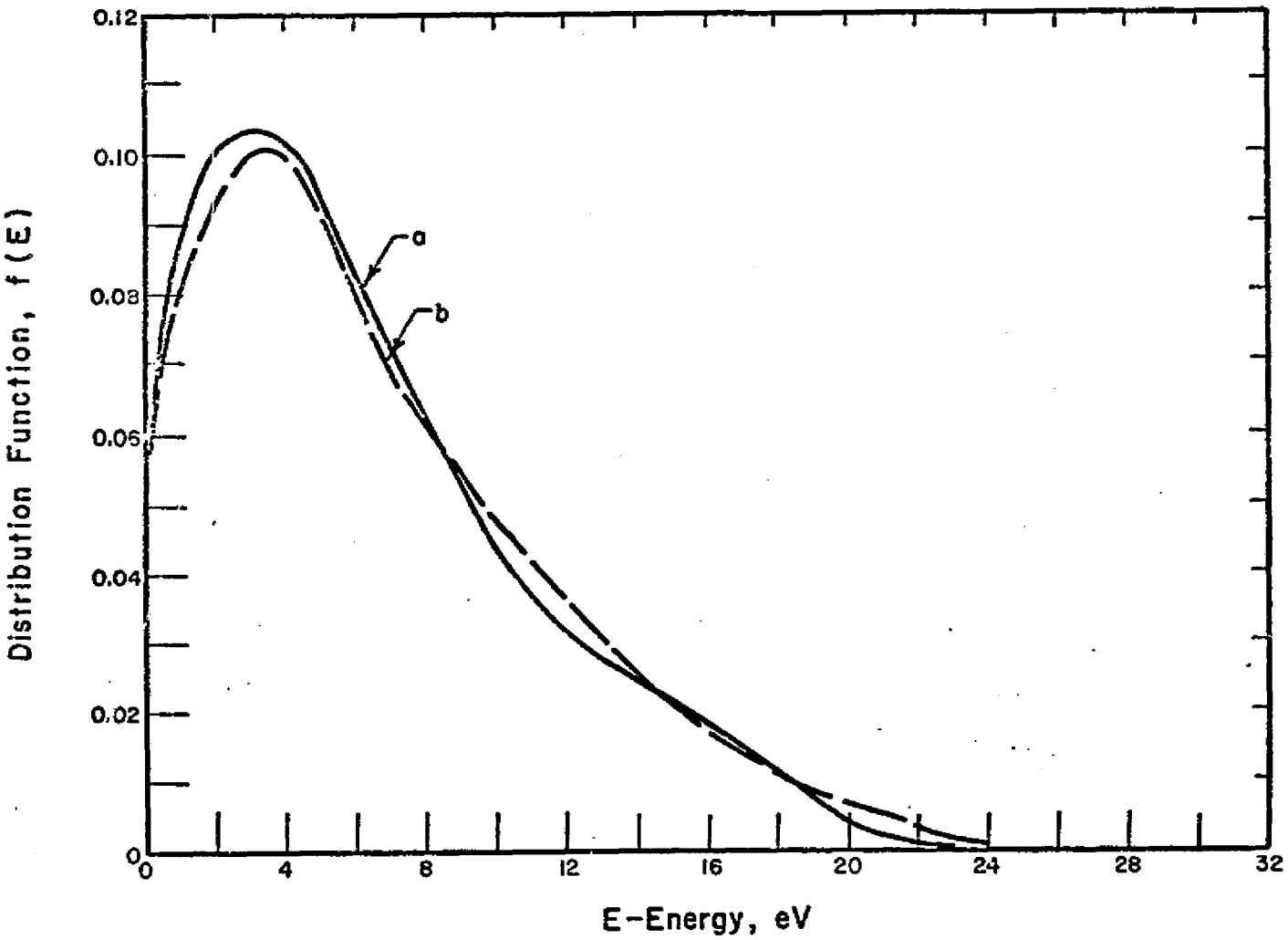


Figure 6. Comparison of the electron energy distribution for He at $E_{f/p} = 4$ for
a) Smit's calculation (1936) ($\overline{E} = 6.4$)
b) present calculation ($\overline{E} = 6.51$)

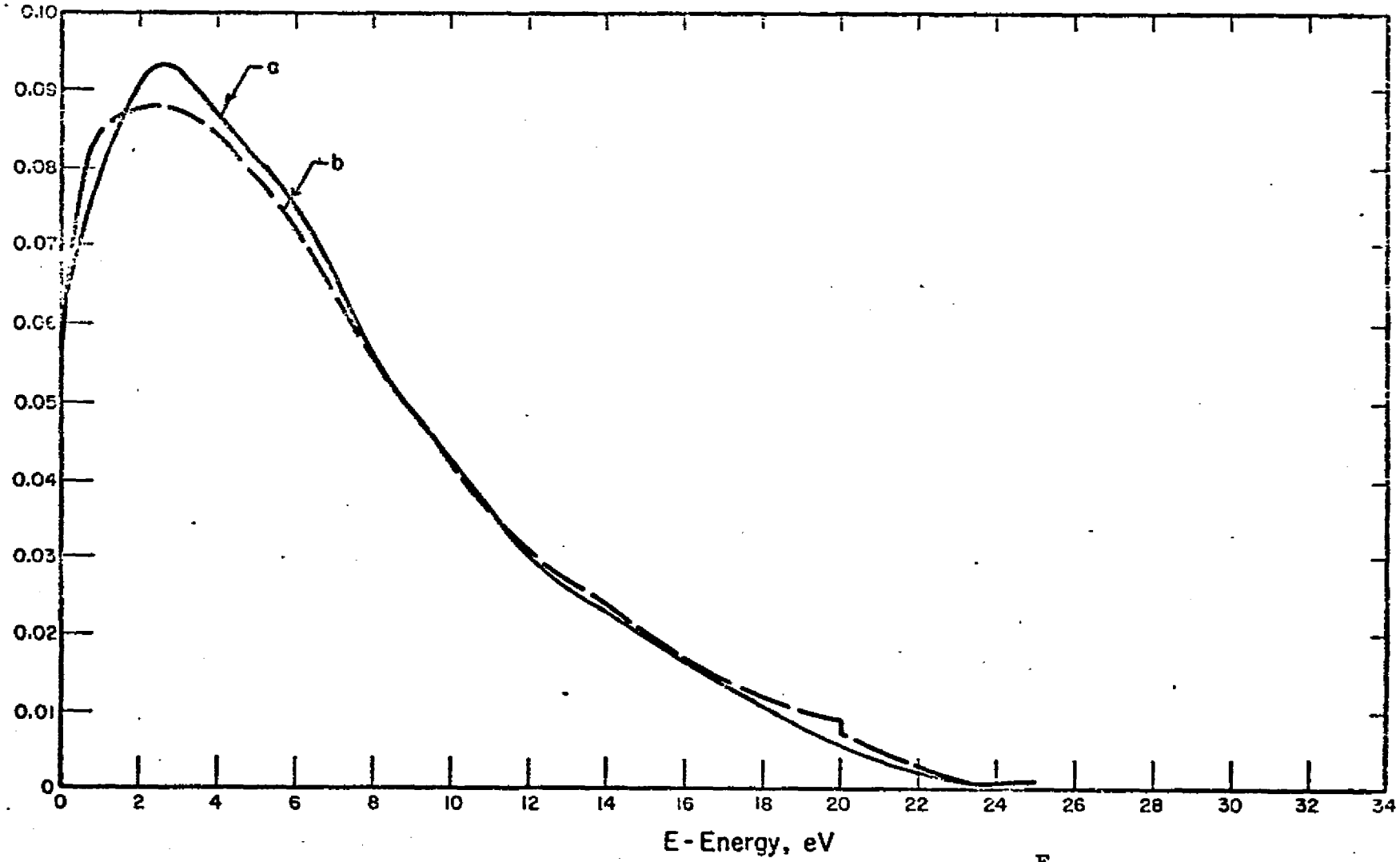


Figure 7. Comparison of electron energy distribution for He at $E_{f/p} = 5$
a) present calculation ($\overline{E} = 6.64$)
b) Heylen and Lewis' result ($\overline{E} = 6.51$)

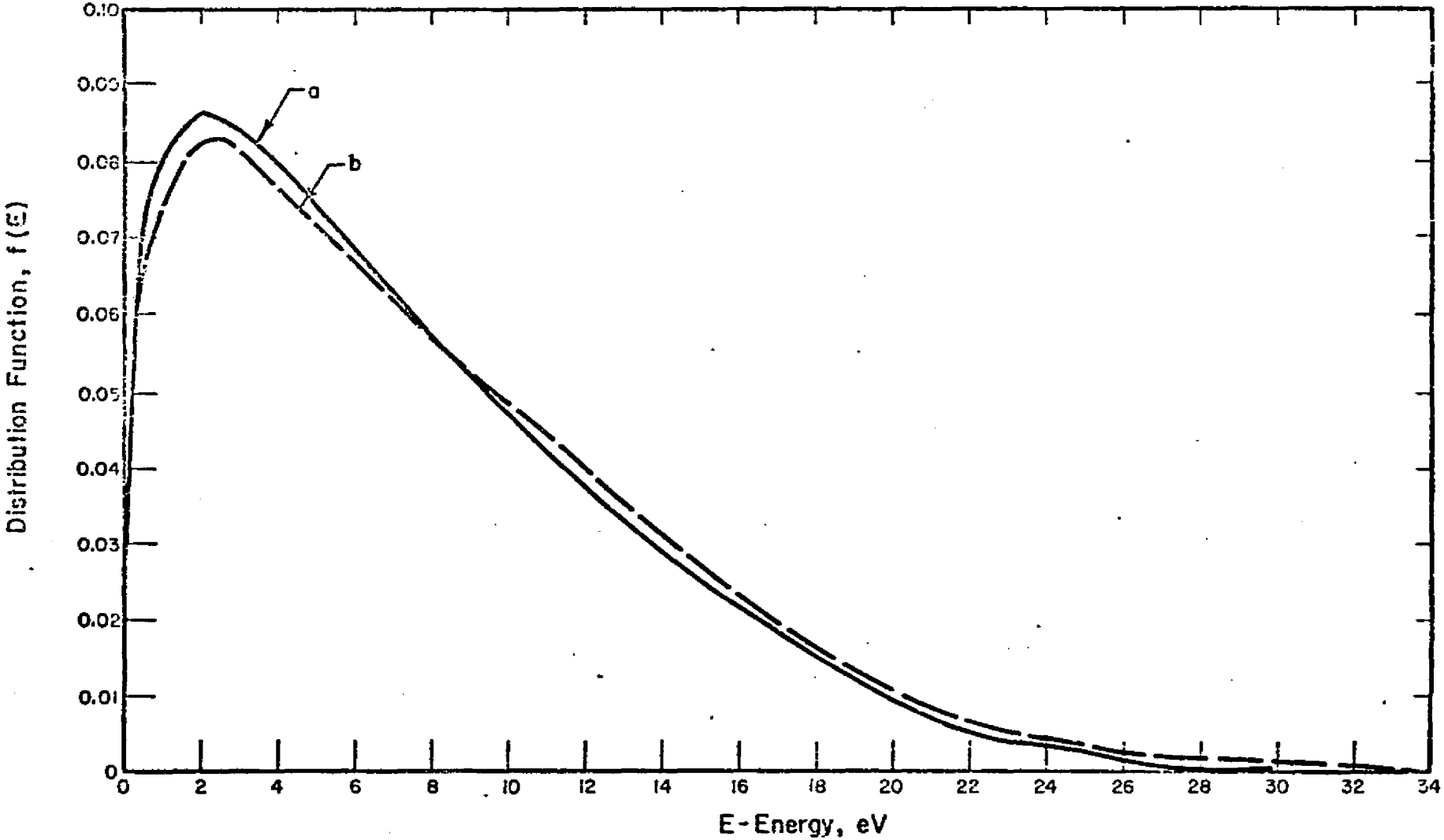


Figure 8. Comparison of electron energy distribution for He at $E_{f/p} = 10$
a) Smit's result ($\bar{E} = 7.7$)
b) present calculation ($\bar{E} = 7.83$)

the many approximations required in their analytic solutions. Also, some of the cross sections used in this calculations are experimental values, which may differ by some amount from that used for analytic solutions.

For the high energy region (near the source energy) checks have been made against the analytic solutions reported by Lo and Miley (32). As shown in Figure 9, the general shape of the electron energy flux distributions are in reasonable agreement. In this case it was not practical to use the same cross sections as incorporated in the analytic solutions, and this may account for some of the differences observed. Also the analytic solutions are obtained for an infinite medium vs. a 1 cm slab in the present case. Thus the analytic solutions do not allow leakage, although this becomes important in the present calculation for low pressures. Lo and Miley's results are plotted in Figure 9 for the source energy E_0 = 1500 eV and E_0 = 500 eV, whereas the present results are for E_0 = 1000 eV. The plot shown is for the flux density (normalized) instead of energy number density. They are related as follows:

$$\varphi(E) = f(E) \cdot V(E) \qquad (63)$$

where $\varphi(E)$: electron flux density

$f(E)$: electron number density

$V(E)$: Electron kinetic velocity

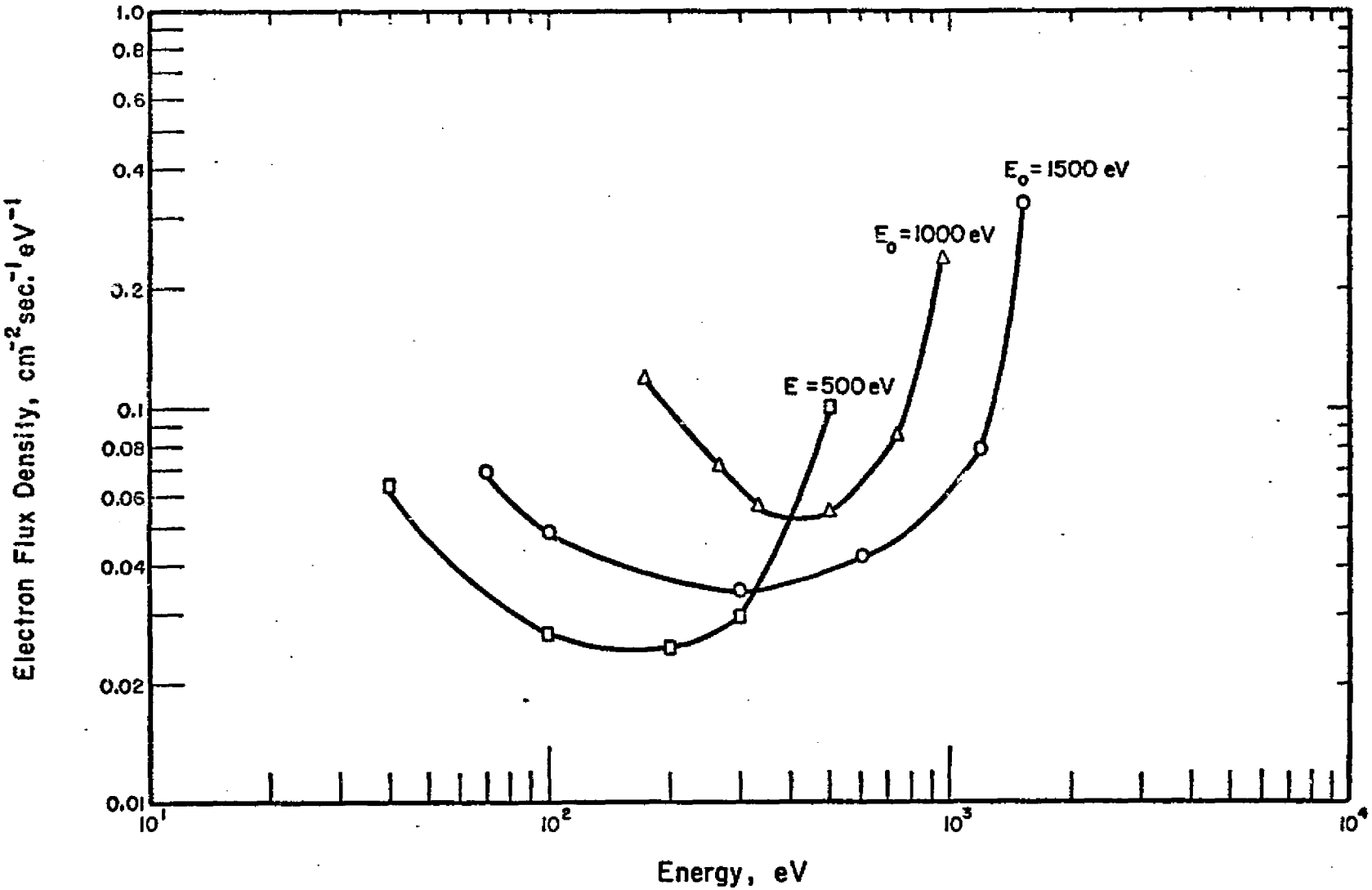


Figure 9. Comparison of the high-energy portion of flux density curve
a) E_0 = 1500 eV and E_0 = 500 eV: Lo and Miley's result
b) E_0 = 1000 eV: present calculations

Since we only intend to compare general shapes of distribution, the normalizations used for the various curve, shown here are arbitrarily choosen for optimum display of information.

2. The Electron Energy Distribution Function without Electric Field

The calculations for electron energy distribution functions for helium with a high energy electron source, but no applied electric field (Figures 10 to 14) were carried out at two different pressures and various source rates. The slab thickness D is chosen to be fixed at D = 1 cm, which is equal to the actual diamter of cylindrical tube used in the laboratory (7). The two pressures are chosen to be P = 10 torr and P = 760 torr, corresponding to a typical low-pressure case and a normal atmospheric pressure.

We have chosen two different source energies (E_0), one at E_0 = 1,000 eV and the other at E_0 = 70 eV. These values roughly correspond to the highest energy and the average energy in the primary electron energy spectrum for MeV alpha-particle irradiation of helium as done in the experimental studies conducted at the University of Illinois (6).

The range of interest for the source rates for a boron coated tube irradiated in the Illinois TRIGA reactor lies between 10^{14} to 10^{18} (#/cm^3-sec). However, results shown for source rates above about 10^{16} are questionable since the resulting ionization density becomes large enough that

electron-electron collisions become important, and this process was neglected here.

The major quantity of interest here is the electron energy distribution function which were defined as $n^-(E)$ in Chapter II. For simplicity, we represent the normalized distribution corresponding to $n^-(E)$ by $f(E)$. This normalization is defined by:

$$\int_0^{E_o} f(E)\, dE = 1 \qquad (64)$$

For the plasma system without electric field ($E_f = 0$), the distribution functions with source energy $E_o = 1000$ eV are shown in Figures 10 and 11 for the pressure $p = 10$ torr, and $p = 760$ torr respectively. Similar distribution functions for $E_o = 70$ eV are shown in Figures 12 and 13.

An idealized Maxwellian distribution plus 1/E tail are shown in Figures 10 to 13 for reference. (The Maxwellian plus 1/E distribution was used as the initial distribution for these calculations. Note that since this curve is for reference only, it is arbitrarily normalized to a point on the Maxwellian curve and does not satisfy Equation 64.) For the range of source rates mentioned, the calculated energy distribution for helium (He) are reasonably Maxwellian at low energy region. The differences due to variations in the source rates show up at the high energy end, especially near the source energy, where the magnitudes seem to depend linearly on the values of source rate. This is somewhat

deceptive, however, since the absolute magnitude at any point is due to the normalization. (This point is stressed again later in connection with Figure 24).

The fact that the distributions drop off rapidly for energies above about 20 eV is attributed to the fact that the threshold energy for inelastic processes is 20 eV, above which energy the ionization and excitation processes bring more electrons into the low-energy region.

In order to investigate the change in distribution function at very high source rates, a calculation for S_o = 10^{22} particles/cm^3-sec was carried out. As mentioned earlier, the neglect of electron-electron collisions makes the accuracy of this calculation questionable. Still the trends are of interest, and it is seen from Figure 14 that the distribution curve changes drastically. In fact, it is no longer Maxwellian at the low-energy end. Due to the high source rate, more high-energy particles and more particles with energy in the range from 0.3 eV to 20 eV are present in the system.

As discussed earlier, the distribution curve drops sharply for energies above the excitation threshold energy (at E = 20 eV) due to the dominance of ionization and excitation collision processes. At the higher pressure, the distribution curves decrease more rapidly compared to low pressure, since increased scattering slows the electrons down more rapidly. This effect is shown in Figures 15 and 16,

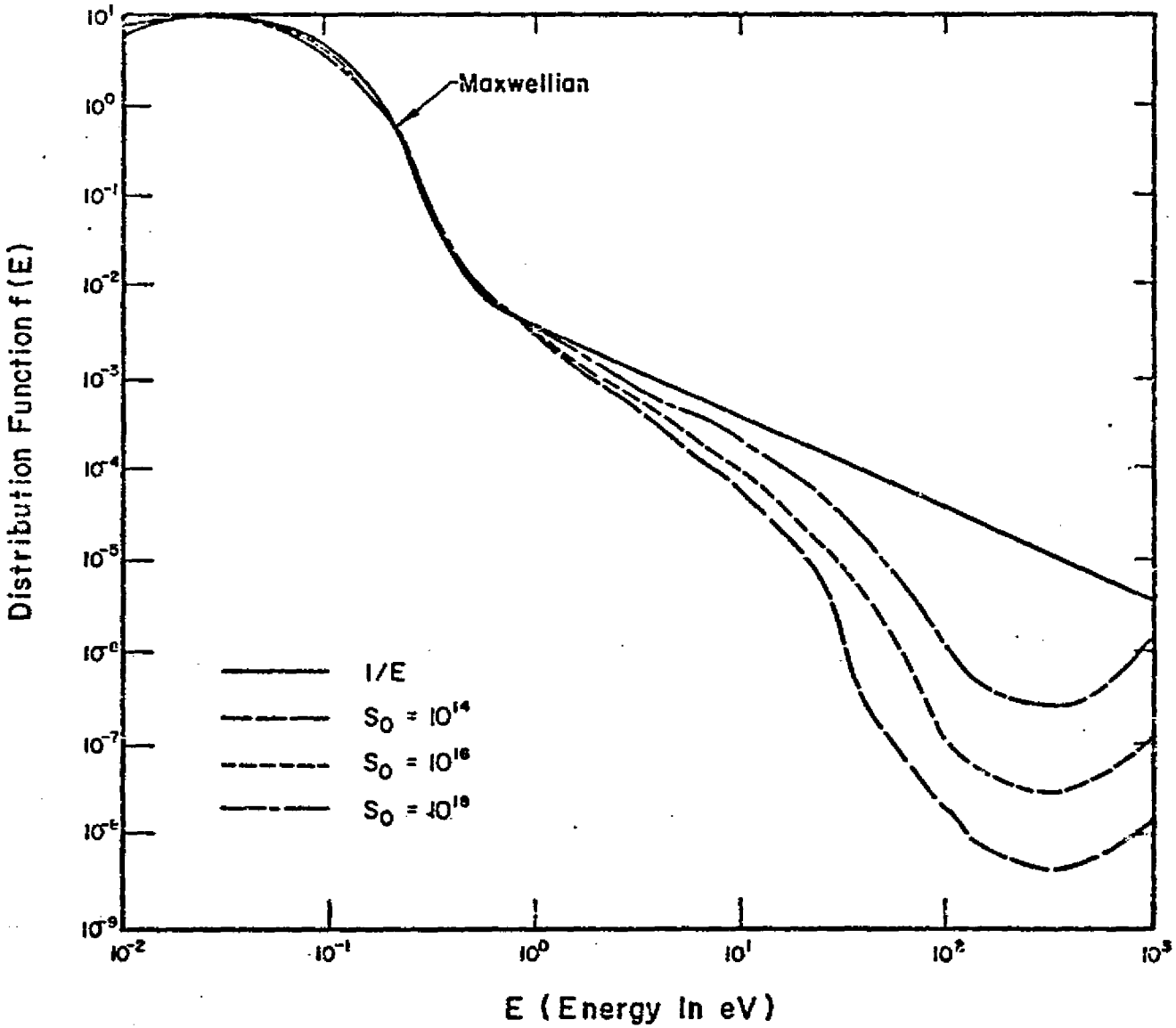


Figure 10. The electron energy distribution for He without an electric field but different source rates at p = 10 torr, E_o = 1000 eV and Maxwellian plus 1/E distribution.

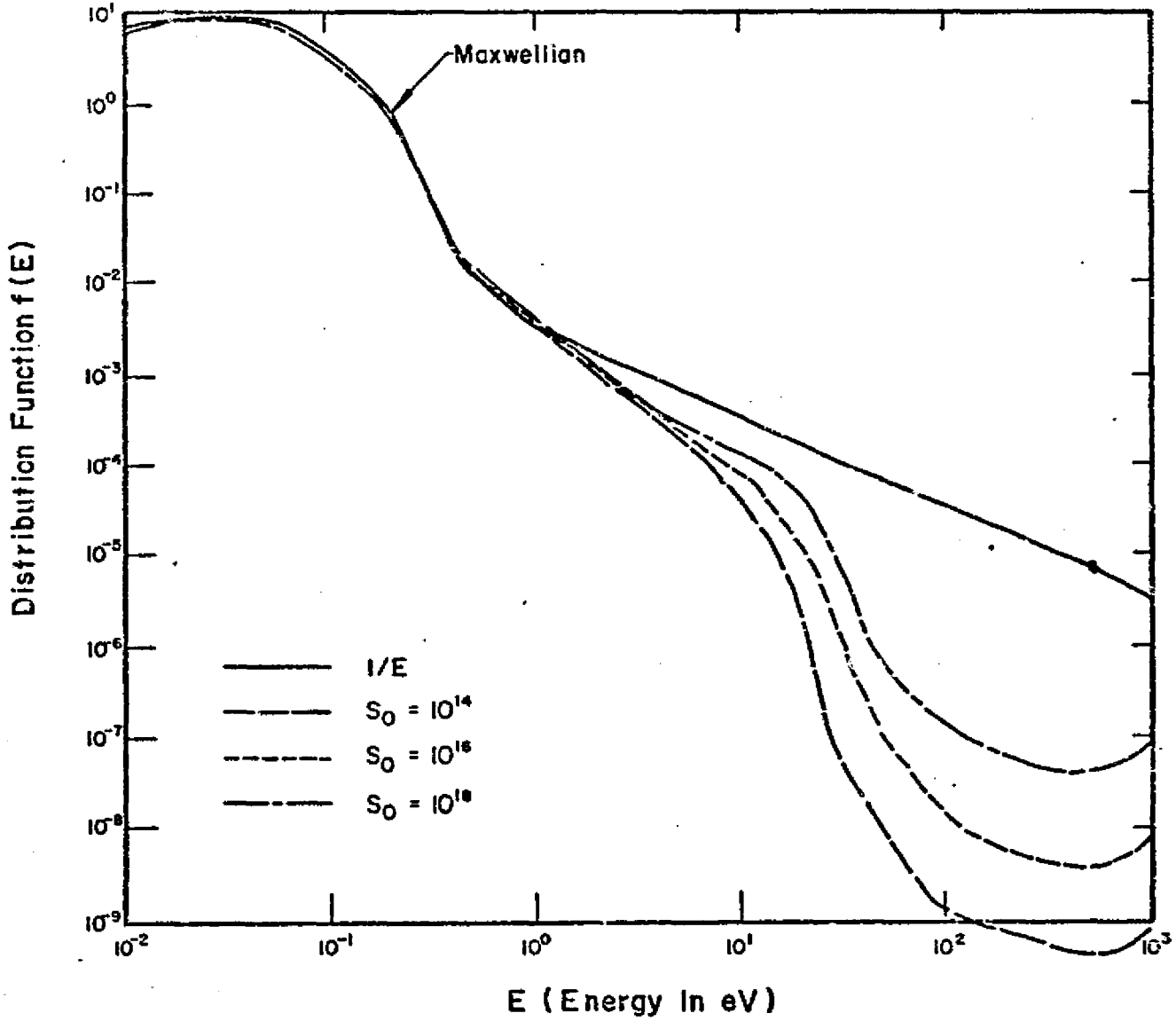


Figure 11. The electron energy distribution for He without an electric field but different source rates at p = 760 torr, E_0 = 1000 eV and Maxwellian plus 1/E distribution.

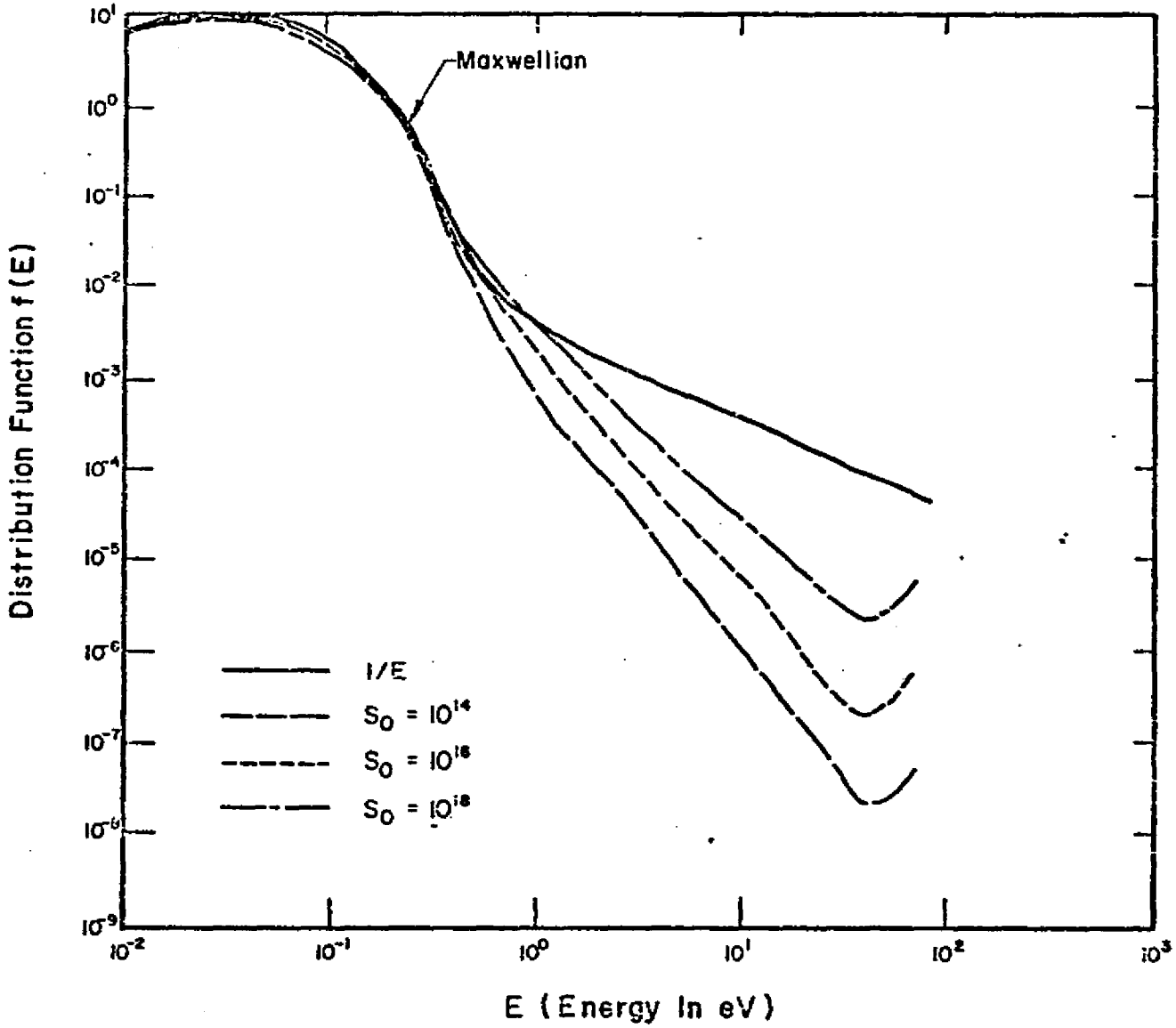


Figure 12. The electron energy distribution for He without an electric field but different source rates at p = 10 torr, E_0 = 70 eV and Maxwellian plus 1/E distribution.

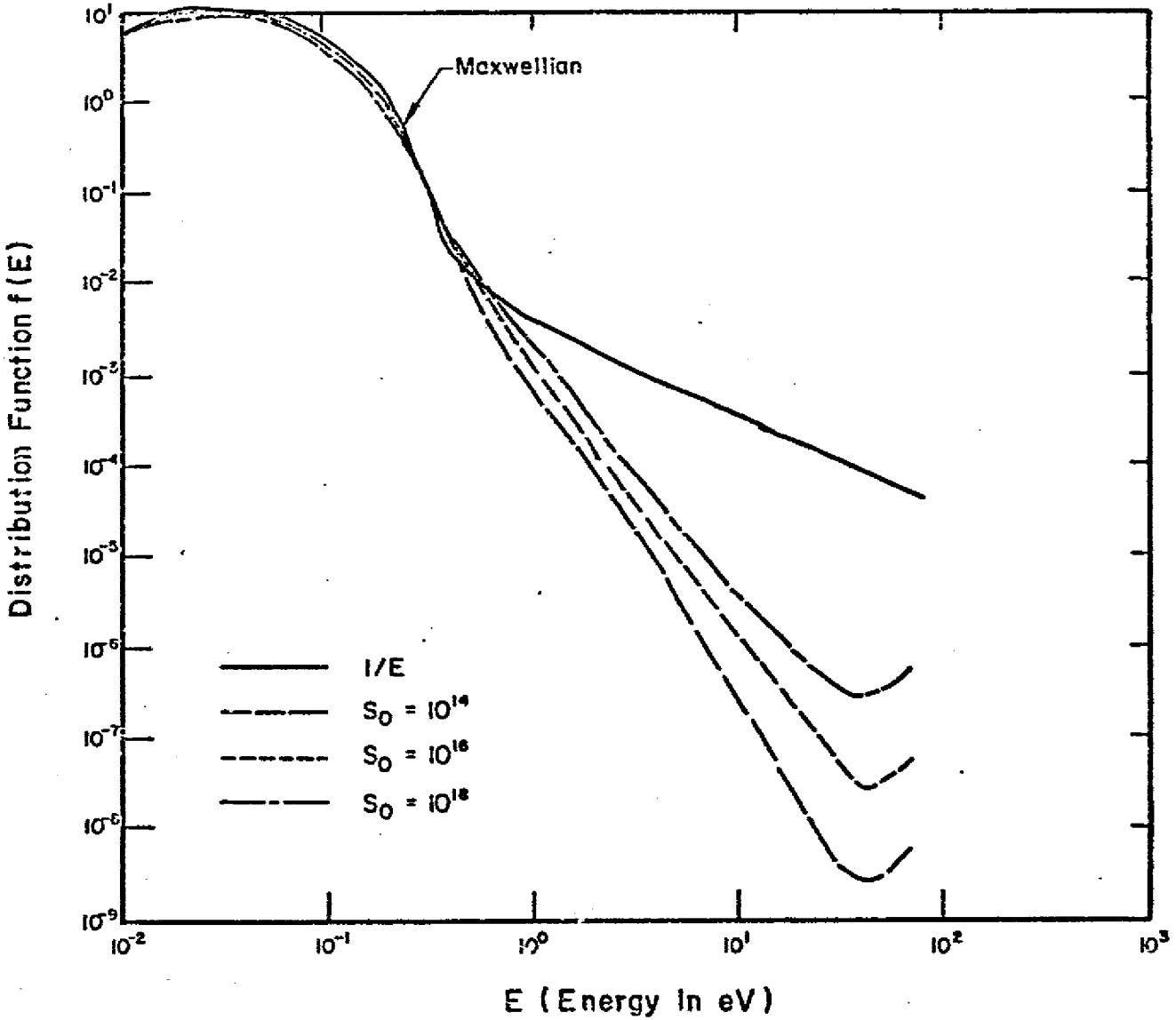


Figure 13. The electron energy distribution for He without an electric field but different source rates at p = 760 torr, E_0 = 70 eV and Maxwellian plus 1/E distribution.

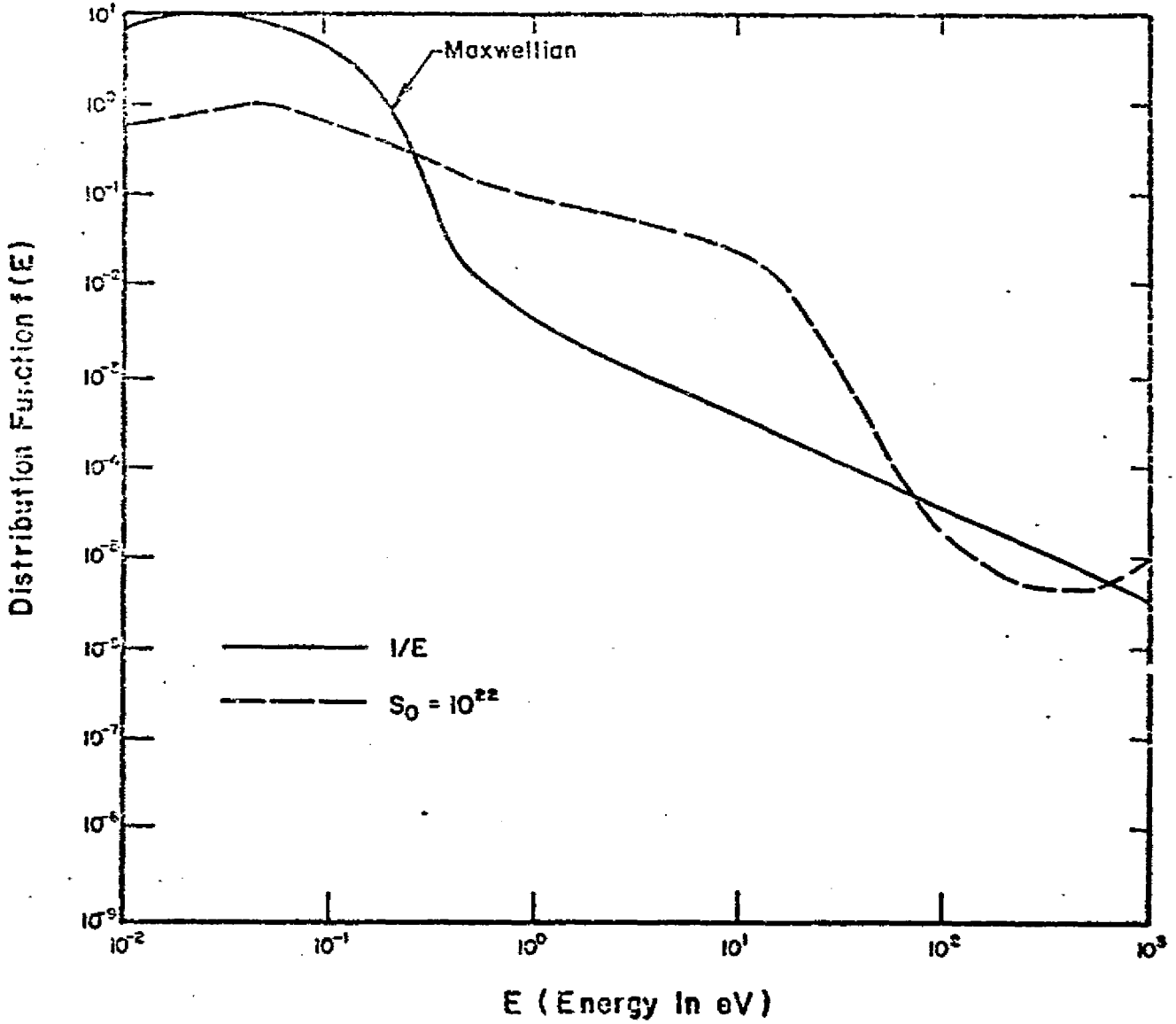


Figure 14. Comparison of f(E) for He without an electric field (source rate S_o = 10^{22} #/cm^3-sec) with the standard Maxwellian distribution plus 1/E tail.

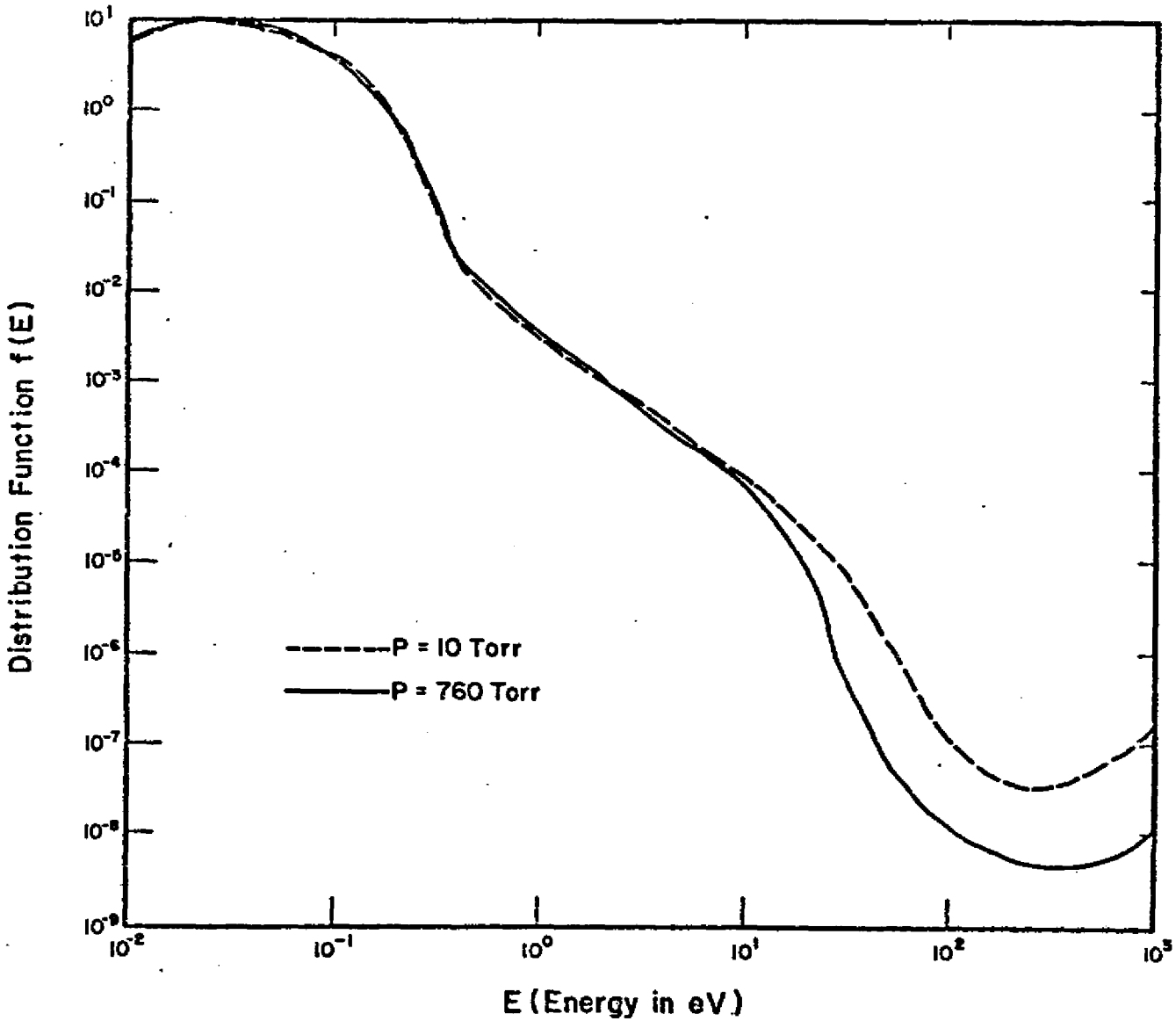


Figure 15. The electron energy distribution for He, without an electric field, for different pressure at $S_o = 10^{16}$, $E_o = 1000$ eV.

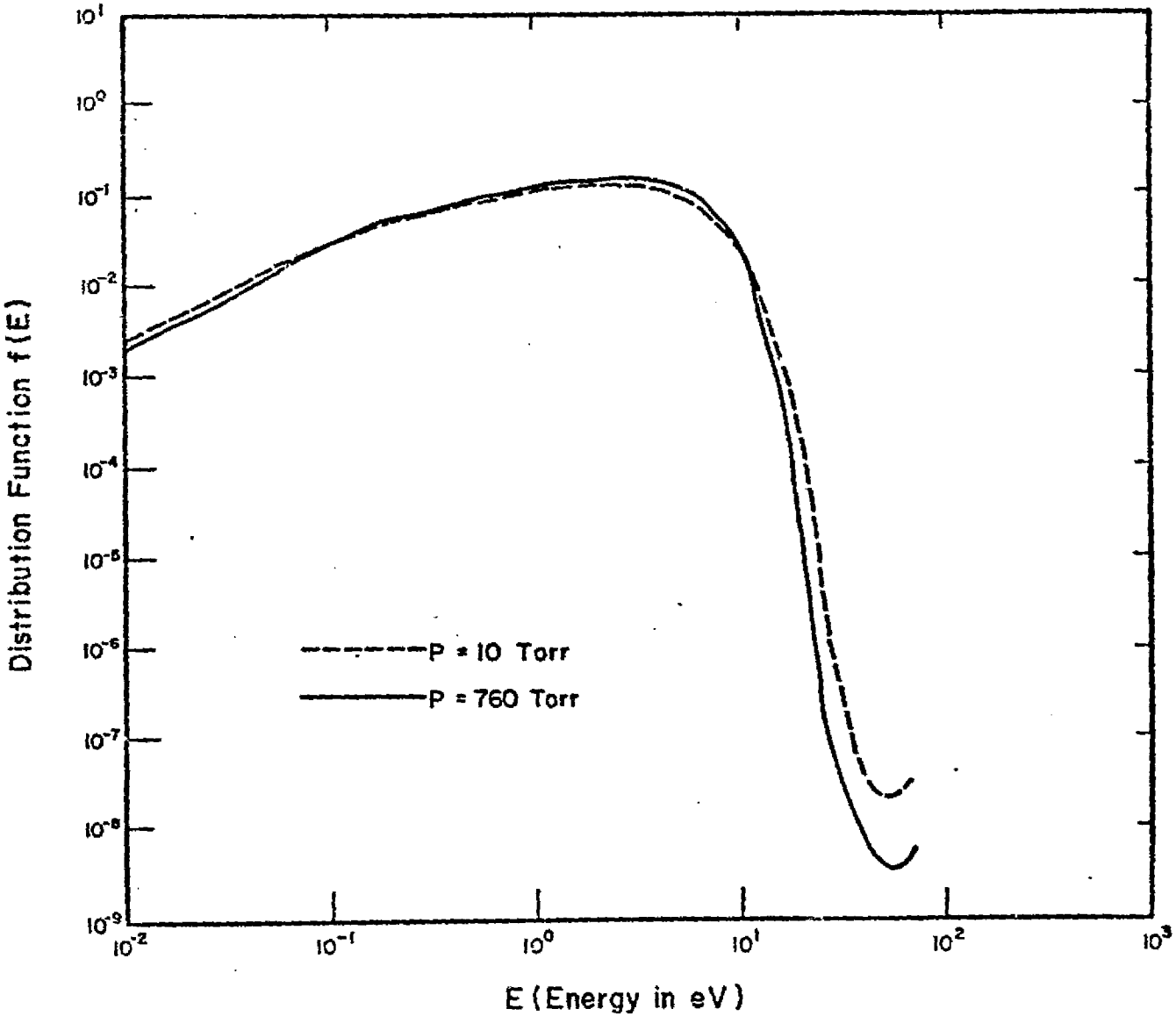


Figure 16. The electron energy distribution for He, without an electric field, for different pressure at $S_o = 10^{16}$, $E_o = 70$ eV.

for the same source rate $S_o = 10^{16}$ (#/cm^3-sec) and source energy, but two different pressures. The reduction in magnitude of the high energy tail as the pressure increases is clearly illustrated.

In these calculations, the rate of both leakage and absorption (recombination) are strongly dependent on the pressure. At higher pressures (e.g., p = 760 torr), very few particles leak out of the system, and the loss are dominated by the absorption process. For lower pressure (p = 10 torr), fewer particles are absorbed, while more escape or leak out of the system. Also, more particles escape in the high-energy region, but the absorped particles are mainly in the low-energy region.

The estimated major parameters are summarized in Table 1, which includes the average energy of the system ($\bar{E}$), the number of positive ions (N^+) and the W-values for various combinations of source rate S_o, source energy E_o and pressure p.

The W-value is defined as the amount of energy required (input) to produce an ionization pair (positive ion and electron), or

$$\text{W-value} = \frac{(N_s.E_o - E_{Tes} - E_{Tab})}{N_z} \qquad (65)$$

Where N_s: the number of source particles introduced into the system during the period Δt.

E_{Tes}: the total energy lost with escaping particles during the time Δt.

Table 1. Calculated Parameter Values for the Plasma Without a Electric Field

* P (torr)	E_o (eV)	S_o (#/cm^3-sec)	$\bar{E}$ (eV)	N^+ (#)	W-value Range
10	1000	10^{18}	0.171	0.125×10^{14}	41 ~ 63
10	1000	10^{16}	0.157	0.125×10^{13}	35 ~ 53
10	1000	10^{14}	0.111	0.125×10^{12}	49 ~ 58
760	1000	10^{18}	0.167	0.583×10^{15}	32 ~ 38
760	1000	10^{16}	0.123	0.446×10^{14}	40 ~ 65
760	1000	10^{14}	0.104	0.67×10^{13}	35 ~ 54
10	70	10^{18}	0.104	0.125×10^{14}	50 ~ 62
10	70	10^{16}	0.082	0.125×10^{13}	47 ~ 63
10	70	10^{14}	0.075	0.124×10^{12}	40 ~ 57
760	70	10^{18}	0.082	0.16×10^{15}	47 ~ 65
760	70	10^{16}	0.075	0.141×10^{14}	42 ~ 58
760	70	10^{14}	0.075	0.157×10^{13}	39 ~ 55
Maxwellian distribution					
		0	0.05	---	---

*P: pressure, E_o: source energy, S_o: source rate, $\bar{E}$: average energy of the particles, N^+: number of positive ions.

E_{Tab}: the total energy lost with recombining particles during the time Δt

N_z: the total number of ionizations produced during the time period Δt.

As seen from Table 1, the average energy $\bar{E}$ tends to be higher for higher source rates and higher source energy, and it is lower for higher pressures. However, the differences are in general very small. (All the average energies are consistently greater than that of a Maxwellian distribution). The values for ion density N^+ are roughly proportional to the square root of source rate, and are higher for higher pressures, virtually independent of source energy.

The W-values are quite sensitive to statistical fluctuations, and the ranges of variation obtained over the last 10 time cycles in a given calculation are shown in Table 1. The W-values obtained here do not show any clear dependence on the system parameters, thus variation indicated may be totally due to the statistical fluctuations. This is not too surprising since the W-value is known to be roughly independent of particle energy, etc. The W-value for helium obtained experimentally by Jesse and Sadouski (34) is 43. A reasonable value to choose for the W-value from Table 1 would be 50, which is little bit higher than the experimental result. The reason for this discrepancy is not clear, but it may be due to small inaccuracies in the ionization-excitation cross section employed here. The W-value is, by definition, quite sensitive to the details

of the ratio of the ionization to the excitation cross section as a function of energy.

3. The Electron Energy Distribution Function with an Applied Electric Field

The calculated electron energy distribution functions in He with an applied electric field ($E_f/p = 10$) are plotted in Figures 17 to 21, in a manner similar to that in the previous section. All these calculations are carried out at the electric field to pressure ratio $E_f/p = 10$. Smit's result (without source or $S_o = 0$) are plotted together with present calculations at given pressure and source energy ($p = 10$, 760 and $E_o = 1000$ eV, 70 eV), with three different source rates. It is observed that the distribution functions are highly non-Maxwellian. For the source rate in the range 10^{14} #/cm^3-sec to 10^{18} #/cm^3-sec, the distribution curves at low energy end are quite close to the Smit's results (11). At the high energy end, they have a shape similar to the curves of no electric fields. Again differences due to the source rate show up at high-energy region. Below 10 eV, all the distribution functions fall on the top of each other, and they begin to spread out as the energy increases above 10 eV. For high pressures or low source energy, more difference (between $S_o = 0$ and $S_o \neq 0$) show up at high-energy end than that of low pressure and/or high source energy. Again at very high source rates ($S_o = 10^{22}$ #/cm^3-sec) the distribution function changes appreicably.

In Figure 21, the comparison is made with a very high source rate ($S_o = 10^{22}/cm^3$-sec) and with one in the region of interest ($S_o = 10^{16}$, p = 760 torr). For very high source rate, the distribution function changes quite a lot; it still preserves the general shape, but at low-energy end, it is much smaller in amplitude. (However, as stressed earlier, the high source rate calculation is questionable due to the neglect of electron-electron interactions.)

The changes in these distribution functions due to different pressures are also shown in Figures 22 and 23. The observed trends are similar to those in the previous section for the electric field $E_f = 0$ case.

So far, the normalized distribution functions f(E) have been displayed. However, to stress the fact that the absolute magnitudes of the curves depends strongly on the source (shapes have received our interest to this point), Figure 24 shows the unnormalized electron energy distribution functions for He (with electric field or $E_f \neq 0$) are plotted, with $E_f/p = 10$, p = 10 torr, $E_o = 1000$ eV, and three different source rates ($S_o = 10^{14}$, $S_o = 10^{16}$, $S_o = 10^{18}$). The magnitude (unnormalized) distribution is defined as

$$N^- (E) = N^+ \cdot f(E) \qquad (66)$$

where N^+: number of positive ions

f(E): normalized distribution

The values of N^+ are 0.24×10^{15}, 0.24×10^{14}, and 0.24×10^{13} corresponding to the three source rates $S_o = 10^{18}$, $S_o = 10^{16}$

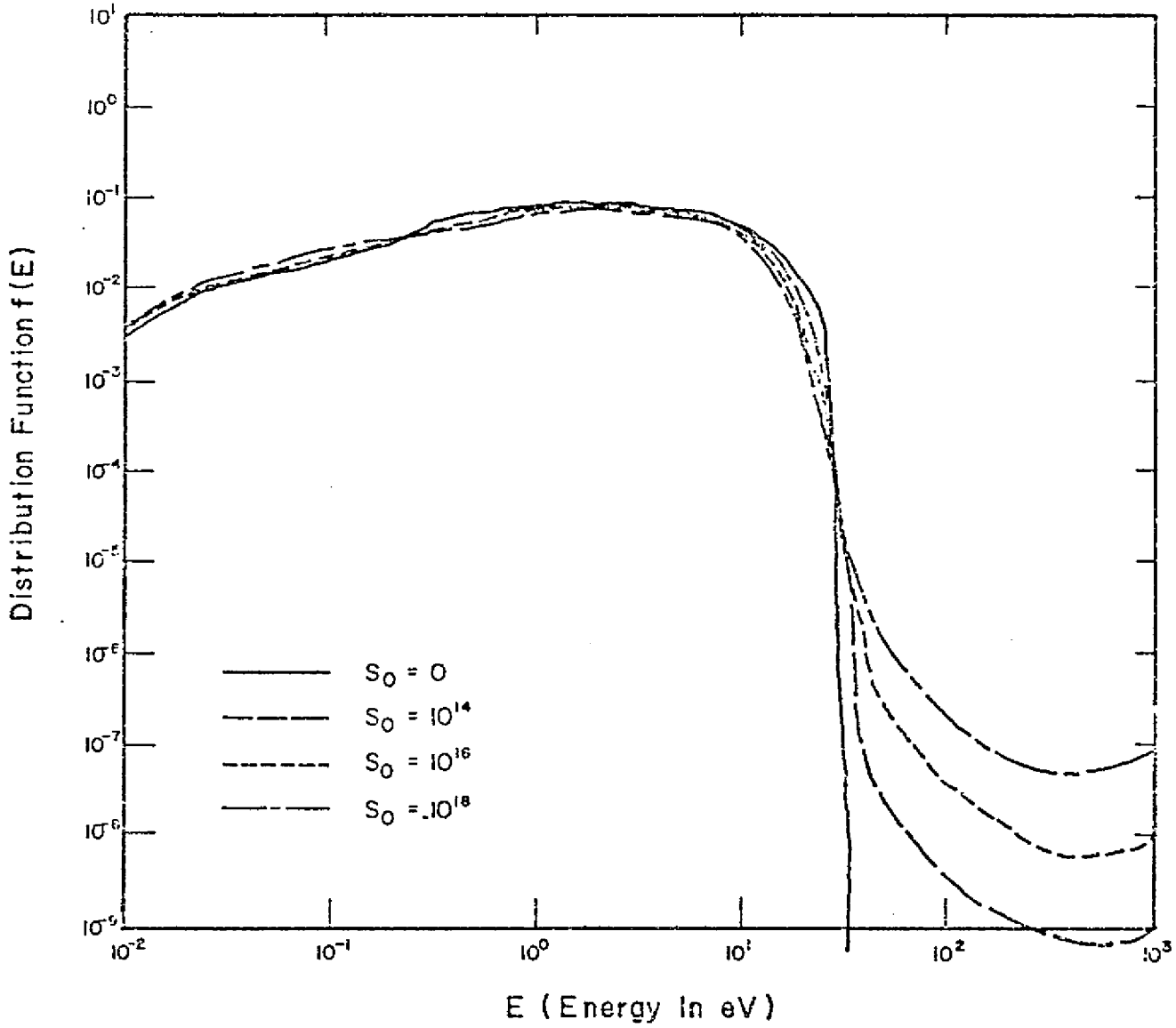


Figure 17. The electron energy distribution for He with an electric field at $E_{f/p}$ = 10, p = 10 torr, E_o = 1000 eV.

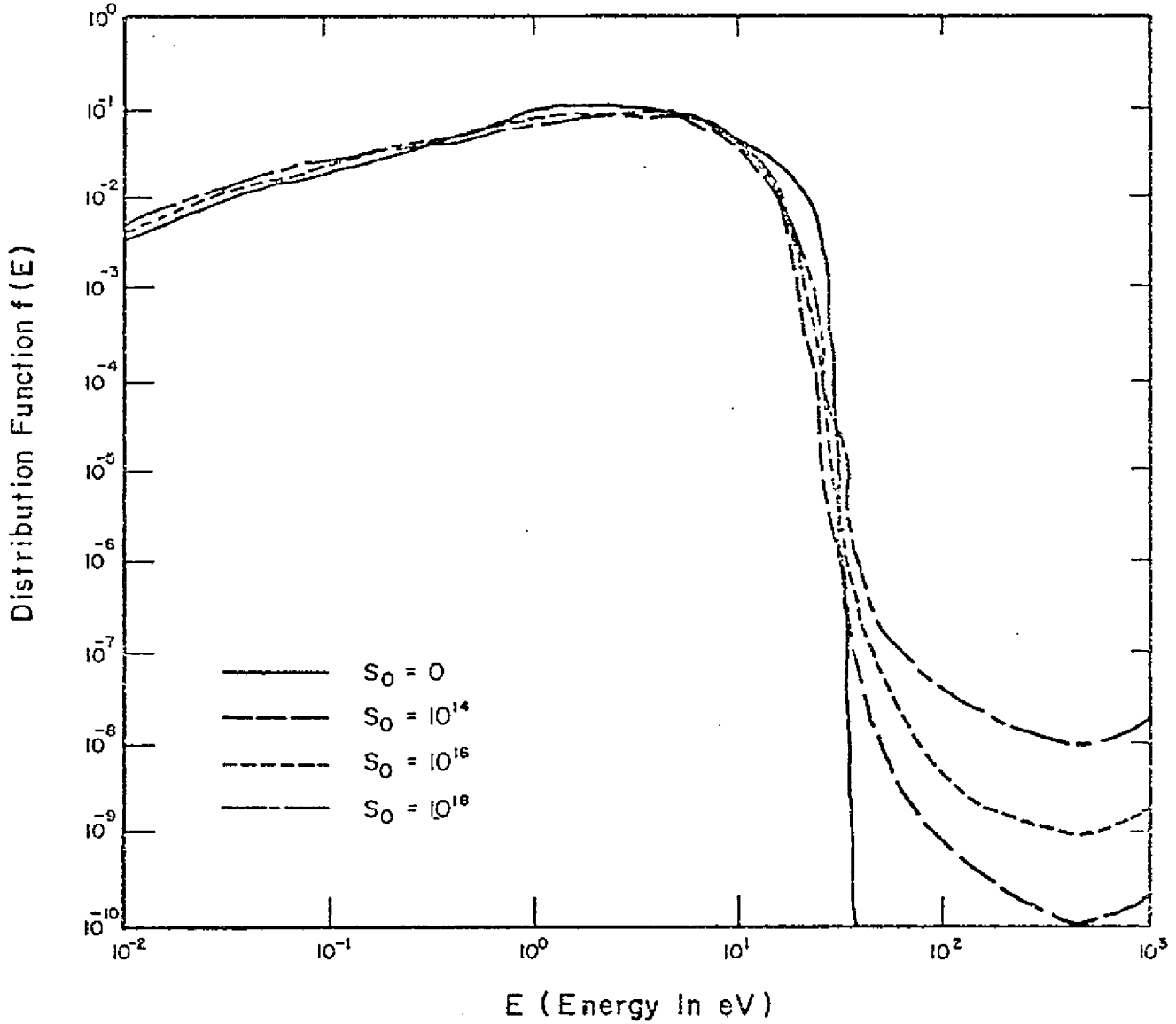


Figure 18. The electron energy distribution for He with an electric field at $E_{f/p}$ = 10, p = 760 torr, E_o = 1000 eV.

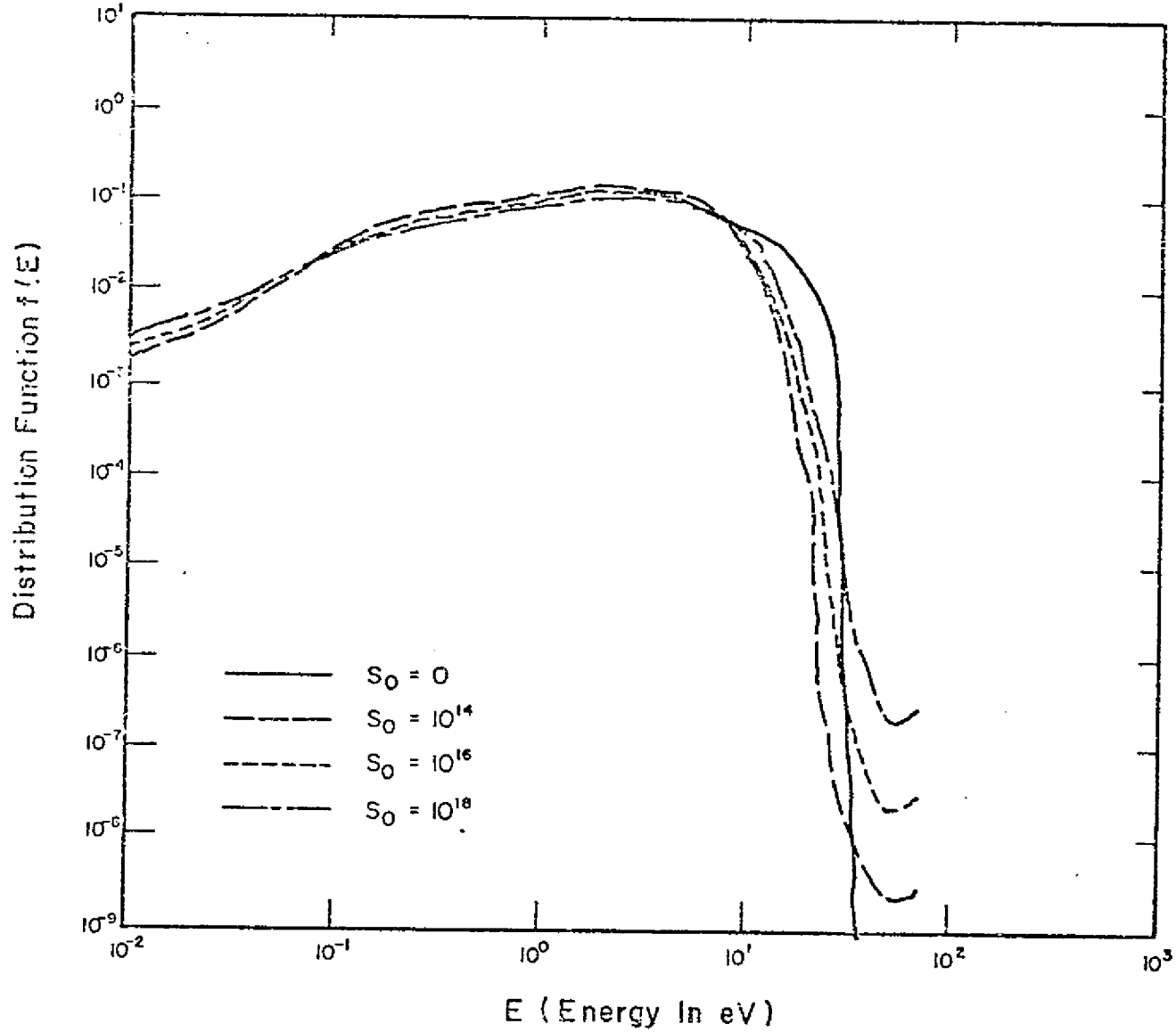


Figure 19. The electron energy distribution for He with an electric field at $E_{f/p}$ = 10, p = 10, E_o = 70 eV.

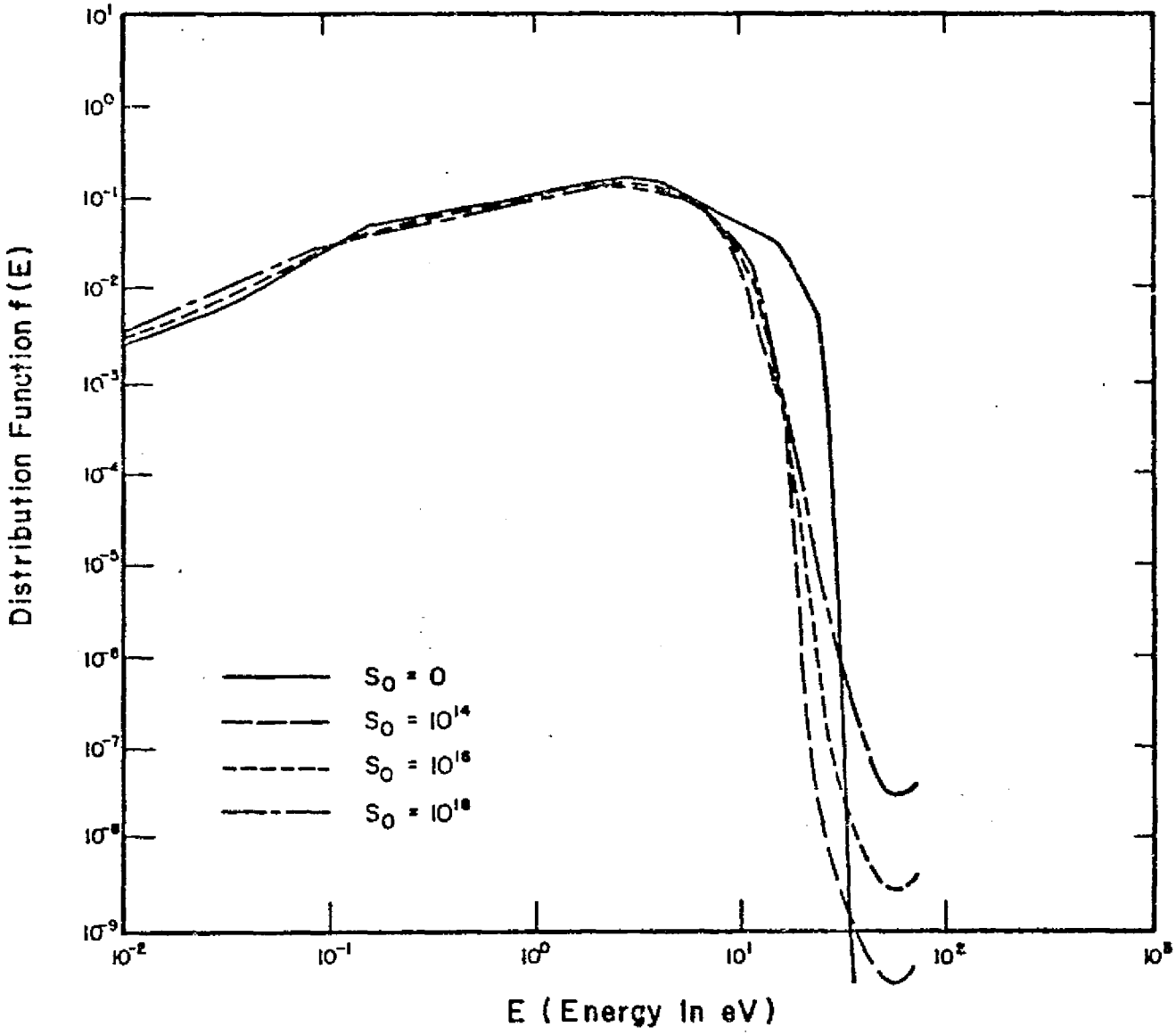


Figure 20. The electron energy distribution for He with an electric field at $E_{f/p}$ = 10, p = 760 torr, E_0 = 70 eV.

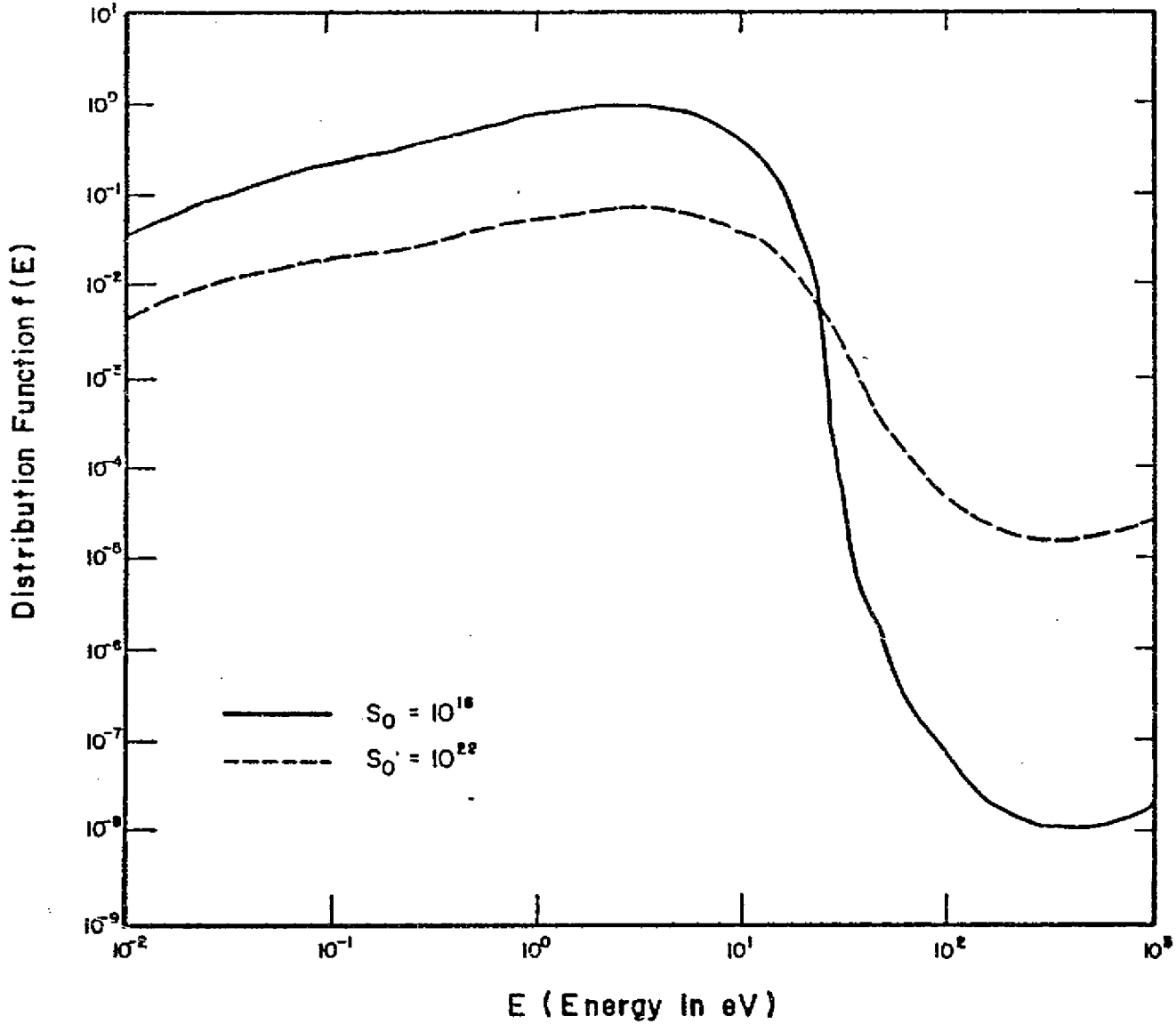


Figure 21. The electron energy distribution function at $E_{f/p} = 10$, p = 760 torr for $S_o = 10^{22}$ and $S_o = 10^{16}$ (#/cm^3-sec).

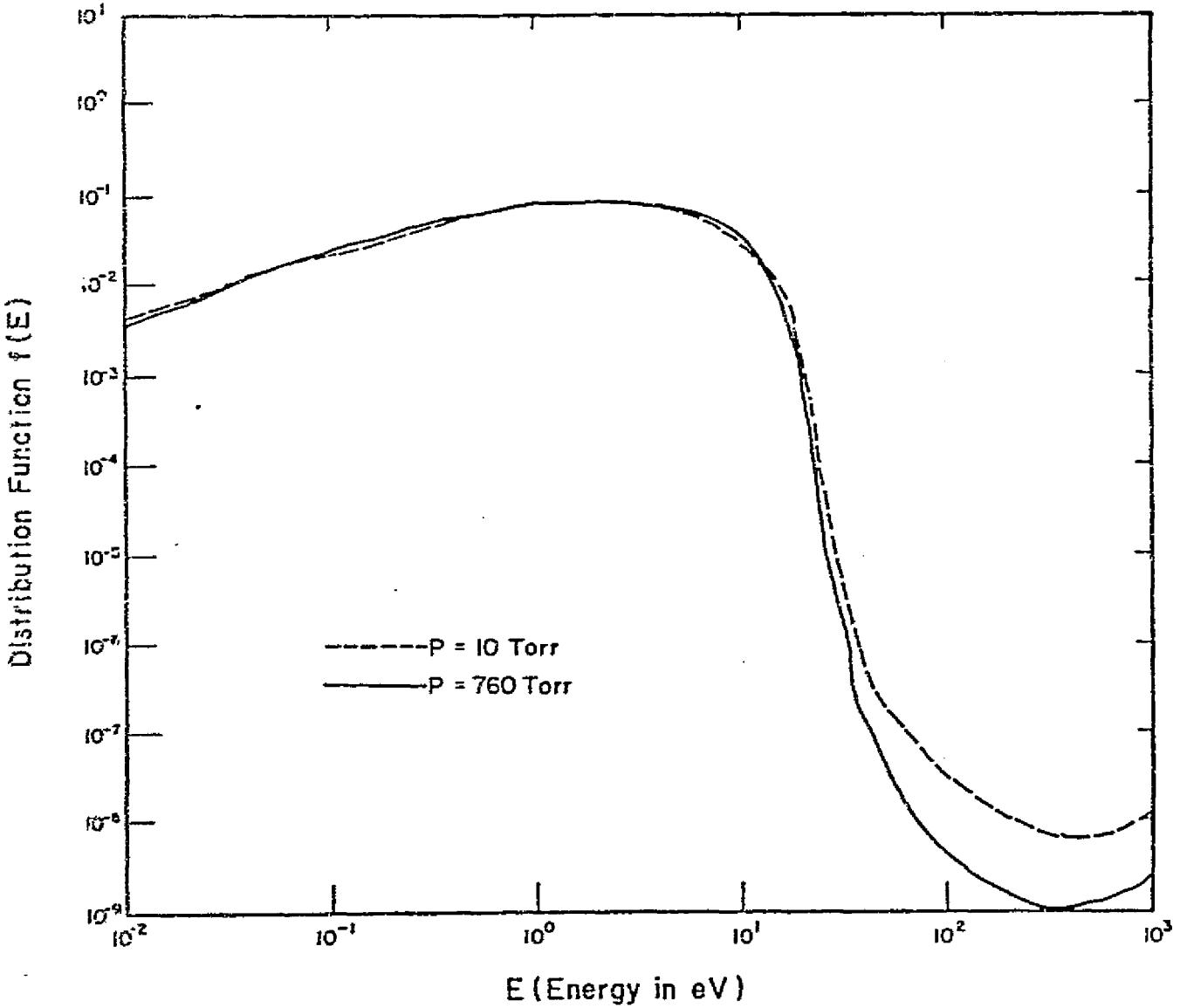


Figure 22. The electron energy distribution for He, with an electric field at $S_o = 10^{16}$, $\mathcal{E}_o = 1000$ eV.

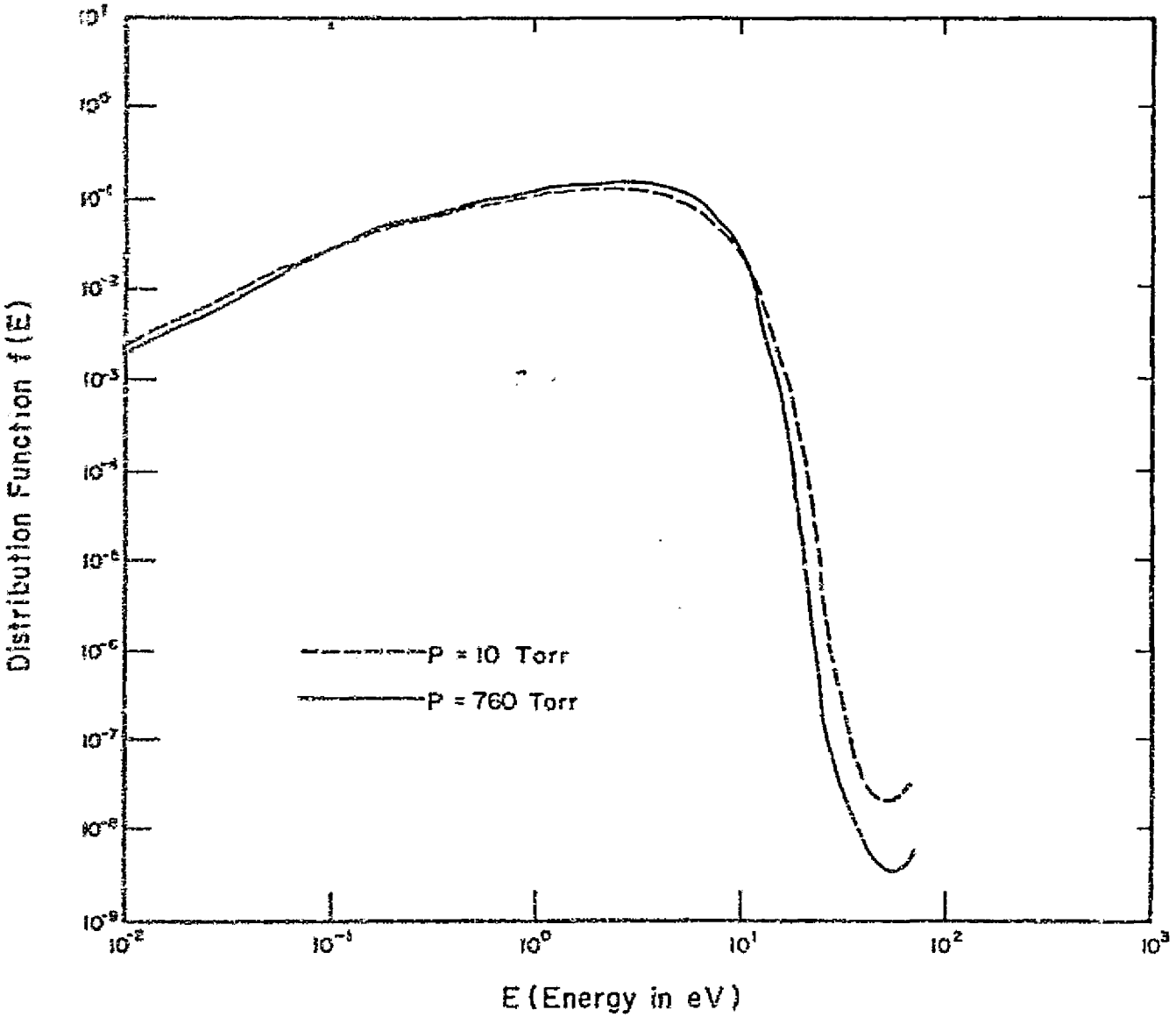


Figure 23. The electron energy distributions for He, with an electric field at $S_o = 10^{16}$, $E_o = 70$ eV.

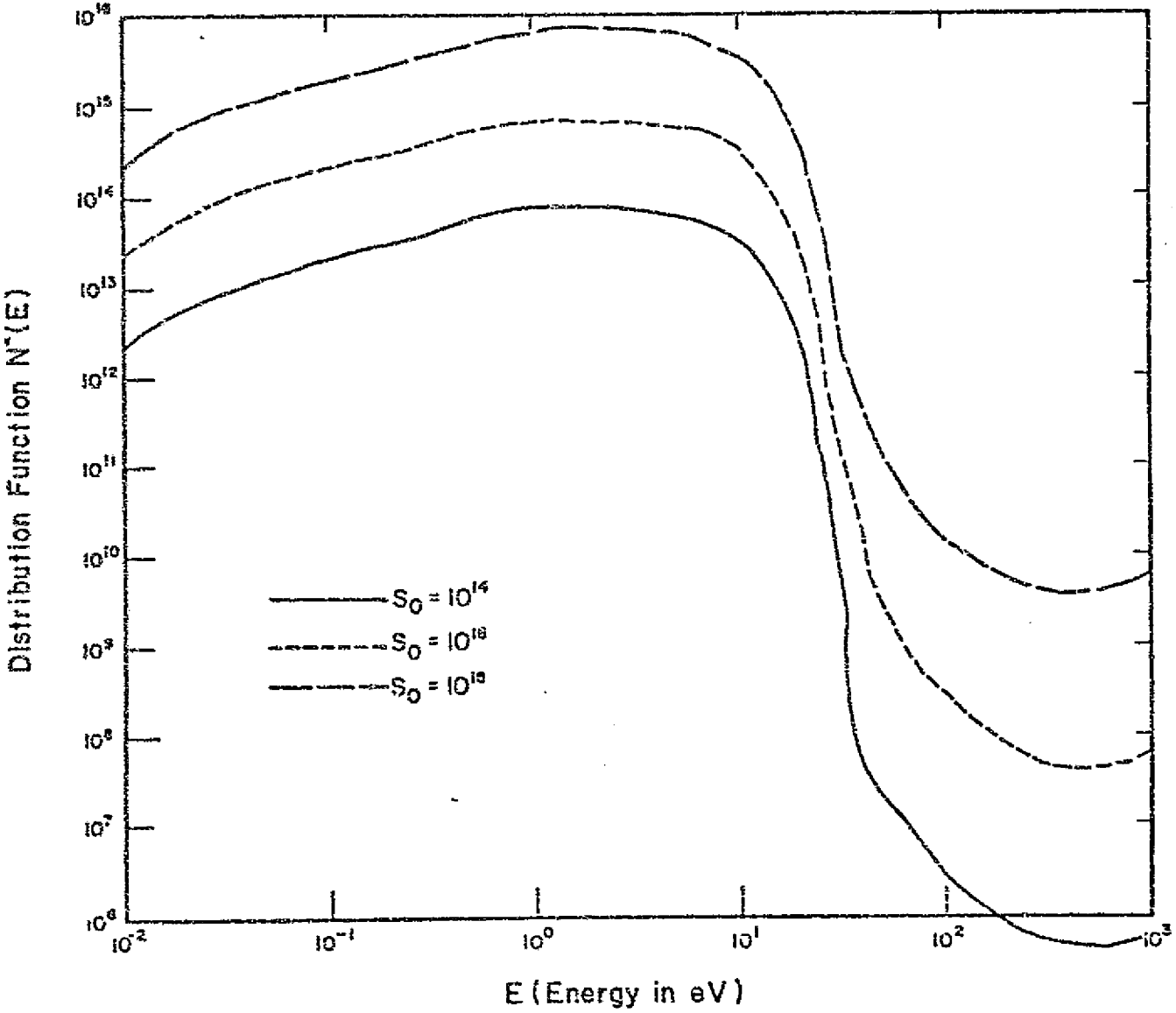


Figure 24. The unnormalized electron energy distribution for He at $E_{f/p}$ = 10, p = 10 torr, E_0 = 1000 eV.

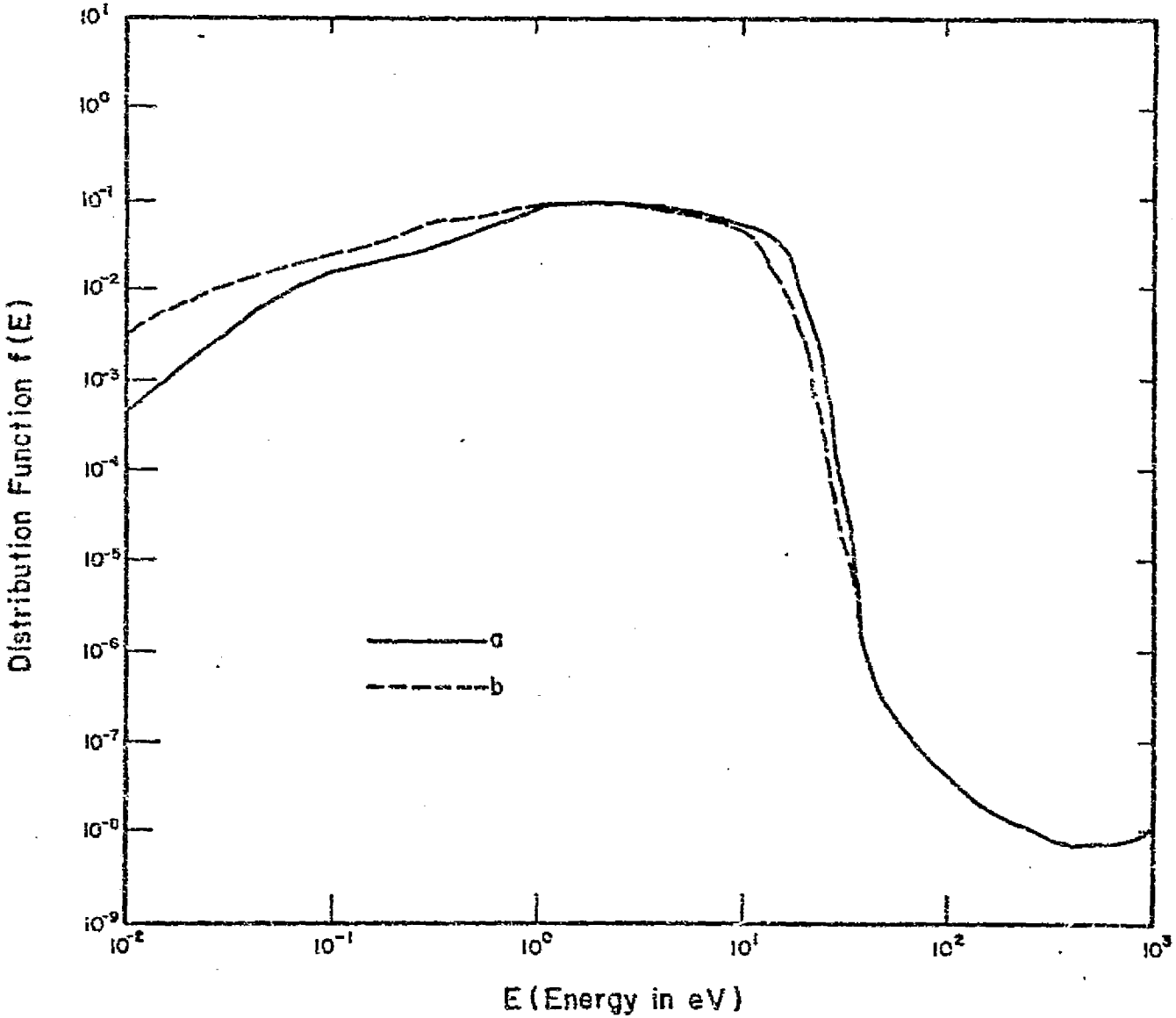


Figure 25. The effect of recombination on the electron energy distribution
a) with normal recombination cross section x 100
b) with normal recombination cross section x 1.

and $S_o = 10^{14}$, respectively.

In order to show the effect of recombination, a normal case is compared in Figure 25 to a calculation where the absorption cross section was arbitrarily increased 100 times. It is observed that the major effect comes in at low-energy region. More and more electrons recombine as they thermalize and hence, the lower energy region of the distribution is depleted. Since recombination does not occur at high energies, this region is not affected.

The calculated major parameter values for the system with an electric field are summarized in Table 2. As expected on physical grounds the values of the average energies for this case are much higher than the corresponding values with no electric field (compare to Table 1). However, other general trends are the same, e.g. the value of the average energies are slightly higher for higher source rate and/or higher source energy, but are lower for higher pressures.

For comparison the average energy (7.7 eV) found by Smit for an electric field and $E_{f/p} = 10$ but with no source present is included in Table 2. It is in general higher than the present values. The reason is not entirely clear. At first thought the additional high-energy source electrons might be expected to raise the average energy. However, as found by T. Ganley (37), the presence of high-energy electrons to produce ionization allows the electrons in the Maxwellian region to decrease in energy, and as a result the overall average may actually decrease.

Table 2. Calculated Parameter Values for Plasma with Applied Electric Field ($E_{f/p} = 10$)

* P (torr)	E_o (eV)	S_{o_3} (#/cm^3-sec)	$\bar{E}$ (eV)	N^+ (#)	Modified W-value range
10	1000	10^{18}	8.1	0.24×10^{15}	35 ~ 44
10	1000	10^{16}	7.5	0.21×10^{14}	34 ~ 54
10	1000	10^{14}	7.3	0.24×10^{13}	37.5 ~ 60
760	1000	10^{18}	7.0	0.23×10^{17}	37 ~ 50
760	1000	10^{16}	6.6	0.204×10^{16}	30 ~ 51
760	1000	10^{14}	5.4	0.24×10^{15}	34 ~ 57
10	70	10^{18}	5.86	0.21×10^{15}	31 ~ 35
10	70	10^{16}	5.18	0.21×10^{14}	34 ~ 50
10	70	10^{14}	4.46	0.213×10^{13}	31 ~ 56
760	70	10^{18}	5.7	0.284×10^{16}	35 ~ 46
760	70	10^{16}	5.0	0.26×10^{15}	39 ~ 48
760	70	10^{14}	4.3	0.24×10^{14}	32 ~ 49
Smit's distribution					
		0	7.7	---	---

*p: pressure; E_o = source energy; S_o: source rate; $\bar{E}$: average energy; N^+: number of positive ions.

The values of N^{+} are roughly proportional to the square root of the source rate (recall Equation 33) and are independent of the pressure and source energy.

Strictly speaking, the definition of the W-value for the field case should include an allowance for energy gained to electrons in the electric field which then result in ionization. There is no way to include this energy transfer in the present case, so the values listed in Table 2 are based on the ion pair production divided by the energy input with the high-energy source electron alone (no field contribution). This is, then the same formulation (Eq. 65) as that of no field case, and we lable the result as the "effective" or "modified" W-value. Again these values fluctuate considerably, so the ranges observed in the last 10 time cycles are given, As would be expected from the neglect of the field contribution, the W-values in this table are in general smaller than that of no field case. As in the earlier calculation, no general trend can be observed, as so far as the dependence of the parameters (E_o, S_o, p) are concerned, and this is consistent with the earlier observations and the literature which indicate that the W-values are fairly insensitive to these parameters.

We have concluded so far that, the distribution functions for non-field case ($E_f = 0$) are more or less assume a Maxwellian distribution in the low-energy region for source rates below 10^{18}. However, when an electric field

is added, the distribution shifts to that found by Smit in the low-energy region. This is, in effect, the well-known Druyvesteyn distribution. It is significant that the addition of a source adds a tail to this distribution, but it does not drastically change the low-energy region. By intuition, it might be expected that Smit's distribution (with electric field or $E_f \neq 0$) would change back to a Maxwellian distribution as the electric field approaches zero. In order to display the transition region, the distribution functions for smaller electric field ($E_{f/p} = 1$) are plotted in Figure 26 together with the calculated distribution functions at $E_{f/p} = 10$ and $E_f = 0$. The high-energy tail of the distribution curve are not significantly affected by reducing the electric field E_f, but the distribution function in the low energy region starts to approach a Maxwellian distribution as expected.

D. Conclusion on Monte Carlo Results

Based on the Monte Carlo solutions we obtained so far, we can make the following comments.

1. For the normal range of source rate ($S \leq 10^{18}$), where the neglect of electron-electron interactions remains valid, the Maxwellian plus 1/E distribution is a good initial approximate distribution for the no electric field case. Likewise the Smit (or Druyvesteyn) plus 1/E distribution is a good approximate initial distribution to be used for the system with an electric field present.

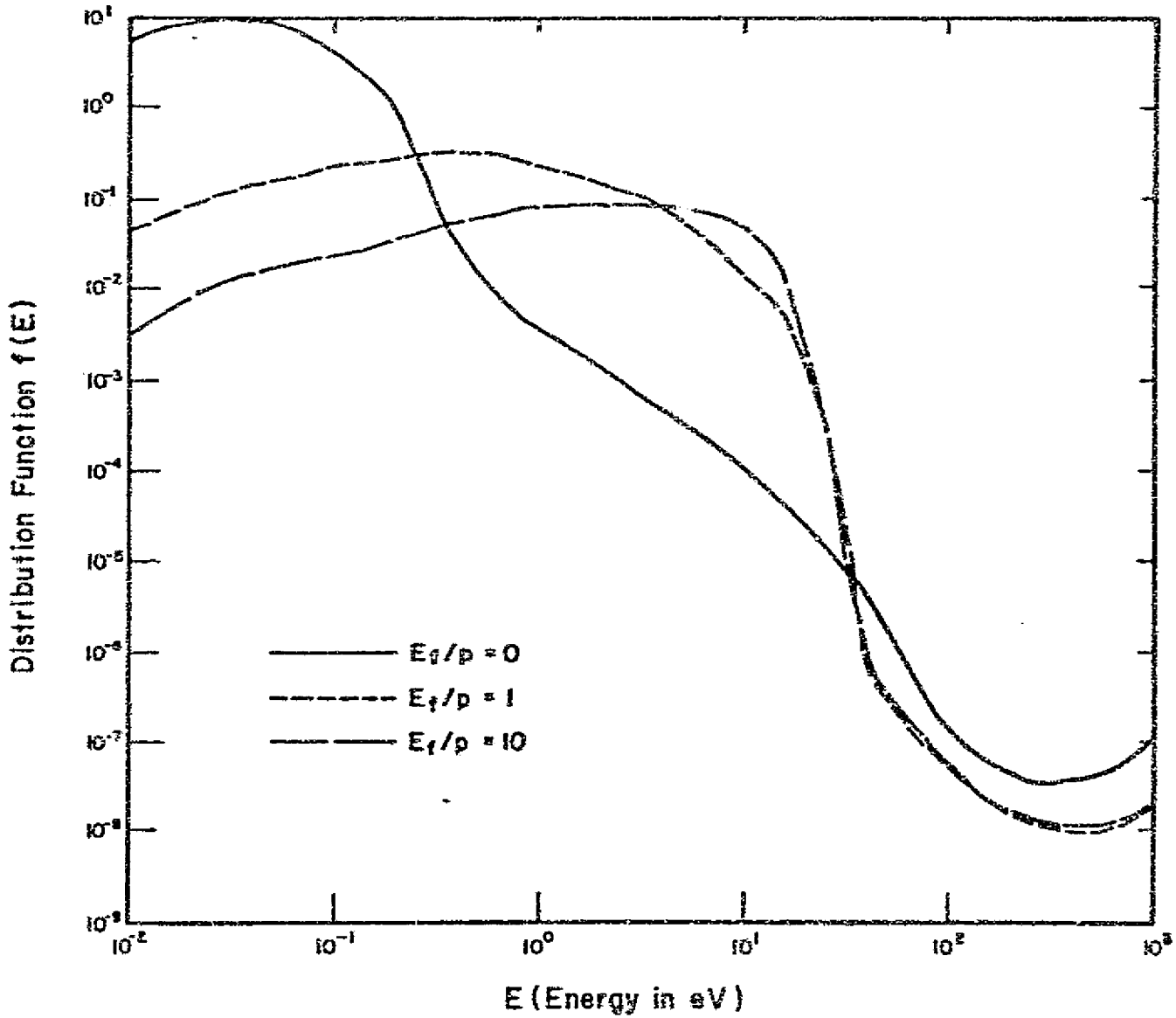


Figure 26. The electron energy distribution for He at p = 10 torr, $S_o = 10^{16}$ for $E_{f/p} = 0$, $E_{f/p} = 1$ and $E_{f/p} = 10$.

2. The dependence of the distribution functions (with or without electric field) upon the pressure and source rate only shows up in the high-energy region, where the distribution functions have higher tails for higher source rate, and have smaller values for higher pressures.

3. The range of variations observed in the W-value calculation reflects statistical fluctuations which limit the calculational accuracy of the absolute numerical W-values. Still the results demonstrate that the W-value is more or less independent or insensitive to the changes of pressure, source rate and source energy.

4. Due to a tight budget, the primary emphasis was on increasing the computation efficiency and reducing the computation time to a possible minimum. Thus most of the calculations were carried out with 2000 to 5000 actual particles and for any particular set of parameter values, and the results were generally obtained within 10 minutes of (IBM 360/75) computer time. Compared with hours of computer time and 5 to 10 times more particles used by others for the similar calculations (e.g., see References 3, 4, and 9) the gain in computation efficiency seems to be significant. Of course, a complete comparison is not possible since all of these calculations achieved different accuracies in the final results.

5. The present results also show close agreement with the analytic solutions of Lo and Miley (32,33) on the

following points:

a. The shape of high-energy portion of the distribution curves are in reasonable agreement as shown in Figure 9.

b. The high-energy part of the distribution function is not changed by the presence of an electric field for field values $E_{\ell/p} \leq 10$.

c. With an electric field, the low-energy part of the distribution function agrees with Smit's result; and this part is not affected appreciably by source rates $< 10^{10}$ particles/cm^3-sec.

CHAPTER VI

CONCLUSIONS

A. Prelude

The application of Monte Carlo techniques to solve particle transport problems is of fundamental interest in many fields of physics and engineering; especially when analytic and/or numerical solutions of the basic governing equations are too complex to be practical. Yet direct application of Monte Carlo techniques to simulate individual interactions is sometimes prohibitively costly because of the large amount of computational time required. Thus techniques for improving the efficiency are required. Various existing variance reduction techniques can be used, but they are problem dependent. Thus techniques that are efficient for one type of problem might not be as effective for other types of problems. Thus such techniques must be judiciously selected and modified for the particular problem under consideration. One basic underlying principle which applies to any Monte Carlo calculation is: Apply as much known information as one can (given in analytic and/or numerical form) to reduce the uncertainties of the problem. Whenever possible Monte Carlo experiments should be checked and replaced by exact theoretical analysis to reduce uncertainties.

The Monte Carlo experimentalist has to exercise ingenuity in distorting and modifying problems in the pursuit

of variance-reduction techniques. Although Monte Carlo methods are general devices, still a great deal of work depends upon the individual's originality to create special methods or numerical schemes to suit their needs. Unlike the physical experimentalist, the mathematical experimentalist using Monte Carlo Methods has the advantage that his experimental material consists of mathematical objects which can be distorted, controlled, and modified more easily.

The greatest successes of the Monte Carlo method have arisen where the basic mathematical problem itself consists of the investigation of some random processes. However, there are exceptions involving deterministic problem-solution of boundry value problems and partial differential equations. The solutions of these problems are closely connected with the characteristics of certain random diffusion type processes (or can be converted to such type of processes), therefore these problems are reduced to the modeling of such processes. Thus the model and the special techniques developed in present research are quite general in scope, and it should be possible to apply them to many classes of Monte Carlo calculations with a certain amount of modification.

B. Summary of Present Work

In this research study, we have devised a general mathematical model for the particle transport or diffusion

type plasma problems which have inherent nonlinear properties due to recombination. Many special variance reduction techniques are incorporated into the system. The model can be used for calculating both time-dependent and steady-state nonlinear transport problems provided that suitable methods exist to calculate, predict, and correct the important parameters which dominate the nonlinearities. A wide range of nonlinear problems can be formulated and handled in this way. The principle idea, the piecewise linearized predict-correct model, is a general, simple, and efficient approach for solving nonlinear problems. In principle, most nonlinear problems can be cast in terms of the basic ideas and algorithms of present work. This represents an extension of earlier works of Musha and Itoh (8), Thomas (9), and Fleck Jr. (11). As for the amount of computation time and efficiency involved, the present study has made significant improvements. Some of the improvements are a natural consequence of the large storage space available in modern computer systems. Additional improvements come from a few simple, but effective, special techniques incorporated into the present model. As described in Chapter II, these include:

1) The initial distribution - straightforward particle tracing procedures would require following every simulated particle from its birth (from source) with initial energy E_o at $t = t_o$ to the end of its life time history. This type of simulation would require hundreds

of cycle times to achieve a final steady state condition. Instead, we start with an apprixmately steady-state system by guessing an initial particle energy distribution. Then we simulate the system by tracing particles starting with energy corresponding to this input distribution. This is equivalent to skipping over the initial transient time cycles and greatly reduces the computation time and increases the calculation efficiency.

2) The application of the negatively-correlated variates (antithetic variates) technique greatly reduces statistical fluctuations with a minimum number of simulating particles (machine particles).

3) The introduction of a weighting scheme and an artificial game of chance (using weighted cross sections) contributed some reduction in calculation time and resolved some practical difficulties.

4) For the system with an applied electric frield, the exact formulation for a parabolic flight path was used to advance the particles in larger steps than would have been possible using the small-step linear approximation employed by Musha and Itoh (8) and also by Thomas (9). This improves the efficiency appreciably.

Throughout the entire calculation, one basic balance equation (Eq. 30) played a very important role. It merely states that at steady state, the source is balanced by losses.

By means of this equation, we predict the nonlinear parameter values and provide checks for convergence of the solution.

The need for large amounts of storage space for Monte Carlo calculations no longer creates serious problems or drawbacks on modern large computer systems, such as was the case as recent as seven or ten years ago. In addition, techniques exist for further reducing the storage requirement. In the present study, we suggest a technique to eliminate major storage requirements, and use only a small amount of storage as a temporary buffer storage (described later).

In other types of numerical solutions, e.g., in finite difference solutions of the diffusion equation, the step size Δt is restricted by stability requirements. However, in this present model, Δt is only restricted by the degree and amount of nonlinearities present in the system. This restriction is far less stringent.

C. Practical Limitations and the Cures

It was necessary to solve several difficulties in the present simulation study in order to provide practical solutions. We shall describe them item by item.

1) The electron energy distribution is the quantity of major interest. The difference between the magnitudes of its maximum and minimum non-zero values is of the order 10^9. This is due to high density in the low-energy region (thermal energy) and the relatively low density

in the high-energy region (near the source energy). It is unrealistic to hope to produce reasonable numbers of particles in the low-density (high-energy) region and at the same time have 10^9 times as many in the low-energy region. This problem is resolved by combining the following techniques:

a) Russian Roulette and splitting.

b) Weight factors which produce fractional particles.

c) Logarithmic energy intervals. The number of particles in each energy class interval ΔE_i is directly proportional to $f(E_i)\cdot\Delta E_i$ ($f(E_i)$: number density at E_i). By using logarithmic energy intervals, we have small intervals at low energy region and large intervals at high energy so that the product $f(E_i)\cdot\Delta E_i$ assumes a practical magnitude.

2) The total number of simulating particles (machine particles) N_T is limited by the storage requirement and the economy of the calculation (computing time is proportional to N_T). We have to set an upper limit (Nmax) on the total number of simulating particles in any time cycle. During each time cycle of duration Δt, N_s source particles are introduced into the system, and N_z secondary particles are produced (by ionization). At the same time N_{es} particles escape from the system, and N_{Ab} particles are lost due to recombination.

There are two conditions to be satisfied at all times:

a) $N_c + N_z + N_s \leq N_T \leq N_{max}$ (67a)

b) $N_z + N_s \simeq N_{es} + N_{AB}$ (67b)

where N_c: the number of census particles (background particles).

Condition a) insures that total number of particles in the system is always under the limit (Nmax). Condition b) indicates that a particle balance should be satisfied at steady state.

In the program, there are two equivalent problems to be considered. One is the balance of the machine particles [corresponding to condition (a)], and the other is the balance of simulated particles [weighted particles, corresponding to condition (b)]. Both conditions must be satisfied simultaneously. Condition (a) allows a certain amount of leeway since the number of particles that survive the integration cycle (time cycle) and hence require storage will be less than $N_c + N_z + N_s$. Here N_s is chosen to be an input number, the weights of these particles are determined by the source rate. If condition (a) can not be met, N_s is taken to be

$$N_s = N_{max} - N_c - N_z - JZ - 1 \quad (68)$$

where JZ is the number of spatial zones. It has been shown (12) that Equation 68 insures the stabilization of census population without overly reducing the number of source particles.

The proper selection of the energy dividing line (EDL) for Russian Roulette and splitting can insure the satisfaction of condition (b). Also as mentioned in earlier chapters, the balance factor (BF) is used to speed up convergence and force condition (b) to be satisfied sooner.

3) In any Monte Carlo calculation, finite samples are used to estimate or simulate the behavior of a large parent population. This is the main source of statistical fluctuations (variance or uncertainty). In the present simulating system, less than 5000 particles were used to simulate approximately 10^{14} physical particles. This creates the problem of matching the values of input and estimated output quantities. Due to the huge scaling factor ($10^{14}/5000$), a small amount of statistical fluctuation or estimation error in the estimated value will be amplified about 10^{11} times. To avoid this, we use the estimated (or equivalent) input quantities which carry the same order of magnitude of statistical fluctuation and weight.

4) There are several reasons that make the logarithmic energy scale the logical choice for the group intervals in the energy axis of phase space. These include:
   a) the wide energy span (0 to 1500 eV)
   b) the relatively high number density at low energies combined with a low rate and small amount of energy

exchange due to collision processes. In the high-energy region, the situation is just the opposite.

c) to produce reasonable number of particles in each energy interval [for the reason mentioned in the first section (II-B-1)]. The size and number of intervals are mainly dictated by the range of data (energy range in this case) and number of observations available. Excessive fragmentation of the data may produce many intervals with few occupants. On the other hand, insufficient divisions may obscure important dispersions. Thus an optimum number of intervals will be a compromise between these extremes.

D. Future Extensions

The Monte Carlo methods are applicable to the most widely diverse branches of computational mathematics including particle physics (neutron transport for instance), operations research problems such as the investigation of servicing processes, modelling the processes of information transmission in communication theory, the evaluation of definite integrals, the solution of partial differential equations (e.g. boundry-value problems); the solution of systems of linear equations - inversion of matrices, and many other applications. Problems handled by such methods are in general of two types, the probabilistic and the deterministic, according to whether or not they are directly concerned with the behavior and outcome of random processes.

Their basis lies in simulating statistical experiments by means of computational techniques. Based on the underlying principles described earlier, various variance reduction techniques can be applied. The present work serves as an example and guide for using the important new techniques such as correlated sampling.

1) The code developed can readily be used to calculate electron energy distribution functions for other gaseous media by merely changing the cross sections input. The model can also be modified to include the electron-electron and electron-ion interactions although this adds more nonlinearities into the system. This would allow calculation for higher source rates and cases where the fractional ionization is large.

2) As for the complex transport problems encountered in areas such as plasmas physics and astrophysics, the model and techniques can readily be extended to the problems with more complex geometry and/or boundry conditions, with inhomogeneous media and multi-dimensions. Also many kinds of nonlinear features can be studied in a realistic fashion.

3) More elaborate error checking and correction procedure (based on the known physical phenomena or theoretical formulations) such as the correction matrix used by Berger (9) should be developed to treat the combined effect of statistical fluctuations and other types of

uncertainties. This is essential for generating good results for Monte Carlo calculation.

4) More elaborate algorithms or techniques for playing the artificial games of chance by means of weighted cross sections would be a good direction for future research. This must be based on the investigator's intuition and experience, but still it is a promising approach for increasing the efficiency.

5) The present model and techniques can be also implemented on smaller computer systems such as PDP-11 or other similar systems where large amount of fast storage space are not available. The following techniques could be used to process the calculation without storing all the particle data (namely E_i, u_i, z_i, Wt_i Ψ_i). At the beginning of each time cycle, we generate a small number (some reasonable amount, $\simeq$ 100 for instance) of particles. The corresponding particle data can be generated from the initial input distribution or from last time cycle. After these particles have been processed, only the accumulated information is stored. A new group of particles come into the system by generating a new set of data, based on the same distributions, which are stored in the same temporary storage space. The previous particle data is destroyed after they have been processed. The same process is repeated until all the necessary prespecified number of particles have been

processed, and the new distributions are obtained. This eliminates the necessity of storing all the particle data at each time cycle, and this eliminates the large arrays that demand the major amounts of storage space. Due to the random nature of the processing algorithms, and the Markovian behavior of the time cycle process (these time cycle processes are a kind of Markov process), this storage saving algorithm can accomplish the desired results.

E. Concluding Remarks

In developing any Monte Carlo models and performing calculations, as mentioned in Chapter V the systematic errors and outright blunders must be detected and removed by testing and comparing with known results.

The various variance reduction techniques are used to increase the relative computation efficiency, reduce inherent random errors or statistical fluctuations. For better calculational results, more samples are favorable even though not economic. In any Monte Carlo calculation there exists a minimum number of samples below which the results or statistics are no longer valid and dependable regardless of the variance reduction schemes employed. In the other words, variance reduction techniques help to reduce statistical fluctuations, but they can not be used to remove all the fluctuations caused by small samples.

The minimum sample size is, of course, problem dependent. There are no definite rules or formula for it, thus a cut and try method is necessary until the calculated variance is acceptable.

Monte Carlo calculations are perhaps best in the preliminary stages where they help to give a general idea of the situation and give hints or feelings for trends. If more accurate results are necessary, Monte Carlo calculations may not always be practical good method. In any case, the Monte Carlo method is an important tool in the inferential analysis of the mathematical-physics problems, but it should be viewed as supplementing, rather than replacing, analytic and other numerical methods.

LIST OF REFERENCES

1. R. W. L. Thomas and W. R. L. Thomas. "Monte Carlo Simulation of Electrical Discharges in Gases," J. Phys. B (Atom. Molec. Phys.) Ser. 2, Vol. 2, 562, (1969).

2. Tomizo Itoh and Toshimitsu Musha. "Monte Carlo Calculations of Motion of Electrons in Helium," J. Phys. Soc. Japan, Vol. 15, 9, 1675 (1960).

3. J. A. Fleck, Jr. "The Calculation of Nonlinear Radiation Transport by a Monte Carlo Method," Methods in Computational Physics, Vol. 1, 43. New York: Academic Press, (1963).

4. J. A. Fleck, Jr. and J. D. Cummings. "An Implicit Monte Carlo Scheme for Calculating Time and Frequency Dependent Nonlinear Radiation Transport," UCRL - 72924 (1970).

5. G. H. Miley, J. T. Verdeyen, T. Ganley, J. Guyot and P. Thies. "Pumping and Enhancement of Gas Lasers via Ion Beams," Symp. on Electron, Ion & Laser Beam Technology, University of Colorado, May, 1971.

6. J. C. Guyot, G. H. Miley, J. T. Verdeyen and T. Ganley. "On Gas Laser Pumping Via Nuclear Radiations," Symp. on Research on Uranium Plasmas and Their Technological Applications, University of Florida, 1970.

7. J. C. Guyot, G. H. Miley and J. T. Verdeyen. "Application of a Two-Region Heavy-Charged Particles Model to Noble-gas Plasmas Induced by Nuclear Radiations," Intern. Rept. Nuclear Engineering Program, University of Illinois (1970).

8. L. Herwig. "Summary of Preliminary Studies Concerning Nuclear Pumping of Gas Laser Systems," Report C-110053-5, United Aircraft Corp., East Hartford, Connecticut (1964).

9. M. J. Berger. "Monte Carlo Calculation of the Penetration and Diffusion of Fast Charged Particles," Methods in Computational Physics, Vol. 1, 135-213. New York: Academic Press, 1963.

10. A. E. D. Heylen and T. J. Lewis. "Electron Energy Distribution Functions and Transport Coefficients for the Rare Gases," Proc. R. Soc. A., Vol. 271, 531 (1963).

11. J. A. Smit. "Berechnung der Geschwindigkeits Verteilung der Elektronen bei Gasentladungen in Helium," Physica, Vol. 3, 543 (1936).

12. J. M. Hammersley and D. C. Handscomb. Monte Carlo Methods. New York: Wiley, 1964.

13. J. M. Hammersley and K. W. Morton. "A New Monte Carlo Technique: Antithetic Variates," Proc. Camb. phil. Soc., Vol. 52,449 (1956).

14. J. M. Hammersley and J. G. Mauldon. "General Principles of Antithetic Variates," Proc. Camb. phil. Soc., Vol. 52, 476 (1956).

15. Y. A. Schreider. The Monte Carlo Method. London: Pergamon Press, 1966.

16. E. D. Cashwell and C. J. Everett. A Practical Manual on the Monte Carlo Method for Random Walk Problems. New York: Pergamon Press, 1959.

17. M. A. Uman. Introduction to Plasma Physics. New York: McGraw Hill, 1964.

18. L. G. Parratt. Probability and Experimental Errors in Science. New York: Wiley, 1961.

19. T. D. Sterling and S. V. Pollack. Introduction to Statistical Data Processing. New Jersey: Prentice-Hall, 1968.

20. S. Karlin and W. J. Studden. Tchebycheff Systems: With Applications in Analysis and Statistics. New York: Wiley, 1966.

21. L. L. House and L. W. Avery. "The Monte Carlo Technique Applied to Radiative Transfer," J. Quant. Spectrosc. Radiat. Transfer, Vol. 9, 1579 (1969).

22. H. Kahn. "Use of Different Monte Carlo Sampling Techniques," Symposium on Monte Carlo Methods. New York: Wiley, 1956.

23. R. J. Cicerone and S. A. Bowhill. "Monte Carlo and Thomson-scatter plasma-line studies of Ionospheric Photoelectrons," Aeronomy Rept., No. 39, University of Illinois, Urbana, Illinois (1970).

24. A. Nordsieck and B. L. Hicks. "Monte Carlo Evaluation of the Boltzmann Collision Integral," CSL Report R-307, University of Illinois (1966).

25. S. M. Yen and B. L. Hick. "On the Accuracy of Approximate Solutions of the Boltzmann Equation," CSL Report R-308, University of Illinois, Urbana, Illinois (1966).

26. M. Y. Youakim, C. H. Liu and K. C. Yeh. "Generation of Random Hypersurfaces," Intern. Rept., Ionosphere Radio Laboratory, University of Illinois, Urbana, Illinois (1970).

27. M. Rich and S. S. Blackman. "A Method for Eulerian Fluid Dynamics," LAMS-2826, UC-32, Mathematics and Computers (1963).

28. W. D. Little. "Hybrid Computer Solutions of Partial Differential Equations by Monte Carlo Methods," Proc. FJCC, 181 (1966).

29. L. B. Miller and G. H. Miley. "Some Monte Carlo Techniques for the Calculation of Doppler Coefficients," Nuclear Science and Engineering, Vol. 40, 438 (1970).

30. C. K. Birdsall and D. Fuss. "Cloud-In-Cell Computer Experiments in Two and Three Dimensions," Proc. APS Topical Conference on Numerical Simulation of Plasma. Los Alamos Scientific Laboratory, University of California (1968).

31. M. Goodrich. "Electron Scattering in Helium," Physical Review, Vol. 54, 259 (1937).

32. G. H. Miley and E. H. Lo. "Calculation of Electron Energy Flux Distribution in Noble Gases," To be published in Proc. 18th Annual Meeting of ANS, June, 1972.

33. R. H. Lo. Department of Nuclear Engineering, University of Illinois, Private Communication, April, 1972.

34. W. Jesse and J. Sadouski. "Ionization in Pure Gas and the Average Energy to Make an Ion Pair for Alpha and Beta Particles," Phys. Rev., Vol. 97, 1665 (1955).

35. S. C. Brown. Basic Data of Plasma Physics. Cambridge, Massachusetts: The M.I.T. Press, 1966.

36. T. Holstein. "Energy Distribution of Electrons in High Frequency Gas Discharges," Phys. Rev. Vol. 70, 5, 367 (1946).

37. J. T. Ganley. Effect of a Heavy Particle External Ionization Source on CO_2 Laser Discharges, Ph.D. Thesis, University of Illinois, 1972.

APPENDIX

```
*IBFTRAN IV G LEVEL 18          MAIN          DATE = 72126      09/31/02
      C MASTER PROGRAM OF MONTE CARLO SIMULATION OF NONLINEAR ALFVEN PLASMAS INCLUDING PAR
      C TICLES INDUCED PLASMA ELECTRON ENERGY DISTRIBUTION STUDY
      C DEFINITIONS AND TERMINOLOGY
      C DI DIAMETER OF THE CYLINDER THE LENGTH OF THE FINITE REGION
      C EO:INITIAL ENERGY OF THE SOURCE PARTICLES SOURCE: SOURCE RATE
      C JZ:TOTAL NUMBER OF ZONES   JE: TOTAL NUMBER OF ENERGY RANGES
      C IAS: ARRAY FOR STORING THE INDEX OF ABSORBED PARTICLES NEXT THE NO. OF THEM
      C DB: DISTANCE TO BOUNDRY  DCOL: DISTANCE TO COLLISION  DCEN: DISTANCE TO CENSUS
      C DZ: ZONE INTERVAL
      C ARRAYS E, Z, U  STORING ENERGY, DISTANCE, AND DIRECTION OF PARTICLES
      C NTP: TOTAL NUMBER OF PARTICLES TO BE PROCESSED  EACH CYCLE
      C NT: TOTAL NUMBER OF TIME CYCLES
      C NS: NUMBER OF SOURCE PARTICLES PER UNIT TIME*DISTANCE  OR SOURCE RATE
      C NI:N: NUMBER OF IONIZATION PROCESS OCCURRED EACH TIME CYCLE
0001                REAL    NE(55), NEZ( 55,10),EZ(11),SX(3),EE(55),  CEZ(11),EKZ(55
     1,10) ,F10(60),NREZ(11),EZI(11),NTPM
0002                DIMENSION  ENE(55), ENREZ(11), NKZI(11),CNE(55), CNREZ(11),TNEY(55
0003                COMMON D, E(5100),Z(5100), U(5100) ,IAS(500),IES(600),NRR,IES(600)
     X,WT(5100),SMF
0004                COMMON /AREA1/ NIZN, NOZN, NAS, DIZN(500),NSE, NNET, NEXC, ETAS  ,
     XWNNET, WNEXC, WNIZN, ELS,WNAS ,WNSE
0005                COMMON /AREA2/ CL,HUO,EI,ZIS,SC,ANEAN,PDPO,PI,EO,NION,JE,JZ,NASW1,
     1NSEN ,AKT, AIKT ,ANI ,ANC1,EDI,PDMUO, PD2A,K1, K2, K3, WN2,EDZ,ED3
0006                COMMON /AREA3/ KSEL,XELC,XRCB,XNET,XEXC,XION,STOL,KELC
0007                COMMON /AREA4/  CONST,DEL2, ELT,IE, DI1, DI2, DI3 ,DELT
0008                COMMON /A5/ EG(55), EDE(55), EBD(55)
0009                COMMON /A6/ ESE(50) , NUP ,NONR,NON, WSE(50)
0010                REAL  MUO, NION,NO
      C STATEMENT FUNCTION FOR EVALUATING  MAXWELLIAN AND RELATED FCNS
0011                FMI(E,ENEM)=EXP(-E*AIKT)*SQRT(E)*ENEM
0012                DATA  TMS,NT,DTOT,ETOT,N(N,NIZNY,NAST/0.0,1,0.0,0.0,0,0,0,0,0/
0013                DATA WSVLE,CNSIZ,CNSS,CNES,CNAS,CNNET,CNEXC,CESAVG,CEASG/9*0.0/
0014                DATA CNE/ 55*0.0/, CEZ/11*0.0/, CNREZ/11*0.0/,TNE/55*0.0/,EKZ/550*0
     1.0/,DSAV$/0.0/
0015                DATA NEZ/ 550*0./,NE/55*0./,EZ/11*0./,NREZ/11*0./,EE/55*0.0/
0016                READ(5,1) NTAPE
0017                READ(5,151) K1, K2, NTC, NTP,NTW,KPR, KELC ,JZ, IM,NCON, NDPN,NYUP
     X, MC ,IUSN, NUST, NBF
0018                READ(5,152) EDI ,NO , EO,ANSN, 0, EDP, PDPO, SR
0019                READ(5,153) FID
0020              1 FORMAT ( I10)
0021            151 FORMAT( 16I5)
0022            152 FORMAT( 8F10.4)
0023            153 FORMAT( 8E10.3)
0024                DELE=1.0
0025                DEL1=1.0/DELE
0026                DEL2=2.0
0027                AKT=0.05
0028                AIKT= 1.0/AKT
0029                PI=3.141593
0030                EMASS1=1.60203E-16/9.108
0031                K3=K1*K2
0032                WN2=1.0/K2
```

```
FORTRAN IV G LEVEL  16                  MAIN                    DATE = 72126          05/37/02

0033                CJO=0.0
0034                B1=24.46
0035                EF=ROP*PDPO
0036                EIS=21.4
0037                C1=2.0*0.000549/4.003873
0038                ETV= SQRT(2.0*1.60203/9.108)*1.0E+08
0039                VTE= 1.0/(ETV*ETV)
0040                CALL  UNDERZ('OPP')
0041                PD2O=PDPO*26.0
0042                HUO=(1.0E-09)/ETV
0043                CMUO=1.0E-09
0044                CMUPOP=CMUO*PDPO
0045                IF (PDPO.GT.10.0)  CMUPOP=CMUO*10.0
0046                PDMUO=CMUPOP/ETV
0047                ELS=1.0
0048                NUP=0
0049                NDN=0
0050                NDNR=0
0051                INO=0
0052                NRR=0
0053                NAS=0
0054                NES=0
0055                NSE=0
0056                NOZN=0
0057                NIZN=0
0058                NIN=0
0059                LK=0
0060                INIV=999999
0061                NNET=0
0062                NEXC=0
0063                ETAS=0.0
0064                ETEP=0.0
0065                MTPS=0
0066                KK=0
0067                NSTP=0
0068                NCOL=0
0069                WNNET=0.0
0070                WNIZN=0.0
0071                WNEXC=0.0
0072                WNAS=0.0
0073                WNES=0.0
0074                WNSE=0.0
0075                CLOSS=0.0
0076                CSFZ=0.0
0077                CNIO2=0.0
0078                CSFL=0.0
0079                CETAS=0.0
0080                CETEP=0.0
0081                DTAVS=0.0
0082                JE=48
0083                IE=JE
0084                DEN=1.54E+16*PDPO
0085                DETZ=D/JZ
```

```
FORTRAN IV G LEVEL  18                    MAIN                    DATE = 72126      05/31/02

0086                DZIV= 1.0/DELZ
0087                CONST=1.0/ALOG(2.0)
0088                ED2=0.44Y513
0089                ED3=19.8
0090                PM=ED2*AIKY
0091                AM1=2.0*AIKY/SQRT(PI*AKY)
0092                CALL   RNSTRZ(2445632111)
0093                SO=POPO*SR*1.0E+16
0094                SP=1.0
0095                AITG=  CMUPOH /(AKY*PI)
0096                DE1=0.016
0097                DE2=0.355
0098                DE3=0.2
0099                DI1=1./0.016
0100                DI2=1./DE2
0101                DI3=1./DE3
0102                AMC1= SQRT(2.0*2.7183*AIKY)*ED2
0103                DO  11  I=6, 36
0104                EBL =(I-5)*DE2 -4.0
0105                EMV= EBL -0.1685
0106                EBD(I)= 2.0**EBL
0107                EG(I-1)= 2.0**EMV
0108             11 EDE(I)=EBD(I)*(2.0**DE2 -1.0)
0109                JE1=JE+1
0110                DO  12  I=31, JE1
0111                EBL= 7.005 +(I-36)*DE3
0112                EMV= EBL - 0.1405
0113                EBD(I)= 2.0**EBL
0114                EG(I-1)= 2.0**EMV
0115             12 EDE(I)= EBD(I)*(2.0**DE3 -1.0)
0116                DO  10  I=1, 5
0117                EDE(I)= 0.016
0118                EG(I)= EDE(I)*( -0.006
0119             10 EBD(I)= EDE(I)*(I-1)
0120                EDE(36)=29.69
0121                EG(36)= 1739.482
0122                IF  (NTAPE.NE.1)  GO  TO  2
0123                READ(10)  EAVG, DAVG, THE, NTW, NT, NTP, NION, DDEG, NSW,SWF, E,U
0124                READ(10) Z, WT, ANIOS,SF3
0125                NS=NSW*K2
0126                WGHT=1.0
0127                REWIND  10
0128                NT=NT ;
0129             2 CONTINUE
0130                IF  (NTAPE.NE.2)  GO  TO  5
0131                READ(9 )  EAVG, DAVG, THE, NTW, NT, NTP, NION, DDEG, NSW,SWF, E, U
0132                READ(9 ) Z, WT, ANIOS,SF3
0133                NS=NSW*K2
0134                WGHT=1.0
0135                REWIND  9
0136                NT=NT+1
0137             5 CONTINUE
0138                IF  (NTAPE.NE.0)  GO  TO  122
```

```
FORTRAN IV G LEVEL 18                MAIN                DATE = 72126          05/31/02
0139              NION= SQRT(SO*10/AITG)
0140               ANEAM=NION*0.33333/DEN
0141              CALL  SORCE2(NTP, EAVG, NTW, PM, FID)
0142              AL=ALOG(EAVG)
0143              DAVG=AL
0144              DDEG=DAVG*ND/SQRT(EAVG)
0145               SF2=1.0
0146              IF  (NION.NE.0)  SF2=NTW/NION
0147              SN=SO*AL*ND*SF2/(ETV*SQRT(EAVG))
0148              NSW=ANSW +0.5
0149              IF (SN.GT.ANSW)  NSW=SN +0.5
0150              NS=NSW*K2
0151              SWF=NSW/SN
0152              SF=SWF
0153              SF1=SF2
0154              SF3=SF2
0155              CALL  SOURCE(NS,NTP)
0156              TME=0.0
0157              ANIOS=NION
0158              WGHT=1.0
0159          122 CONTINUE
                C  GENERATE SOURCE PARTICLES TO INITIALIZE THE PROGRAM
0160              ANEAM=NION*0.33333/DEN
0161              WRITE(6,70)
0162           70 FORMAT('1','PROGRAM INITIALIZATION AND THE INITIAL PARAMETERS',//)
0163              WRITE(6,71) AITG, NION, NTP, AL, EAVG
0164           71 FORMAT(' ', 'AITG =',E14.6, 3X,'NION =',E14.6,3X,'NTP =',I12,3X,
                  1'OCEN(EO) =', E14.6,3X, 'EAVG= ',E14.4,/)
0165           72 FORMAT(' ',8X,'OCOL',13X,'OCEN',13X,'ENERGY',11X,'Z',14X,'U', 10X,
                  X'WEIGHT',/)
0166           75 FORMAT(' ',F14.6,3X, F14.6, 3X, F14.5,3X, F12.4,3X,F12.4,3X,E13.4)
0167              WRITE(6,72)
0168              RIP=RAN3Z(0)
0169               I=1
                C WITHIN THE OUTER I LOOP THE I TH PARTICLE HISTORY
                C  GEOMETRIC ROUTINE ENTERING HERE
0170            7777 IF ((E(I).LE.0.01.OR.(E(I).GT.EO*1.5)) GO  TO  9008
0171              TCEN=DDEG/ETV
0172            7778 IF ((NIZN.GT.0).AND.(I.GT.NTP1))  TCEN= DIZN(NOZN+1)
0173              U(I)=U(I)*TWSN
0174              LFG2=0
0175              ETAG1=1.0
0176              IF (E(I).GT.EO1) ETAG1=-1.0
0177              SQE= SQRT(E(I))
0178              DVA=EF*TCEN*EMASSI*0.5
0179              VZA=SQE*ETV*U(I) + DVA
0180              VYS=E(I)*ETV*(1.0-U(I)*U(I))*ETV
0181              OCEN= SQRT(VZA**2+VYS)*TCEN
0182              NFLG=0
0183            900 IF (E(I).LE.0.01  GO  TO  9008
0184              IF  (NFLG .EQ.0)  GO  TO  809
0185              SQE= SQRT(E(I))
0186              DVA=EF*TCEN*EMASSI*0.5
```

```
FORTRAN IV G LEVEL 18                    MAIN                    DATE = 72126          05/31/02

0187                VZA=SQE*ETV*U(I) + DVA
0188            809 DZ=VZA*TCEN
0189                ENEW= E(I) + EF*DZ
0190                IF (ENEW.LE.0.0)  GO  TO  9008
0191                SQEP=SQRT(ENEW)
0192                UNEW= SQE*U(I)/SQEP +DVA*2./(SQEP*ETV)
0193                ALI=ALOA(E(I))
0194                AL2=ALOA(ENEW)
0195                ALI=1.0/ALI + 1.0/AL2
0196                ALO=2.00/ALI
0197                RI=RAN3Z(0)
0198                ALR=ABS(ALOG(RI))
0199                DCOL= ALR*ALO
0200                IF (NFLG.EQ.1)  GO  TO  901
0201                IF(MOD(I,IMI.EQ.0)  WRITE(6,75)     DCOL,DCEN,E(I),Z(I),U(I),WY(I)
0202                DCOL2=ABS(ALOG(1.-RI))*ALO
0203                DTOT=DTOT +DCOL+DCOL2
0204                ETOT=ETOT+E(I)
0205                IF (NOPN.LT.2)  GO  TO  901
0206                TCEN2=TCEN
0207                DCEN2=DCEN
0208                EIO=E(I)
0209                UIO= U(I)*IUSN
0210                ZIO=Z(I)
0211                ENWO=ENEW
0212                UNWO=UNEW
0213                DZO=DZ
0214            901 NFLG=1
0215            902 IF  (DCEN.LT.DCOL)  GO  TO  905
0216                TCEN=TCEN - DCOL*TCEN/DCEN
0217                 DZC=DZ*DCOL/DCEN
0218                E(I)=E(I) + EF*DZC
0219                Z(I)=Z(I) +DZC
0220                DCEN= DCEN -DCOL
0221                NCOL=NCOL+1
0222                IF (MOD(NCOL,MC).EQ.0) RIP=RAN3Z(0)
0223                IF (MOD(NCOL,MC).NE.0) RIP=1.0-RIP
0224                CALL  COLPRO(I, TCEN, NTP, RIP, NUST, &9008, &9045)
0225                IF((E(I).LE.0.0).AND.(LFG2.EQ.1))  GO  TO  9008
0226                GO  TO  900
0227            905 E(I)=ENEW
0228                U(I)= UNEW
0229                Z(I)= Z(I) + DZ
0230                IF ((Z(I).LE.0.0).OR.(Z(I).GT. D ))   GO  TO  9008
0231                J=INT(Z(I)*DZIV)+1
0232                IF (J.GT.JZ)  J=JZ
0233                K= INDEXE(E(I))
0234                IF (K.GT.JE) K=JE
0235                IF (K3.EQ.1)  GO  TO  904
0236                ETAG2=1.0
0237                IF (E(I).GT.ED1)  ETAG2=-1.0
0238                IF ((ETAG1*ETAG2).GT.0.0)  GO  TO  904
0239                IF (ETAG2.GT.0.0)  GO  TO  903
```

```
FORTRAN IV G LEVEL 18                    MAIN                    DATE = 72126          09/31/07

0240              NUP=NUP+1
0241              ESE(NUP)=E(I)
0242              WT(I)=WT(I)/K3
0243              WSE(NUP)=WT(I)
0244              GO  TO  904
0245           903 IF(((LFG2.EQ.1).OR.(NDPN.LT.2))  NDNR=NDNR+1
0246              IF(((LFG2.EQ.2).OR.(NDPN.LT.2))  IRR(NDNR)=I
0247              NDN=NDN+1
0248              IF (MOD(NDN,K3).NE.0)  GO  TO  9045
0249              WT(I)=WT(I)*K3
0250           904 NEZ(K,J)= NEZ(K,J) + WT(I)
0251              EKZ(K,J)=EKZ(K,J)+E(I)*WT(I)
0252          9045 IF(((LFG2.EQ.1).OR.(NDPN.LT.2))  GO  TO  9009
0253              LFG2=1
0254              E(I)=EIO
0255              U(I)=UIO
0256              Z(I)=ZIO
0257              TCEN=TCEN2
0258              DCEN=DCEN2
0259              DCOL=DCOL2
0260              ENEW=ENWO
0261              UNEW=UNWO
0262              DZ=DZO
0263              IF (IUSN.LT.0)  GO  TO  900
0264              GO  TO  902
0265           9008 NES=NES+1
0266              IF (E(I).LT.ED1) WT(I)=WT(I)/K3
0267              IF (WT(I).GT.1)  WT(I)=1.0
0268              WNES=WNES +WT(I)
0269              IES(NES)=I
0270              IF  (E(I).GT.0.0)  ETEP=ETEP+E(I)*WT(I)
0271           9009 I= I+1
0272              LFG2=0
0273              IF (I.LE.NTP)  GO  TO  7777
0274              IF ((I.GT.NTP).AND.(NIZN.GT.0))  NOZN= NOZN-1
0275              IF (NOZN.LT.0)  NIN=1
0276              IF ((NIZN.GT.0) .AND.(NIN.EQ.0))  GO  TO  7778
0277              NIZN=WNIZN +0.5
0278              NWS=WNES +0.5
0279              NASW=WNAS  +0.5
0280              WNAS=WNAS/KELC
0281              WNSE=WNSE/KELC
0282              NSEW=WNSE +0.5
0283              DO  191  M=1,JE
0284              DO  191  N=1,JZ
0285              EE(M)=EE(M) + EKZ(M,N)
0286           191 NE(M)= NE(M) + NEZ(M,N)
0287              IF ((K3.EQ.1).OR.(NUP.EQ.0))  GO  TO  24
0288               N3=K3/JZ
0289              IF  (N3.LT.1)  M3=1
0290              NK3=K3-1
0291              IF (NK3.LT.JZ) IZ=NK3
0292               IF  (N3.GE.1)  M3=N3
```

```
FORTRAN IV G LEVEL  18                  MAIN                    DATE = 72126        09/31/02

0293              DO  23  N=1,NUP
0294              K=INDEXE(ESE(N))
0295              EE(K)=EE(K) + (K3-1)*ESE(N)*WSE(N)
0296              NE(K)=NE(K) +(K3-1)*WSE(N)
0297           22 IF (NK3.GE.JZ)  IZ=JZ
0298              DO  21  L=1,IZ
0299              NREZ(L)=NREZ(L) + M3*WSE(N)
0300           21 EZ(L)=EZ(L) + ESE(N)*M3*WSE(N)
0301              NK3=NK3 -JZ*M3
0302              IF (NK3.LT.JZ)  IZ=NK3
0303              M3=1
0304              IF (NK3.GT.0)  GO  TO  22
0305           23 CONTINUE
0306           24 CONTINUE
0307              DO  192  M=1, JZ
0308              DO  1915  N=1,JE
0309              NREZ(M)=NREZ(M) + NEZ(N,M)
0310         1915 EZ(M)=EZ(M) +EKZ(N,M)
0311          192 CONTINUE
0312              NTPN=NTP+NIZN-NES-NAS
0313              NTPW=0
0314              ETOL=0.0
0315              DO  14  M=1,JE
0316              ETOL= ETOL +EE(M)
0317           14 NTPW=NTPW +NE(M)
0318              EWTOL=ETOL
0319              NTPT=NTPN
0320              DAVGO= DAVG*ND
0321              DAVS=(DAVGO*1000.)**NTVP
0322              DSAVS=DSAVS+ DAVS
0323              DTAVS=DTAVS +DAVS
0324              DISG=1.0/DSAVS
0325              ANTPO = 1.0/NTPW
0326              NSTP= NSTP + NTPW +0.5
0327              NTPS=NTPS+ NTPN*DAVS
0328              NTPO=NTP
0329              EWAVG=EWTOL*ANTPO
0330              NSO=NS
0331              NSWO=NSO*WN2 +0.5
0332              NTP=NTP+NIZN
0333              NTPP=NTP
0334              DAVG=DTOT*0.5/NTP
0335              EAVG= ETOT/NTP
0336              DEG=DDEG/ETV
0337              THE=THE+DEG
0338              THS=THS+DEG
            C IF SOURCE PARTICLES SO(NS)< NES THEN ELIMINATE AS MANY RESIDUE PARTICLES AT TH
0339              NTPS=NTPW+ NWS +NASW
0340              NTW=NTPW +.5
0341              DCENI= ALOG( EAVG)
0342              DDEG=DAVG*ND/SQRT( EAVG)
0343              BF=1.0
0344              IF  ((WNES+WNAS).GT.0.0)  BF=(WNIZN+SN)/(WNES+WNAS)
```

```
FORTRAN IV G LEVEL  18                  MAIN                    DATE = 72126          05/31/02
0345              SF2=1.0
0346              IF (ANIOS.NE.0)  SF2=NTPS/ANIOS
0347              ESAVG=0.0
0348              IF(WNES.NE.0.0)  ESAVG= ETEP/WNES
0349              EASAVG=0.0
0350              IF (WNAS.NE.0)  EASAVG=ETAS/WNAS
0351              SWFO=SWF
0352              CN5S=CN5S+ SN=DAYS*WGHT
0353              CSF1= CSF1 +(ETEP+ETAS+40.*WNIZM)/TMS
0354              SF1=CSF1*D15G/(SO*EO)
0355              CSF2= CSF2 + SF2*DAYS
0356              IF ((MOD(NT,9),EQ,0),AND.(NT.NE.0))  SF3=CSF2*D15G
0357              WGHT=SF1/SF3
0358              SN=SO*DDEG*SF3/ETV
0359              IF (NTP.GT.4920)  ANSW=2
0360              NSN=ANSW +0.5
0361              IF (SN.GT.ANSW)  NSW=SN +0.5
0362              SWF=NSW/SN
0363              NS=NSW*K2
0364              SF=SWF
0365              NMEW=WNMET +0.5
0366              NEXN=WNEXC +0.5
0367              W5VLE=W5VLE+ WVALUE*DAYS
0368              CETEP=CETEP + ETSP*DAYS
0369              CETAS=CETAS +ETAS*DAYS
0370              CN5IZ=CN5IZ+WNIZN*DAYS
0371              CNES=CNES+WNES*DAYS
0372              CNAS=CNAS+WNAS*DAYS
0373              CNMET=CNMET+WNMET*DAYS
0374              CNEXC=CNEXC+WNEXC*DAYS
0375              CESAVG=CESAVG+ ESAVG*DAYS
0376              CEASG=CEASG+EASAVG*DAYS
0377              CLOSS= CLOSS+(WNIZN-WNES)*DAYS/(TMS*SF3)
0378              WVLE1=W5VLE*D15G
0379              NSAVG=CN5S*D15G +0.5
0380              NIZAVG=CN5IZ*D15G +0.5
0381              ANIZVG=CN5IZ*D15G
0382              ANSVG=CN5S*D15G
0383              AETEP=CETEP/DTAVS
0384              AETAS=CETAS/DTAVS
0385              IF (ANIZVG.NE.0)  WVLE2=(EO*ANSVG-AETEP-AETAS)/ANIZVG
0386              ALOSS=CLOSS*D15G
0387              ANES=CNES/DTAVS
0388              ANAS=CNAS/DTAVS
0389              AESAVG=CESAVG*D15G
0390              AEASG=CEASG*D15G
0391              ANMET=CNMET*D15G
0392              ANEXC=CNEXC*D15G
0393              ANS=ANES+ ANAS -ANIZVG
0394              IF (ANS.LE.0.0)  ANS=NSW/SWF
0395              IF (WNIZN .NE.0.0)  WVALUE= (SN*EO*WGHT-ETEP- ETAS)/WNIZN
0396              IF (ANIZVG.NE.0)  WVC=(ANS*EO-AETEP-AETAS)/ANIZVG
0397              IF (NS.GE.1)  TMS=0.0
```

```
FORTRAN IV G LEVEL  18                  MAIN                DATE = 72126        03/31/02

0398                NET=NES+NAS
0399                CALL  SOURCE(NS,NTP)
0400                 IF (NET.GT.0)  CALL  FILHOL(NTP,NES, NAS)
0401              90 NIZNT= NIZNT +NIZW
0402                IF  (NUP .GT.0)  CALL  SEPROC(NTP, K3)
0403                IF (NDNR.GT.0)  CALL  RRPROC(NTP,K3)
0404                NAST= NAST +NWS
                  C  INTRODUCE THE SOURCE PARTICLES FOR THE NEXT CYCLE
                  C EVERY FIFTH CYCLE RENORMALIZE NION AND OUTPUT QUANTITIES ARE PRINTED
                  C RENORMALIZATION PROCESS OF NION
0405                AN5TP= 1.0/N5TP
0406                ANTP5=D5AVS/NTP5
0407                N5TPN=N5TP
0408                DO  16  M=1,JZ
0409                CNREZ(M)= CNREZ(M) + NREZ(M)*DAVS
0410              16 CEZ(M)= CEZ(M) + EZ(M)*DAVS
0411                DO  18  M=1,JZ
0412                FNREZ(M)=NREZ(M)*ANTPO
0413              18 EZ(M)= EZ(M)/ NREZ(M)
0414                SUMNE=0.0
0415                DO   193  M=1,JE
0416                SUMNE= SUMNE + NE(M)
0417                TNE(M)= TNE(K) + NE(M)/EDE(M)
0418             193 FNE(M)=NE(M)*ANTPO/EDE(M)
0419                EAV3=0.
0420                DO  1755  M=1, JE
0421                EAV3= EG(M)*FNE(M)*EDE(M) +EAV3
0422             1755 CNE(M)= TNE(M)*AN5TP
0423                CNION=CMUPDP*2.0*AIKT/SQRT(AKT*PI)
0424                ANION=0.0
0425                ANIO2=0.0
0426                DO  20  M=1,JE
0427                ANIO2= ANIO2 + FM1(EG(M),CNE(M))*EDE(M)
0428                ANION= ANION + FM1(EG(M),FNE(M))*EDE(M)
0429              20 CONTINUE
0430                DNION=ANION*CNION
0431                DNIO2=ANIO2*CNION
0432                CJD=ALOSS
0433                IF (ALOSS.LT.0.0)  CJD=0.0
0434                IF (ABS(CJD).GT.110.*S0)  CJD=110.*S0
0435                TNION=ABS(S0+CJD)/DNION
0436                TNIO2=ABS(S0+CJD)/DNIO2
0437                TNIO2=SQRT(TNIO2)
0438                TNION=SQRT(TNION)
0439                CNIO2=CNIO2 + TNIO2*DAVS
0440                ANIOS=CNIO2*D15G
0441                WF=1.0
0442                IF (ANES+ANAS.NE.0.)  WF=(ANIZVG+ANSVG)/(ANES+ANAS)
0443                IF  (MOD(NT,NPM).NE.0)  GO  TO  1790
0444                NION=ANIOS
0445                IF ((WF.GT.0.005).AND.(WF.LT. 200.)) NION=ANIOS*WF**NBF
0446             1790 CONTINUE
0447                DAVT=DAVG*ND
```

```
FORTRAN IV G LEVEL  18                  MAIN                    DATE = 72126          05/31/02

0448                ALPHA=WNIZN/(NTPW*DAVT)
0449                ANEAM=ANION*0.33333/DEN
0450             1785 FORMAT('0','ANION=',E13.4,3X,'ONION=',E13.4,3X,'TNION=',E13.4,3X,
                    X'ALPHA=',F12.5,4X,'LEAK=',E13.4,/)
0451                WRITE(6,1785)  ANION, ONIOZ, TNIOZ, ALPHA, ALOSS
0452                WRITE(6,171)  EO, PDPO, D, DETZ, DEG, NDPN
0453             171 FORMAT('1','GAS: HE',2X,'SOURCE ENERGY:',F5.0,3X,'PRESSURE:',F4.0,
                    X 'TORR',3X,'SLAB THICKNESS:',F4.1,' CM',' DX:',F4.1,' CM',4X,'DELT
                    X=',E12.5,3X,'#PROC=',I2,/ )
0454                WRITE(6,180) NT, TME, NS,NSN
0455             180 FORMAT(' ','NUMBER OF TIME CYCLES=',I6,3X,'TIME =', E12.5,' SEC',
                    13X, 'GENERATED SOURCE PART =', I5,3X, 'WGTED # OF SOURCE PART.=',
                    1I5 )
0456                WRITE(6,1805) DAVT, DCEN1,EAVG,EHAVG, EAV3
0457             1805 FORMAT(' ','DAVG =',F12.5,3X, 'DCEN =',F12.5,3X,'OVERALL EAVG=',
                    1F10.5,3X,' WGTED EAVG=',F10.5,3X, 'EAVG CHECK=',F10.5,/)
0458                WRITE(6,181)
0459             181 FORMAT(' ', 1X,'NUMBER DISTRIBUTION',2X, 'FLUX DENSITY',4X,'AVERAGE
                    1E ENERGY',3X,'CUMULATED # DISTR',3X,'ENERGY INTERVAL',2X,'CUM FLUX
                    XDEN',4X,'#DEN-DEVIATION',/)
0460                CESUM=0.0
0461                DO  281  M=1, JE
0462                CESUM=CESUM + CNE(M)*EDE(M)
0463                IF (NE(M).GT.0)  DEAVG=EG(M)-EE(M)/NE(M)
0464                IF (NE(M).LE.0)  DEAVG=0.0
0465                 IF (NE(M).EQ.0)  DNAVG=0.
0466                FNOV=CNE(M)-FNE(M)
0467                 VFE=SQRT(EG(M))*ETV
0468                FLX1=FNE(M)*VFE
0469                FLX2=CNE(M)*VFE
0470                IF (NE(M).EQ.0)  GO  TO  281
0471                IF((M.EQ.1).AND.(DEAVG.GT.0.0)).OR.((M.EQ.JE).AND.(DEAVG.LT.0.0))
                    X) DNAVG=-DEAVG*FNE(M)*0.5/EDE(M)
0472                IF ((DEAVG.LT.0.0).AND.((M+1).LE.JE)) DNAVG=-DEAVG*(FNE(M+1)-FNE(M
                    X))/(EG(M+1)-EG(M))
0473                IF ((DEAVG.GT.0.).AND.(M.GT.1)) DNAVG=DEAVG*(FNE(M)-FNE(M-1))/(EG(
                    XM)-EG(M-1))
0474             281 WRITE(6,1821 M,FNE(M),FLX1,EG(M),CNE(M),EDE(M),FLX2 , DNAVG
0475             182 FORMAT(' ',I3,E12.5,5X,E12.4, 5X, F12.6,8X, E12.5,6X, F12.6,4X,E12
                    X.5,4X,E12.4)
0476                DET=DDEG/ETV
0477                WRITE(6,183) NTPP, NSTPN,NTPW ,DET, DAVGO, NCOL
0478             183 FORMAT(' ',//,3X,'TOTAL # OF PARTICLES PROCESSED =',I5,5X,'CUMULAT
                    1ED # OF PARTICLES =',I10,3X,'WGTED TOTL # OF PARTICLES=',F12.3,//,
                    X' NEW DT=',E12.5,4X,'OLD DAVG=',F12.5,4X,'# OF COLLISIONS=',I5,///)
0479                WRITE(6,187)
0480             187 FORMAT(' '    ,1X, 'ZONE NUMBER', 5X, 'AVERAGE ENERGY',5X,'NUMBER O
                    1F ELECTRONS',3X,'CUMULATIVE # DEN(Z)',3X,'CUMU. AVG ENERGY',3X,
                    1'ZONE # DENSITY',/)
0481                DO 282  M=1, JZ
0482                ANZ=CNREZ(M)*ANSTP*D15G
0483                PCZ=CEZ(M)/CNREZ(M)
0484             282 WRITE(6,184) M, EZ(M), NREZ(M), ANZ, PCZ,FNREZ(M)
```

```
FORTRAN IV G LEVEL 18                    MAIN                    DATE = 72126          05/31/02
0485          184 FORMAT(' ', 5X, I5, 3X, F14.6,8X,F14.6,  3X, F14.5, 7X,F14.6,7X,
                 1F12.5)
0486                WRITE(6,185)  NION,WNIZN,NIZN, ANIZVG,WNES, NES,ANES, WNAS,NAS,
                    XANAS, NTP, K1, K2, WF,BF
0487          185 FORMAT('0', 'NUMBER OF POSITIVE ION=',E14.6,//,' NUMBER OF IONIZAY
                    1IONS OCCURED=',F9.3,5X,'ACTUAL #=',I5,3X,'CUMULATIVE AVG=' ,F10.3,
                    X//,' # OF ESCAPED PARYICLES=',F9.3,5X,'ACTUAL#=',I5,3X,'CUMULAYIVE
                    X AVG=',F9.2,//,' # OF ABSORPED PARTIC=',E12.4,3X,'ACTUAL #=',I5,3X
                    X,'CUMULATIVE AVG=',F9.2,//, ' TOYAL # OF PARYICLES IN YHE NEW CYCL
                    XE =',I5,//,' K1=',I2, 3X,'K2=',I2,     4X,' BALANCE   FACTOR WF=',
                    XF10.5,3X,'BF2=', F10.5,/)
0488                WRITE(6,188) ED1, NPN, ND, KELC
0489          188 FORMAT(' ENERGY DIVIDING LINE FOR  RR PLAY E=',F10.4,4X,'EVERY',I2
                    X,'TH CYCLE CHANGE N+',//,' TIME INTERVAL MULTIPLY FACTOR=',F9.4,
                    X3X, 'ELASTIC XSEC / BY',I2,/)
0490                NOR=NON-NON/K3
0491                WRITE(6,186) NSO, NSW, ANSVG,NOR,WNSE,WNNET,ANNET,WNEXC,ANEXC,NUP,
                    XWGHT
0492          186 FORMAT(' ',    ' SOURCE PARTICLES GENERATED OLD =',I10,//,
                    1' SOURCE PARTICLES (WEIGHTED) IN THE NEW CYCLE =',I10,3X, 'CUMULAT
                    XIVE AVG=',F10.3,//
                    1,' # OF PARYICLES KILLED OFF IN RR PROC=',I10,//,
                    1' # OF SUPER ELASTIC COLLISION=',F10.4,//,
                    1                 ' # OF EXCITATIONS TO META STABLE LEVEL='
                    X,F10.4, 3X, 'CUMULATIVE AVG=', F9.3,
                    1                  //,' # OF EXCITATIONS TO OTHER LEVELS=',F10.4,
                    X4X, 'CUMULATIVE AVG=',F9.2,//,' # OF PARTICLES CROSS OVER TO HIGH
                    XENERGY REGION NUP=',I5,4X,'OVERALL WT=',E12.4,/)
0493                WRITE(6,195)  EF, EOP, POPO ,SF1, SF2, SF3
0494          195 FORMAT (' E-FIELD=', F9.2,'V/CM', 4X,'E/P=',F9.2,4X,'P/PO =',F7.1,
                    X3X, 'SF1=', E12.4,3X, 'SF2=',E12.4,3X, 'SF3=', E12.4,/)
0495                WRITE(6,170) SO,WVC,SWFO,WVALUE,WVLE1, WVLE2,ESAVG, AESAVG,EASAVG,
                    XAEASG
0496          170 FORMAT(' SOURCE RATE =',E11.3,' PER SEC',4X,'W-VALUE(LOCAL  )=',
                    XF10.3, 3X,'SOURCE RED.FACT=',E13.4,//,' W-VALUE=',
                    XF10.3,4X,'CUM AVG WV1=',F9.3,3X,'CUM AVG WV2=',F9.3,
                    1      //,' AVERAGE ENERGY OF ESCAPED PARTICLES=',F12.5,3X,'CUMULAY
                    XIVE AVG=',F11.4,//,' AVERAGE ENERGY OF ABSORPED PARTICLES=',F12.5,
                    X3X, 'CUMULATIVE AVG=',F11.4,/ )
0497                CKFNE=SUMNE*ANTPO
0498                WRITE(6,189)CKFNE, CESUM, TNION, TNIO2
0499          189 FORMAT(' ','THE SN # CHECK =',F12.4,3X,'THE CN # CHECK=',F12.4,//,
                    1' CALCULATED N+ USE F=',E14.6,3X,'CALC. N+ USE CN =',E14.6,3X, 'AR
                    XTIF CN+=',E14.6,//)
0500                IF  ((MOD(NT,5).NE.0).OR.(MOD(NT,10).EQ.0))  GO  TO  3
0501                WRITE(9) EAVG, DAVG, TME, NTW, NT, NTP, NION,DDEG, NSW,SWF, E, U
0502                WRITE(9) Z, WT ,ANIOS, SF3
0503                ENDFILE 9
0504                REWIND 9
0505              3 CONTINUE
0506                IF  (MOD(NT,10).NE.0)  GO  TO  4
0507                WRITE(10) EAVG, DAVG, TME, NTW, NT, NTP, NION, DDEG, NSW,SWF, E, U
0508                WRITE(10) Z, WT  ,ANIOS, SF3
```

```
FORTRAN IV G LEVEL  18                  MAIN                    DATE = 72126          05/31/02
0509                  ENDFILE 10
0510                  REWIND 10
0511                4 CONTINUE
0512                  IF (MOD(NT,NCUM).NE.0) GO TO 6666
0513                  NSTP=0
0514                  NT#5=0
0515                  D5AVS=0.0
0516                  NAST=0
0517                  NIZNT=0
0518                  W5VLE=0.0
0519                  CN5IZ=0.0
0520                  CN55=0.0
0521                  CNMET=0.0
0522                  CNEXC=0.0
0523                  CESAVG=0.0
0524                  CEASG=0.0
0525                  CSF2=0.0
0526                  CLOSS=0.0
0527                  CNIO2=0.0
0528                  CSF1=0.0
0529                  DO  54  M=1,JZ
0530                  CNREZ(M)=0.0
0531               54 CEZ(M)=0.0
0532                  DO  55  M=1,JE
0533                  TNE(M)=0.
0534               55 CNE(M)=0.0
0535                  CNES=0.0
0536                  CNAS=0.0
0537                  DTAVS=0.0
0538                  CETEP=0.0
0539                  CETAS=0.0
0540                  IF (MOD(NT,10).NE.0) GO TO 6666
0541            6666  CONTINUE
0542                  NT= NT + 1
0543                  DO  56  M=1,500
0544                  DIZN(M)=0.0
0545                  IES(M)=0
0546                  IRR(M)=0
0547               56 IAS(M)=0
0548                  DO  51  N=1,50
0549                  WSE(N)=0.0
0550               51 FSE(N)=0.0
0551                  I=1
                C.  REINITIALIZE THE OUTPUT QUANTITIES
0552                  DO  52  M=1, JE
0553                  FE(M)=0.
0554               52 NE(M)= 0.0
0555                  DO  53  M=1,JZ
0556                  NREZ(M)= 0.0
0557                  FZ(M)=0.0
0558                  DO  53  N=1,JE
0559                  FKZ(N,M)=0.0
0560               53 NEZ(N,M)=0.0
```

```
FORTRAN IV G LEVEL 18                    MAIN                    DATE = 72126          05/31/02
0561                  NIZN=0
0562                  NOZN=0
0563                  NIN=0
0564                  NES=0
0565                  NSE=0
0566                  NCOL=0
0567                  NAS=0
0568                  NASN1=0
0569                  DTOT=0.0
0570                  NMET=0
0571                  NEXC=0
0572                  WNES=0.0
0573                  WNAS=0.0
0574                  WNEXC=0.0
0575                  WNIZN=0.0
0576                  WNMET=0.0
0577                  WNSE=0.0
0578                  ETAS=0.0
0579                  ETEP=0.0
0580                  ETOT=0.0
0581                  NW=0
0582                  NUP=0
0583                  NDN=0
0584                  NDNR=0
0585                  INIV=INIV+1100
0586                  CALL   RN3INZ(INIV)
0587                  WRITE(6,72)
0588                  IF (NT.LE.NTC) GO TO 7777
0589            9999  STOP
0590                  DEBUG  SUBCHK
0591                  END
```

```
FORTRAN IV G LEVEL 18                INDEXE                DATE = 72126          05/31/02

0001                  FUNCTION  INDEXE(E)
0002                  COMMON /AREA4/  CONST,DEL2, ELT,IE, DI1, DI2, DI3 ,DELI
0003                  IF (E.LT.0.08)  GO  TO  11
0004                  IF (E.GT.158.1335 ) GO  TO  13
0005               12 INDEXE= 5+(ALOG(E)*CONST +4.01)*DI2
0006                  RETURN
0007               11 INDEXE=1 + E*DI1
0008                  RETURN
0009               13 INDEXE=36 + (ALOG(E)*CONST-7.005)*DI3
0010                  RETURN
0011                  END
```

```
FORTRAN IV G LEVEL 18                SORCE2            DATE = 72126        05/31/12

0001              SUBROUTINE  SORCE2(NSO, EAVG, NTW,PM, FID)
0002              REAL  NI(55),FID(50) ,WI(55)
0003              COMMON D, E(5100),Z(5100), U(5100) ,IAS(500),IRN(650),NRN,IFS(600)
                 X,WT(5100),SWF
0004              COMMON /AREA2/ C1,MUO,EI,FIS,SF,ANEAM,PDPC,PI,EC,NION,JF,JZ,NASWI,
                 1USEW ,AKT, AIKT ,AMI ,AMC1,ED1,PDMUO, P026,K1, K2, K3, WN2,ED2,ED3
0005              COMMON /AS/ EG(55), EDE(55), EBD(55)
0006              FM1(E)= EXP(-E*AIKT)*SQRT(E)*AMI
0007              FM2(E)=AA/E
0008              FM3(E)=EXP(-E*AIKT)*SQRT(E) *AMC1
0009              REAL  MUO, NION
0010          604 FORMAT(' ',F12.6,3X, I3, 3X, E12.4, 2X, F12.6,2X,F12.5,3X,E12.4)
0011          605 FORMAT(' ','J1=',I3,3X,'ED1=',F9.3,      3X,'NSO=',I8,3X,'K1=',
                 1I8,3X,'K2=',I8,//,' INITIAL DISTRIBUTION',//,' # OF PTS',7X,'INDEX
                 1',5X, 'W DENSITY',4X,'AVG ENERGY',6X,'DEL-E',6X,'WEIGHT',/)
0012          606 FORMAT(' ','NS1=',I5,3X,'NS2=',I5,3X,'NSO=',I5,3X,'NTW=',I5,3X,
                 X'NTW(CALC)=',I5,//)
0013          607 FORMAT(' EAVG=', F12.5,3X,'WGHTED EAVG=',F12.5,//)
0014              AA=2.0*PM*SQRT(PM/PI)*EXP(-PM)
0015              J1=INDEXC(ED1-0.5)
0016              J2=J1+1
0017              ISUM=0
0018              WRITE(6,605)   J1, ED1,NSO, K1, K2
0019              DO  100  I=1,J1
0020              DNI   =FID(I)*EDE(I)*NTW/K1
0021              NI(I)=DNI
0022              WI(I)=1.0
0023              IF (FID(I).LE.0.0)  GO  TO  112
0024              IF (DNI.GT.200.)  NI(I)=200
0025              IF (DNI.GT.200.)  WI(I)=DNI/200.
0026              IF (DNI.LT.12)  NI(I)=12
0027              IF (DNI.LT.12)  WI(I)=DNI/12.0
0028          112 WRITE(6,604) NI(I), I, FID(I),EG(I), EDE(I),WI(I)
0029          100 ISUM=ISUM+ NI(I)
0030              JSUM=0
0031              DO  200  I=J2, JF
0032              DNI   =FID(I)*EDE(I)*NTW*K2
0033              NI(I)=DNI
0034              WI(I)=1.0
0035              IF (FID(I).LE.0.0)  GO  TO  202
0036              IF (DNI.GT.200.)  NI(I)=200
0037              IF (DNI.GT.200.)  WI(I)=DNI/200.
0038              IF (DNI.LT.12)  NI(I)=12
0039              IF (DNI.LT.12)  WI(I)=DNI/12.0
0040          202 WRITE(6,604) NI(I), I, FID(I),EG(I), EDE(I), WI(I)
0041          200 JSUM= JSUM + NI(I)
0042              NSUM=ISUM+JSUM
0043              NTW2=ISUM*K1 +JSUM*WN2 +0.5
0044              WRITE(6,606) ISUM, JSUM, NSUM, NTW, NTW2
0045              NTW=NTW2
0046              IE=1
0047              EAG=0.0
0048              DO  300  I=1, JE
```

```
FORTRAN IV G LEVEL  18                  SORCE2          DATE = 72126        05/31/02

0049              IF  (NI(I).LT.1)  GO  TO  300
0050               JI=NI(I) +IE -1
0051               DO  301  J=IE, JI, 2
0052              RI=RAN3Z(0)
0053              R2=1.0-RI
0054              E(J)= ERD(I) + EDE(I)*RI
0055              E(J+1)= ERD(I) + EDE(I)*R2
0056              WT(J)=WI(I)
0057              WT(J+1)=WI(I)
0058              EAG =EAG + E(J) +E(J+1)
0059              IF((MOD((JI-IE),2).EQ.0).AND.(J.GE.JI)) EAG=EAG-E(J+1)
0060          301 CONTINUE
0061              IE=IE+NI(I)
0062          300 CONTINUE
0063              NSO=NSUM
0064              EAVG=EAG/NSO
0065              I2=NSO
0066              ETW=0.0
0067              DO  400  I=1,NSO
0068              IF  (E(I).LE.ED1) WT(I)=K1 *WT(I)
0069              IF  (E(I).GT.ED1) WT(I)=WN2*WT(I)
0070              IF (E(I).LE.0.0)  E(I)=E(I-1)
0071              ETW=ETW + E(I)*WT(I)
0072          400 CONTINUE
0073              EWAVG=ETW/NTW2
0074              WRITE(6,607) EAVG, EWAVG
0075              QDZ=D*0.25
0076              I=1
0077          101 RZ=RAN3Z(0)
0078              RU=RAN3Z(0)
0079              Z(I)=D*RZ
0080              U(I)=2.0*RU -1.0
0081              I= I+ 1
0082          102 Z(I)=D*RZ
0083              U(I)=1.0- 2.0*RU
0084              I= I +1
0085          103 Z(I)=D*(1.0-RZ)
0086              U(I)= 2.0*RZ-1.0
0087              I= I +1
0088          104 Z(I)=D*(1.0-RZ)
0089              U(I)= 1.-2.*RZ
0090              I= I+ 1
0091              IF((I2 -I-3).GE.0)  GO  TO  101
0092              IF (I2.LT.I)  GO  TO  105
0093              RZ=RAN3Z(0)
0094              RU=RAN3Z(0)
0095              IF((I2 -I).EQ.2)  GO  TO   102
0096              IF((I2 -I).EQ.1)  GO  TO  103
0097              IF((I2 -I).EQ.0)  GO  TO  104
0098          105 RETURN
0099              END
```

```
FORTRAN IV G LEVEL  18                  SOURCE            DATE = 72126          05/31/02

0001                  SUBROUTINE  SOURCE(NS,NTP)
0002                  COMMON D, E(5100),Z(5100), U(5100) ,IAS(500),IRR(850),NRR,IES(600)
                     X,WT(5100),SWF
0003                  COMMON /AREA2/ C1,MUO,EI,EIS,SF,ANEAM,PDPO,PI,EO,NION,JE,JZ,NASW1,
                     1NSEW ,AKT, AIKT ,AM1 ,AMC1,ED1,PDMUO, PD26,K1, K2, K3, WN2,ED2,ED3
0004                  REAL  MUO, NION
                    C SOURCE PARTICLE PARAMETERS GENERATING PROCEDURE
                    C II: THE INITIAL INDEX VALUE FOR THE SOURCE PARTICLE
0005                  I=NTP+1
0006                  DO  100  K=1, NS, 2
0007                  E(I)=EO
0008                  RZ=RAN3Z(0)
0009                  RU=RAN3Z(0)
0010                  Z(I)=D*RZ
0011                  U(I)=RU+RU-1.0
0012                  WT(I)=WN2/SWF
0013                  IF  ((NS-K).LE.0)  GO  TO  100
0014                  E(I+1)=EO
0015                  Z(I+1)= RZ*D
0016                  U(I+1)=-U(I)
0017                  WT(I+1)=WN2/SWF
0018                   I=I+2
0019                100 CONTINUE
0020                  NTP=NTP+NS
0021                105 RETURN
0022                  END
```

```
FORTRAN IV G LEVEL  18                  MAIN                DATE = 72126        05/31/02

            C WHENEVER THERE ARE MORE HOLES THAN SOURCE PARTICLES, FILL IN THE EXTRA HOLES W
            C WITH PARTICLES AT THE END
0001                SUBROUTINE    FILHOL(NTP, NES, NAS)
0002                COMMON D, E(5100),Z(5100), U(5100) ,IAS(500),IRR(850),NRR,IES(600)
                   X,WT(5100),SWF
0003                K=NTP
0004                IF (NES.LE.0)  GO  TO  105
0005                DO  100  I=1, NES
0006                J=IES(I)
0007                E(J)= E(K)
0008                Z(J)= Z(K)
0009                U(J)= U(K)
0010                WT(J)=WT(K)
0011              100 K=K-1
0012                NTP=NTP-NES
0013              105 IF (NAS.LE.0)  RETURN
0014                K=NTP
0015                DO  300  I=1,NAS
0016                J=IAS(I)
0017                E(J)=E(K)
0018                Z(J)=Z(K)
0019                U(J)=U(K)
0020                WT(J)=WT(K)
0021              300 K=K-1
0022                NTP=NTP-NAS
0023              200 RETURN
0024                END
```

```
FORTRAN IV G LEVEL  18                  RRPROC              DATE = 72126          05/31/02

0001                  SUBROUTINE  RRPROC(NTP,K3)
0002                  COMMON D, E(5100),Z(5100), U(5100) ,IAS(500),IRR(850),NRR,IES(600)
                     X,WT(5100),SWF
0003                  COMMON /A6/ ESE(50) , NUP ,NONR,NON, WSE(50)
0004                  NR=NONR
0005                  K=NTP
0006                  IF (K3.LE.1)   RETURN
0007              100 CONTINUE
0008                  I=IRR(NR)
0009                  E(I)=E(K)
0010                  U(I)=U(K)
0011                  Z(I)=Z(K)
0012                  WT(I)=WT(K)
0013                  K=K-1
0014              200 NR=NR-1
0015                  IF  (NR.GT.0)  GO  TO  100
0016                  NTP=K+1
0017              300 RETURN
0018                  END
```

```
FORTRAN IV G LEVEL  18                  ALDA                DATE = 72126          05/31/02

0001                FUNCTION  ALDA(E)
0002                COMMON /AREA2/ C1,MUO,EI,EIS,SF,ANEAM,PDPO,PI,EO,NION,JE,JZ,NASW1,
                   1NSEW ,AKT, AIKT ,AM1 ,AMC1,ED1,PDMUO, PD26,K1, K2, K3, WN2,ED2,ED3
0003                COMMON /AREA3/ XSEL,XELC,XRCB,XMET,XEXC,XION,XTOL,KELC
0004                REAL  MUO, NION
0005                WELC=1.0/KELC
0006                IF (E.LE.5.)  XELC=23.0*PDPO
0007                IF((E.GT.5.).AND.(E.LE.19.8))  XELC=23.0*PDPO*SQRT(5./E)
0008                IF (E.GT.19.8)  XELC=EXP(-0.02*E)*16.76*PDPO
0009                XELC=XELC*WELC
0010                IF (E.LE.ED3 )  GO  TO  5001
0011           4001 AE1= 0.25 +(E    - 19.80)**2
0012                XMET= 0.109*(E     -19.80)/AE1 + 0.0311
0013                XMET=XMET*PDPO
0014                E19= E - 21.4
0015                AE2= 1850. + E19**2
0016                XEXC= 68.80*E19*PDPO/AE2
0017                IF (E.LT.21.4)  XEXC=0.0
0018                E24= E - E1
0019                AE3= 4900. + E24**2
0020                XION= 172.*E24/AE3
0021                XION=XION*PDPO
0022                IF  (E.LT.24.46)  XION=0.0
0023                XTOL= XMET +XEXC +XION +XELC
0024                XRCB=0.0
0025                XSEL=0.0
0026                GO  TO  6001
0027           5001 BE1=0.250+ (E-19.8)**2
0028                XSEL=0.109*(19.8 -E    )/BE1
0029                XSEL=XSEL*PDPO*ANEAM
0030                XRCB=PDMUO*NION/SQRT(E)
0031                XION=0.0
0032                XEXC=0.0
0033                XMET=0.0
0034                XTOL= XSEL +XELC +XRCB
0035           6001 ALDA=1.0/XTOL
0036                RETURN
0037                END
```

```
FORTRAN IV G LEVEL  18                  COLPRO              DATE = 72126          09/31/02

0001                  SUBROUTINE   COLPRO(I, DCEN, NTP, RTP, NUST, *, *)
0002                  COMMON D, E(5100),Z(5100), U(5100) ,IAS(500),IRR(850),NRR,IES(600)
                     X,WT(5100),SWF
0003                  COMMON /AREA1/ NIZN, NOZN, NAS, DIZN(500),NSE, NMET, NEXC, ETAS  ,
                     XWNMET, WNEXC, WNIZN, ELS,WNAS ,WNSE
0004                  COMMON /AREA2/ C1,MUO,EI,EIS,SF,ANEAM,PDPO,PI,E0,NION,JE,JZ,NASW1,
                     1NSEW ,AKT, AIKT ,AMI ,AMCI,E0I,PDMUO, PDZ6,K1, K2, K3, WN2,E02,E03
0005                  COMMON /AREA3/ XSEL,XELC,XRCB,XMET,XEXC,XION,XTOL,KELC
0006                  COMMON /A6/ ESE.50) , NUP ,NDNR,NON, NSE(50)
0007                  DIMENSION  PX(4)
0008                  PM3(E)=EXP(-E*AIKT)*SQRT(E)*AMC1
0009                  REAL  MUO, NION
                   C DECIDE WHICH KIND OF COLLISION
0010                  XTI =ALOA(E(I))
0011                  IF (E(I).LE.E03 )  GO  TO  5000
                   C  ENERGY REGION 1  19.0EV  TO  1 KEV
0012                4000 PMET= XMET*XTI
0013                  PEXC= XEXC*XTI
0014                  PION= XION*XTI
0015                  PELC=XELC*XTI
0016                  PX(1)= PEXC
0017                  PX(2)= PEXC +PMET
0018                  PX(3)= PX(2) +PION
0019                  IP=1
0020                  DO  45  IX=1,3
0021                 45 IF(RIP.GT.PX(IX))  IP=IX+1
0022                  GO  TO  (4015,4013,4017,4011), IP
0023                4011 R1=RAN3Z(0)
0024                  U(I)= 2.0*R1 -1.0
0025                  RETURN
                   C INELASTIC COLLISION PROCESS EXCITE TO META STABLE STATE
                   C EXCITATION TO METASTABLE LEVEL
0026                4013 WNMET= WNMET +WT(I)
0027                  E(I)= E(I) - 19.80
0028                  IF (E(I).LE.0.0)  RETURN1
0029                  NMET=NMET+1
0030                  RI1= RAN3Z(0)
0031                  U(I) = RI1 +RI1 - 1.0
0032                  RETURN
                   C EXCITION TO OTHER LEVELS
0033                4015 WNEXC =WNEXC +WT(I)
0034                  E(I)= E(I) -21.40
0035                  IF (E(I).LE.0.0)  RETURN1
0036                  NEXC=NEXC+1
0037                  RI1= RAN3Z(0)
0038                  U(I) = RI1 +RI1 - 1.0
0039                  RETURN
                   C IONIZITION PROCESS  PRODUCE SECONDRY ELECTIONS
0040                4017  WNIZN= WNIZN+ WT(I)
0041                  E(I)= E(I) - EI
0042                  IF (E(I).LE.0.0)  RETURN1
0043                  NIZN=NIZN+1
0044                  NOZN=NOZN   +1
```

```
FORTRAN IV G LEVEL  18                  COLPRO              DATE = 72126          05/31/02

            C IONIZATION PRODUCED PARTICLES ALWAYS PUT TO THE END OF RESIDUES
 0045             DIZN(NOZN)= DCEN
 0046             RI=RAN3Z(0)
 0047             U(I)=RI+RI-1.0
 0048             NNTP =NTP +NIZN
 0049             E(NNTP)= E(I)*0.1
 0050             Z(NNTP)=Z(I)
 0051             U(NNTP)=-U(I)
 0052             E(I)= E(I)*0.9
 0053             WT(NNTP)= WT(I)
 0054             IF ((E(NNTP)    .GT.ED1).OR.((E(I)*1.111+E(I).LT.ED1))  GO  TO 4018
 0055             NDN=NDN+1
 0056             NDNR=NDNR+1
 0057             WT(NNTP)=WT(I)*K3
 0058             IRR(NDNR)=NTP+NIZN
 0059          4018 CONTINUE
 0060             RETURN
            C ENERGY REGION 2 LOW ENERGY RANGE DOWN TO THERMAL REGION
 0061          5000 PELC= XELC*XTI
 0062             PSEL= XSEL*XTI
 0063             PRCB= XRCB*XTI
 0064             PX(1)= PRCB
 0065             PX(2)= PSEL +PRCB
 0066             IP=1
 0067             DO  55  IX=1,2
 0068          55 IF(RIP.GT.PX(IX))  IP=IX+1
 0069             GO  TO  (5013,5015,5011),IP
            C ELASTIC COLLISION PROCESS
 0070          5011 IF (ELS.LE.0.0)  GO  TO  50125
 0071             IF ((NUST.EQ.0).OR.(E(I).GT.2.5))  GO  TO  50124
 0072          50115 RE=RAN3Z(0)
 0073             R2=RAN3Z(0)
 0074             E1=ED2*RE
 0075             E2=ED2*R2
 0076             FR1=FM3(E1)
 0077             FR2=FM3(E2)
 0078             IF  (RE.LE.FR2)  RS=RE
 0079             IF  (R2.LE.FR1)  RS=R2
 0080             IF  ((RE.LE.FR2).OR.(R2.LE.FR1))  GO  TO  5012
 0081             GO  TO  50115
 0082          5012 ENN=RS*ED2
 0083          50124 IF((NUST.EQ.0).OR.(E(I).GT.2.5))  ENN=0.0
 0084             R1=RAN3Z(0)
 0085             U1=2.0*R1 -1.0
 0086             UM=SQRT((1.-U1**2)*(1.-U(I)**2))
 0087             US=2.0*RIP-1.0
 0088             COSST=U1*U(I) +SIGN(UM,US)
 0089             DELE=C1*(E(I)-ENN)*(1.0-COSST)*ELS*KELC
 0090             E(I)= E(I) -DELE
 0091             U(I)=U1
 0092          50125 IF( ELS.LE.0.0)  U(I)= 2.0*RAN3Z(0) -1.0
 0093             RETURN
            C RECOMBINATION PROCESS
```

```
FORTRAN IV G LEVEL  18                  COLPRO              DATE = 72126          05/31/02

0094          5013 NAS=NAS+1
0095               IAS(NAS)=I
0096               WNAS= WNAS+WT(I)
0097               IF  (E(I).GT.0.0)  ETAS=ETAS+E(I)*WT(I)
0098               RETURN2
                 C SUPERELASTIC COLLISION PROCESS
0099          5015 E(I)= E(I) + EIS
0100               WNSE=WNSE +WT(I)
0101               NSE=NSE+1
0102               RP= RAN3Z(0)
0103               U(I)= RP +RP -1.0
0104               RETURN
0105               END
```

```
FORTRAN IV G LEVEL 18                MAIN                DATE = 72125        01/28/29

      C MASTER PROGRAM OF MONTE CARLO SIMULATION OF NONLINEAR ALPHA PLASMAS (ALPHA PAR
      C TICLES INDUCED PLASMA ELECTRON ENERGY DISTRIBUTION STUDY
      C DEFINITIONS AND TERMINOLOGY
      C D: DIAMETER OF THE CYLINDER THE LENGTH OF THE FINITE REGION
      C E0:INITIAL ENERGY OF THE SOURCE PARTICLES SO(NS): SOURCE RATE
      C JZ:TOTAL NUMBER OF ZONES   JE: TOTAL NUMBER OF ENERGY RANGES
      C IAS: ARRAY FOR STORING THE INDEX OF ABSORBED PARTICLES NAS: THE NO. OF THEM
      C DD: DISTANCE TO BOUNDRY  DCOL: DISTANCE TO COLLISION  DCEN: DISTANCE TO CENSUS
      C DZ: ZONE INTERVAL
      C ARRAYS E, Z, U  STORING ENERGY, DISTANCE, AND DIRECTION OF PARTICLES
      C NTP: TOTAL NUMBER OF PARTICLES TO BE PROCESSED  EACH CYCLE
      C NT: TOTAL NUMBER OF TIME CYCLES
      C NS: NUMBER OF SOURCE PARTICLES PER UNIT TIME*DISTANCE  OR SOURCE RATE
      C NIZN: NUMBER OF IONIZATION PROCESS OCCURRED EACH TIME CYCLE
0001            REAL    NE(55), NEZ( 55,10),EZ(11),DX(3),EE(55),  CEZ(11),EKZ(55
     1,10) ,FIC(60),NREZ(11),EZI(11),NTPW
0002            DIMENSION  ENE(55), ENREZ(11), NRZI(11),CNE(55), CNREZ(11),TNE(55)
0003            COMMON D, E(5100),Z(5100), U(5100) ,IAS(500),IRR(950),NRP,IFS(600)
     X,WI(5100),SWF
0004            COMMON /AREA1/ NIZN, NDZN, NAS, DIZN(500),NSE, NMET, NEXC, ECAS  ,
     XWNMET, WNEXC, WNIZN, ELS,WNAS ,WNSE
0005            COMMON /AREA2/ C1,MUO,EI,EIS,SF,ANEAM,PDPO,PI,EO,NION,JE,JZ,NASW1,
     1NSEW ,AKT, AIKT ,AM1 ,AMC1,ED1,PDMUU, PD26,K1, K2, K3, WN2,ED2,ED3
0006            COMMON /AREA3/ XSEL,XELC,XRCR,XMET,XEXC,XION,XTOL,KELC
0007            COMMON /AREA4/  CONST,DEL2, ELT,IE, DI1, DI2, DI3 ,DELI
0008            COMMON /A5/ EG(55), EDE(55), EBD(55)
0009            COMMON /A6/ ESE(50) , NUP ,NONR,NDN, WSE(50)
0010            REAL  MUO, NION,ND
      C STATEMENT FUNCTION FOR EVALUATING  MAXWELLIAN AND RELATED FCNS
0011            FMI(E,FNEM)=EXP(-E*AIKT)*SQRT(E)*FNEM
0012            DATA  TMS,NT,DTOT,ETOT,NIN,NIZNT,NAST/0.0,1,0.0,0.0,0.0,0.0,0/
0013            DATA WSVLE,CNSIZ,CNSS,CNES,CNAS,CNMET,CNEXC,CESAVG,CEASG/9*0.0/
0014            DATA CNE/ 55*0.0/, CEZ/11*0.0/, CNREZ/11*0.0/,TNE/55*0./,EKZ/550*0
     1.0/,DSAVS/0.0/
0015            DATA NEZ/ 550*0./,NE/55*0./,EZ/11*0./,NREZ/11*0./,EE/55*0.0/
0016            READ(5,1) NTAPE
0017            READ(5,151) K1, K2, NTC, NTP,NTW,NPH, KELC ,JZ, IM,NCUN, NDPN,NTMP
     X, MC ,IUSN, NUST, NBF
0018            READ(5,152) ED1 ,ND , EO, ELS, D,ANSW, PDPO, SR
0019            READ(5,153) FID
0020          1 FORMAT ( I10)
0021        151 FORMAT( 16I5)
0022        152 FORMAT( 8F10.4)
0023        153 FORMAT( 8E10.3)
0024            AKT=0.05
0025            AIKT= 1.0/AKT
0026            PI=3.141593
0027            EMASSI=1.60203E+16/9.108
0028            K3=K1*K2
0029            WN2=1.0/K2
0030            CJD=0.0
0031            EI=24.46
0032            EIS=21.4
```

```
FORTRAN IV G LEVEL 19                MAIN                DATE = 72125          01/28/29

0033                C1=2.*0.010549/4.003873
0034                ETV= SQRT(2.0*1.60203/9.108)*1.0E+08
0035                VTE= 1.0/(ETV*ETV)
0036                CALL  UNDERZ('OFF')
0037                PD26=PDP0*26.0
0038                MU0=(1.0E+09)/ETV
0039                CMU0=1.0E-09
0040                CMUPDP=CMUJ*PDP0
0041                IF (PDP0.GT.10.0)  CMUPDP=CMU0*10.0
0042                PDMU0=CMUPDP/ETV
0043                CLOSS=1.0
0044                NUP=0
0045                NON=0
0046                NONR=0
0047                IN0=0
0048                NKR=0
0049                NAS=0
0050                NES=0
0051                NSE=0
0052                NOZN=0
0053                NIZN=0
0054                NIN=0
0055                LK=0
0056                INIV=999999
0057                NMET=0
0058                NEXC=0
0059                ETAS=0.0
0060                ETEP=0.0
0061                NIP5=0
0062                NW=0
0063                N5IP=0
0064                DTAVS=0.0
0065                CSF2=0.0
0066                CNIO2=0.0
0067                CSF1=0.0
0068                CETAS=0.0
0069                CETEP=0.0
0070                NCOL=0
0071                WNMET=0.0
0072                WNSE=0.0
0073                WNEXC=0.0
0074                WNES=0.0
0075                WNAS=0.0
0076                WNIZN=0.0
0077                JE=48
0078                IE=JE
0079                DEN=3.54E+16*PDP0
0080                DETZ=D/JZ
0081                DZIV= 1.0/DETZ
0082                CONST=1.0/ALOG(2.0)
0083                ED2=0.447513
0084                ED3=19.8
0085                PM=ED2*AIKT
```

```
FORTRAN IV G LEVEL 18                    MAIN                    DATE = 72125          01/28/29

0086                AMI=2.0*AIKT/SQRT(PI*AKT)
0087                CALL   RNDINZ(994563211)
0088                SO=PDPO*SR*1.0E+16
0089                AITG=  CHUPDP /(AKT*PI)
0090                DE1=0.016
0091                DE2=0.355
0092                DE3=0.3
0093                DI1=1./0.016
0094                DI2=1./DE2
0095                DI3=1./DE3
0096                AMC1= SQRT(2.0*2.7183*AIKT)*ED2
0097                DO  11  I=6, 36
0098                EBL =(I-5)*DE2 -4.0
0099                EMV= EBL -0.1685
0100                EBD(I)= 2.0**EBL
0101                EG(I-1)= 2.0**EMV
0102             11 EDE(I)=EBD(I)*(2.0**DE2 -1.0)
0103                JE1=JE+1
0104                DO  12  I=37, JE1
0105                EBL= 7.005 +(I-36)*DE3
0106                EMV= EBL - 0.1405
0107                EBD(I)= 2.0**EBL
0108                EG(I-1)= 2.0**EMV
0109             12 EDE(I)= EBD(I)*(2.0**DE3 -1.0)
0110                DO  10  I=1, 5
0111                EDE(I)= 0.016
0112                EG(I)= EDE(I)*( -0.008
0113             10 EBD(I)= EDE(I)*(I-1)
0114                EDE(36)=29.69
0115                EG(48)= 1737.482
0116                IF  (NTAPE.NE.1)  GO  TO  2
0117                READ(10)  EAVG, DAVG, TME, NTW, NT, NTP, NION, DDEG, NSW,SWF, E,U
0118                READ(10) Z, WT, ANIOS,SF3
0119                NS=NSW*K2
0120                REWIND  10
0121                NT=NT+1
0122             2  CONTINUE
0123                IF (NTAPE.NE.2)  GO  TO  5
0124                READ(9 )  EAVG, DAVG, TME, NTW, NT, NTP, NION, DDEG, NSW,SWF, E, U
0125                READ(9 ) Z, WT, ANIOS,SF3
0126                NS=NSW*K2
0127                REWIND  9
0128                NT=NT+1
0129             5  CONTINUE
0130                IF  (NTAPE.NE.0)  GO  TO  122
0131                NION= SQRT(SO*10/AITG)
0132                 ANEAM=NION*0.33333/DEN
0133                CALL  SORCE2(NTP, EAVG, NTW, PM, FID)
0134                AL=ALOG(EAVG)
0135                DAVG=AL
0136                DDEG=DAVG*ND/SQRT(EAVG)
0137                 SF2=1.0
0138                IF  (NION.NE.0)  SF2=NTW/NION
```

```
FORTRAN IV G LEVEL 18                MAIN                DATE = 72125        01/28/29

0139                SN=SO*AL*ND*SF2/(ETV*SQRT(EAVG))
0140                NSW=ANSW+0.5
0141                IF (SN.GT.ANSW)  NSW=SN +0.5
0142                NS=NSW*K2
0143                SWF=NSW/SN
0144                SF=SWF
0145                ANIOS=NION
0146                CALL  SOURCE(NS,NTP)
0147                SF1=SF2
0148                SF3=SF2
0149                TME=0.0
0150              122 CONTINUE
0151                SF=SWF
                C  GENERATE SOURCE PARTICLES TO INITIALIZE THE PROGRAM
0152                ANEAN=NION*0.33333/DEN
0153                WRITE(6,70)
0154               70 FORMAT('1','PROGRAM INITIALIZATION AND THE INITIAL PARAMETERS',//)
0155                WRITE(6,71) AITG, NION, NTP, AL,EAVG
0156               71 FORMAT(' ', 'AITG =',E14.6, 3X,'NION =',E14.6,3X,'NTP =',I12,3X,
                     1'DCEN(EO) =', F14.6,3X, 'EAVG= ',F14.4,/)
0157               72 FORMAT(' ',8X,'DCOL',13X,'DCEN',13X,'ENERGY',11X,'Z',14X,'U', 10X,
                     X'WEIGHT',/)
0158               75 FORMAT(' ',F14.6,3X, F14.6, 3X, F14.5,3X, F12.4,3X,F12.4,3X,F13.4)
0159                WRITE(6,72)
0160                RIP=RAN3Z(0)
0161                I=1
                C WITHIN THE OUTER I LOOP THE I TH PARTICLE HISTORY
                C  GEOMETRIC ROUTINE ENTERING HERE
0162             7777 IF ((E(I).LE.0.01.OR.(E(I).GT.EO))  GO TO  9008
0163                DCEN=DCEG*SQRT(E(I))
0164             7776 IF ((NIZN.GT.0).AND.(I.GT.NTP))  DCEN= DIZN(NOZN+1)
0165                NFLG=0
0166                U(I)=U(I)*IUSN
0167                LFG2=0
0168                ETAG1=1.0
0169                IF (E(I).GT.EO) ETAG1=-1.0
0170             3000 IF ((Z(I).LE.0.01.OR.(Z(I).GT.01.OR.(E(I).LE.0.0)) GO TO 9000
0171                J=INT(Z(I)*DZIV) +1
0172                RI=RAN3Z(0)
0173                ALD=ALDA(E(I))
0174                DCOL=ALD*ABS(ALOG(RI))
0175                IF (NFLG.EQ.1)  GO  TO  809
0176                IF(MOD(I,IM).EQ.0)  WRITE(6,75)   DCOL,DCEN,E(I),Z(I),U(I),MT(I)
0177                DCOL2=ABS(ALOG(1.-RI))*ALD
0178                DTOT=DTOT +DCOL+DCOL2
0179                ETOT=ETOT+E(I)
0180                IF (NDPN.LT.2)  GO  TO  809
0181                DCEN2=DCEN
0182                EIO=E(I)
0183                UIO= U(I)*IUSN
0184                ZIO=Z(I)
0185              809 NFLG=1
0186              900 IF (DCEN.LT.DCOL)  GO  TO  902
```

```
FORTRAN IV G LEVEL 18          MAIN          DATE = 72125          01/26/29

0187          901 DCEN=DCEN- DCOL
0188              Z(I)=Z(I) +DCOL*U(I)
0189              NCOL=NCOL+1
0190              IF (MOD(NCOL,MC).EQ.0) RIP=RANDZ(0)
0191              IF (MOD(NCOL,MC).NE.0) RIP=1.0-RIP
0192              CALL  COLPRO(I, DCEN, NTP, RIP, NUST, &9008, &9009)
0193              GO  TO  3 0
0194          902 Z(I)= Z(I) +DCEN*U(I)
0195               IF ((Z(I).LE.0.0).OR.(Z(I).GT.D)) GO  TO  9008
0196              J=INT(Z(I)*DZIVI+1
0197              IF (J.GT.JZ)  J=JZ
0198              K= INDEXE(E(I))
0199              IF (K.GT.JE) K=JE
0200              IF (K3.EQ.1)  GO  TO  904
0201              ETAG2=1.0
0202              IF (E(I).GT.ED1)  ETAG2=-1.0
0203              IF ((ETAG1*ETAG2).GT.0.0)  GO  TO  904
0204              IF (ETAG2.GT.0.0)  GO  TO  903
0205              NUP=NUP+1
0206              ESE(NUP)=E(I)
0207              WT(I)=WE(I)/K3
0208              WSE(NUP)=WT(I)
0209              GO  TO  904
0210          903 IF((LFG2.EQ.1).OR.(NDPN.LT.2))  NONR=NONR+1
0211              IF((LFG2.EQ.1).OR.(NDPN.LT.2))  IRR(NONR)=I
0212              NON=NON+1
0213              IF (MOD(NON,K3).NE.0)  GO  TO  9045
0214              WT(I)=WT(I)*K3
0215          904 NEZ(K,J)= NEZ(K,J) + WT(I)
0216              EKZ(K,J)=EKZ(K,J)+E(I)*WT(I)
0217         9045 IF((LFG2.EQ.1).OR.(NDPN.LT.2))  GO  TO  9009
0218              LFG2=1
0219              E(I)=EIO
0220              U(I)=UIO
0221              Z(I)=ZIO
0222              DCEN=DCEN2
0223              DCOL=DCOL2
0224              GO  TO  900
0225         9008 NES=NES+1
0226              IF (E(I).LT.ED1) WT(I)=WT(I)/K3
0227              WNES=WNES +WT(I)
0228              IES(NES)=I
0229              IF  (E(I).GT.0.0)  ETEP=ETEP+E(I)*WT(I)
0230         9009 I= I+1
0231              LFG2=0
0232              IF (I.LE.NTP)  GO  TO  7777
0233              IF ((I.GT.NTP).AND.(NIZN.GT.0))  NOZN= NOZN-1
0234              IF (NOZN.LT.0)  NIN=1
0235              IF ((NIZN.GT.0) .AND.(NIN.EQ.0))  GO  TO  7778
0236              NIZN=WNIZN +0.5
0237              NNS=WNES +0.5
0238              NASW=WNAS  +0.5
0239              WNAS=WNAS/KELC
```

```
FORTRAN IV G LEVEL 18          MAIN          DATE = 72125          01/26/29

0240                WNSE=WNSE/KELC
0241                NSEW=WNSE +0.5
0242                DO  191  M=1,JE
0243                DO  191  N=1,JZ
0244                EE(M)=EE(M) + EKZ(M,N)
0245          191   NE(M)= NE(M) + NEZ(M,N)
0246                IF ((K3.EQ.1).OR.(NUP.EQ.0))  GO  TO  24
0247                 N3=K3/JZ
0248                IF   (N3.LT.1)  M3=1
0249                NK3=K3-1
0250                IF (NK3.LT.JZ) IZ=NK3
0251                 IF (N3.GE.1)  M3=N3
0252                DO  23  K=1,NUP
0253                K=INDEXE(ESE(N))
0254                EE(K)=EE(K) + (K3-1)*ESF(N)*WSE(N)
0255                NE(K)=NE(K) +(K3-1)*WSE(N)
0256           22   IF (NK3.GE.JZ)  IZ=JZ
0257                DO  21  L=1,IZ
0258                NREZ(L)=NREZ(L) + M3*WSF(N)
0259           21   EZ(L)=EZ(L) + ESE(N)*M3*WSE(N)
0260                NK3=NK3 -JZ*M3
0261                IF (NK3.LT.JZ)  IZ=NK3
0262                M3=1
0263                IF (NK3.GT.0)  GO  TO  22
0264           23   CONTINUE
0265           24   CONTINUE
0266                DO  192  M=1, JZ
0267                DO  1915  N=1,JE
0268                NREZ(M)=NREZ(M) + NEZ(N,M)
0269         1915   EZ(M)=EZ(M) +EKZ(N,M)
0270          192   CONTINUE
0271                NTPN=NTP+NIZN-NRS-NAS
0272                NTPW=0.0
0273                ETOL=0.0
0274                DO  14  M=1,JE
0275                ETOL= ETOL +EE(M)
0276           14   NTPW=NTPW +NE(M)
0277                EWTOL=ETOL
0278                NTPT=NTPN
0279                DAVGO= DAVG*WO
0280                DAVS=(DAVGO*1000.)**NTMP
0281                DSAVS=DSAVS+ DAVS
0282                DISO=1.2/DSAVS
0283                ANTPO = 1.0/NTPW
0284                N5TP= N5TP + NTPW +0.5
0285                DTAVS=DTAVS +DAVS
0286                NTP5=NTP5+ NTPN*DAVS
0287                NTPO=NTP
0288                EWAVG=EWTOL*ANTPO
0289                NSO=NS
0290                NSWO=NSO*WN2 +0.5
0291                NTP=NTP+NIZN
0292                NTPP=NTP
```

```
FORTRAN IV G LEVEL 18                    MAIN                    DATE = 72125          01/28/29

0293                DAVG=DTOT*0.5/NTP
0294                EAVG= ETOT/NTP
0295                DEG=DDEG/ETV
0296                TME=TME+DEG
0297                TMS=TMS+DEG
          C IF SOURCE PARTICLES SO(NS)C NES THEN ELIMINATE AS MANY RESIDUE PARTICLES AT TH
0298                NTPS=NTPW + NWS +NASW
0299                DCEN1= ALOG( EAVG)
0300                NTW=NTPW +.5
0301                DDEG=DAVG*ND/SQRT( EAVG)
0302                SF2=1.0
0303                IF (ANIOS.NE.0)  SF2=NTPS/ANIOS
0304                ESAVG=0.0
0305                IF(WNES.NE.0.0)  ESAVG= ETEP/WNES
0306                EASAVG=0.0
0307                IF (WNAS.NE.0)  EASAVG=ETAS/WNAS
0308                SWFO=SWF
0309                BF=1.0
0310                IF  ((WNES+WNAS).GT.0.0)  BF=(WNIZN+SN)/(WNES+WNAS)
0311                CN5S=CN5S+ SN*DAVS
0312                CSF2= CSF2 +(ETEP+ETAS+40.*WNIZN)/TMS
0313                IF((MOD(NT,5).EQ.0).AND.(NT.NE.0)) SF1=CSF2 /(SO*EC*5.)
0314                CSF1= CSF1 + SF2*DAVS
0315                 IF ((MOD(NT,5).EQ.0).AND.(NT.NE.0))  SF3=CSF1*DI5G
0316                 IF ((MOD(NT,5).EQ.0).AND.(NT.NE.0).AND.(SF3.LT.SF1*0.01)) SF3=
               XSQRT(SF1*SF3)
0317                WGHT=SF1/SF3
0318                SN=SO*DDEG*SF3*WGHT/ETV
0319                NSW=ANSW+0.5
0320                IF (SN.GT.ANSW)  NSW=SN +0.5
0321                SWF=NSW/SN
0322                NS=NSW*K2
0323                SF=SWF
0324                NMEW=WNMET +0.5
0325                NEXW=WNEXC +0.5
0326                W5VLE=W5VLE+ WVALUE*DAVS
0327                CETEP=CETEP + ETEP*DAVS
0328                CETAS=CETAS +ETAS*DAVS
0329                CN5IZ=CN5IZ+WNIZN*DAVS
0330                CNES=CNES+WNES*DAVS
0331                CNAS=CNAS+WNAS*DAVS
0332                CNMET=CNMET+WNMET*DAVS
0333                CNEXC=CNEXC+WNEXC*DAVS
0334                CESAVG=CESAVG+ ESAVG*DAVS
0335                CEASG=CEASG+EASAVG*DAVS
0336                CLOSS= CLOSS+(WNIZN-WNES)*DAVS/(TMS*SF3*WGHT)
0337                WVLF1=W5VLE*DI5G
0338                ALOSS=CLOSS*DI5G
0339                ANSVG=CN5S*DI5G
0340                ANIZVG=CN5IZ*DI5G
0341                AETEP=CETEP/DTAVS
0342                AETAS=CETAS/DTAVS
0343                IF (ANIZVG.NE.0)  WVLE2=(EO*ANSVG-AETEP-AETAS)/ANIZVG
```

```
FORTRAN IV G LEVEL 18          MAIN          DATE = 72125          01/28/29

0344                ANES=CNES/DTAVS
0345                ANAS=CNAS/DTAVS
0346                AESAVG=CESAVG*DI5G
0347                AEASG=CEASG*DI5G
0348                ANNET=CNNET*DI5G
0349                ANEXC=CNEXC*DI5G
0350                ANS=ANES+ ANAS -ANIZVG
0351                IF (ANS.LE.0.0)  ANS=NSW/SNF
0352                IF (ANIZVG.NE.0.0)  WVALUE= (SN*EO- AETEP-AETAS)/ANIZVG
0353                IF (WNIZN.NE.0.)  WVC=(ANS*EO -ETEP-ETAS)/WNIZN
0354                IF (NS.GE.1)  TMS=0.0
0355                NET=NES+NAS
0356                CALL  SOURCE(NS,NTP)
0357                 IF (NET.GT.0)  CALL  FILHOL(NTP,NES, NAS)
0358             90 NIZNT= NIZNT +NIZW
0359                IF  (NUP .GT.0)  CALL  SEPROC(NTP, K3)
0360                IF (NONR.GT.0)  CALL  RRPROC(NTP,K3)
0361                NAST= NAST +NWS
                 C  INTRODUCE THE SOURCE PARTICLES FOR THE NEXT CYCLE
                 C EVERY FIFTH CYCLE RENORMALIZATION AND OUTPUT QUANTITIES ARE PRINTED
                 C RENORMALIZATION PROCESS OF NION
0362                ANSTP= 1.0/NSTP
0363                ANTPS=DSAVS/NTPS
0364                NSTPN=NSTP
0365                DO  16  M=1,JZ
0366                CNREZ(M)= CNREZ(M) + NREZ(M)*DAVS
0367             16 CEZ(M)= CEZ(M) + EZ(M)*DAVS
0368                DO  18  M=1,JZ
0369                FNREZ(M)=NREZ(M)*ANTPO
0370             18 EZ(M)= EZ(M)/ NREZ(M)
0371                SUMNE=0.0
0372                DO   193  M=1,JE
0373                SUMNE= SUMNE + NE(M)
0374                TNE(M)= TNE(M) + NE(M)/EDE(M)
0375            193 FNE(M)=NE(M)*ANTPO/EDE(M)
0376                EAV3=0.
0377                DO  1755  M=1, JE
0378                EAV3= EG(M)*FNE(M)*EDE(M) +EAV3
0379           1755 CNE(M)= TNE(M)*ANSTP
0380                CNIO4=CMUPDP*2.0*AIKT/SORT(AKT*PI)
0381                ANION=0.0
0382                ANIO2=0.0
0383                DO  20  M=1,JE
0384                ANIO2= ANIO2 + FMI(EG(M),CNE(M))*EDE(M)
0385                ANION= ANION + FMI(EG(M),FNE(M))*EDE(M)
0386             20 CONTINUE
0387                DNION=ANION*CNION
0388                DNIO2=ANIO2*CNION
0389                CJD=ALOSS
0390                IF (ALOSS.LT.0.0)  CJD=0.0
0391                IF (ABS(CJD).GT.110.*SO)  CJD=110.*SO
0392                TNION=ABS(SO+CJD)/DNION
0393                TNIO2=ABS(SO+CJD)/DNIO2
```

```
FORTRAN IV G LEVEL 18                    MAIN                    DATE = 72125          01/28/29

0394                TNIO2=SQRT(TNIO2)
0395                TNION=SQRT(TNION)
0396                CNIO2=CNIO2 + TNIO2*DAVS
0397                ANIOS=CNIO2*DISG
0398                WF=1.0
0399                IF (ANES*ANAS.NE.0.0)  WF=(ANIZVG*ANSVG)/(ANES*ANAS)
0400                IF  (MOD(NT,NPM).NE.0)  GO  TO  1790
0401                NION=ANIOS
0402                IF ((WF.GT.0.005).AND.(WF.LT. 200.)) NION=ANIOS*WF**NRF
0403             1790 CONTINUE
0404                DAVT=DAVG*ND
0405                ALPHA=WNIZN/(NTPW*DAVT)
0406                ANEAM=NICN*0.33333/DEN
0407             1785 FORMAT('0','ANION=',E13.4,3X,'ONION=',E13.4,3X,'TNION=',E13.4,3X,
                    X'ALPHA=',F12.5,4X,'LEAK=',E13.4,/)
0408                WRITE(6,1785)  ANION, ONIO2, TNIO2, ALPHA, ALOSS
0409                WRITE(6,171)  EO, PDPO, D, DETZ, DEG, NDPN
0410              171 FORMAT('1','GAS: HE',2X,'SOURCE ENERGY:',F5.0,3X,'PRESSURE:',F4.0,
                    X 'TORR',3X,'SLAB THICKNESS:',F4.1,' CM',' DX:',F4.1,' CM',4X,'DELT
                    X=',E12.5,3X,'#PROC=',I2,/ )
0411                WRITE(6,180) NT, TME, NS,NSW
0412             180 FORMAT(' ','NUMBER OF TIME CYCLES=',I6,3X,'TIME =', F12.5,' SEC',
                    1X, 'GENERATED SOURCE PART =', I5,3X, 'WGTED # OF SOURCE PART.=',
                    I15 )
0413                WRITE(6,1805) DAVT, DCEN1,EAVG,EWAVG, EAV3
0414             1805 FORMAT(' ','DAVG =',F12.5,3X, 'DCEN =',F12.5,3X,'OVERALL EAVG=',
                    1F10.5,3X,' WGTED EAVG=',F10.5,3X, 'EAVG CHECK=',F10.5,/)
0415                WRITE(6,191)
0416              191 FORMAT(' ', 1X,'NUMBER DISTRIBUTION',2X, 'FLUX DENSITY',4X,'AVERAGE
                    1E ENERGY',3X,'CUMULATED # DISTR',3X,'ENERGY INTERVAL',2X,'CUM FLUX
                    XDEN',4X,'#DEN-DEVIATION',/)
0417                CESUM=0.0
0418                DO  281  M=1, JE
0419                CESUM=CESUM + CNE(M)*EDE(M)
0420                IF (NE(M).GT.0)  DEAVG=EG(M)-EE(M)/NE(M)
0421                IF (NE(M).LE.0)  DEAVG=0.0
0422                 IF (NE(M).EQ.0)  DNAVG=0.
0423                FNOV=CNE(M)-FNE(M)
0424                 VFE=SQRT(EG(M))*ETV
0425                FLX1=FNE(M)*VFE
0426                FLX2=CNE(M)*VFE
0427                IF (NE(M).EQ.0)  GO  TO  281
0428                IF(((M.EQ.1).AND.(DEAVG.GT.0.0)).OR.((M.EQ.JE).AND.(DEAVG.LT.0.0))
                    X) DNAVG=-DEAVG*FNE(M)*0.5/EDE(M)
0429                 IF ((DEAVG.LT.0.0).AND.((M+1).LE.JE)) DNAVG=-DEAVG*(FNE(M+1)-FNE(M
                    X))/(EG(M+1)-EG(M))
0430                 IF ((DEAVG.GT.0.).AND.(M.GT.1)) DNAVG=DEAVG*(FNE(M)-FNE(M-1))/(EG(
                    XM)-EG(M-1))
0431              281 WRITE(6,182) M,FNE(M),FLX1,EG(M),CNE(M),EDE(M),FLX2 , DNAVG
0432              182 FORMAT(' ',I3,E12.5,5X,E12.4, 5X, F12.6,8X, E12.5,6X, F12.6,4X,E12
                    X.5,4X,E12.4)
0433                DET=DDEG/ETV
0434                WRITE(6,183) NTPP, NSTPN,NTPW ,DET, DAVGO, NCOL
```

```
FORTRAN IV G LEVEL 18          MAIN          DATE = 72125          01/28/29

0435      183 FORMAT(' ',//,3X,'TOTAL # OF PARTICLES PROCESSED =',I5,5X,'CUMULAT
              1ED # OF PARTICLES =',I10,3X,'WGTED TOTL # OF PARTICLES=',F12.3,//,
              X' NEW DT=',E12.5,4X,'OLD DAVG=',F12.5,4X,'# OF COLLISIONS=',I5,//)
0436            WRITE(6,187)
0437      187 FORMAT(' '    ,1X, 'ZONE NUMBER', 5X, 'AVERAGE ENERGY',3X,'NUMBER O'
              1F ELECTRONS',3X,'CUMULATIVE # DEN(Z)',3X,'CUMU. AVG ENERGY',3X,
              1'ZONE # DENSITY',/)
0438            DO 282  M=1, JZ
0439            ANZ=CNREZ(M)*ANSTP*DISG
0440            PCZ=CEZ(M)/CNREZ(M)
0441      282 WRITE(6,184) M, EZ(M), NREZ(M), ANZ, PCZ,ENREZ(M)
0442      184 FORMAT(' ', 5X, I5, 5X, F14.6,8X,F14.6,  5X, F14.5, 7X,F14.6,7X,
              1E12.5)
0443            WRITE(6,185)  NION,WNIZN,NIZN, ANIZVG,WNES, NES,ANES, WNAS,NAS,
              XANAS, NTP, K1, K2, WF,BF
0444      185 FORMAT('0', 'NUMBER OF POSITIVE ION=',E14.6,//,' NUMBER OF IONIZAT
              1IONS OCCURED=',F9.3,5X,'ACTUAL #=',I5,3X,'CUMULATIVE AVG=' ,F10.3,
              X//,' # OF ESCAPED PARTICLES=',F9.3,5X,'ACTUAL#=',I5,3X,'CUMULATIVE
              X AVG=',F9.2,//,' # OF ABSORBED PARTIC=',E12.4,3X,'ACTUAL #=',I5,3X
              X,'CUMULATIVE AVG=',F9.2,//, ' TOTAL # OF PARTICLES IN THE NEW CYCL
              XE  ',I5,//,' K1=',I2, 3X,'K2=',I2,     4X,' BALANCE   FACTOR WF=',
              XF10.5,3X,'BF2=', F10.5,/)
0445            WRITE(6,188) EO1, NPH, ND, KELC
0446      188 FORMAT(' ENERGY DIVIDING LINE FOR  RR PLAY E=',F10.4,4X,'EVERY',I2
              X,'TH CYCLE CHANGE N+',//,' TIME INTERVAL MULTIPLY FACTOR=',F9.4,
              X3X, 'ELASTIC XSEC / BY',I2,/)
0447            NDR=NDN-NDN/K3
0448            WRITE(6,186) NSO, NSW, ANSVG,NDR,WNSE,WNMET,ANMET,WNEXC,ANEXC,NUP,
              XWGHT
0449      186 FORMAT(' ',    ' SOURCE PARTICLES GENERATED OLD =',I10,//,
              1' SOURCE PARTICLES (WEIGHTED) IN THE NEW CYCLE =',I10,3X, 'CUMULAT
              XIVE AVG=',F10.3,//
              1,' # OF PARTICLES KILLED OFF IN RR PROC=',I10,//,
              1' # OF SUPER ELASTIC COLLISION=',F10.4,//,
              1                          ' # OF EXCITATIONS TO META STABLE LEVEL='
              X,F10.4, 3X, 'CUMULATIVE AVG=', F9.3,
              1                          //,' # OF EXCITATIONS TO OTHER LEVELS=',F10.4,
              X4X, 'CUMULATIVE AVG=',F9.2,//,' # OF PARTICLES CROSS OVER TO HIGH
              XENERGY REGION NUP=',I5,4X,'OVERALL WT=',E12.4,/)
0450            WRITE(6,195)  EF, EDP, PDPO ,SF1, SF2, SF3
0451      195 FORMAT (' E-FIELD=', F9.2,'V/CM', 4X,'E/P=',F9.2,4X,'P/PO =',F7.1,
              X3X, 'SF1=', E12.4,3X, 'SF2=',E12.4,3X, 'SF3=', E12.4,/)
0452            WRITE(6,170) SO,WVC,SWEO,WVALUE,WVLE1, WVLE2,ESAVG, AESAVG,EASAVG,
              XAEASG
0453      170 FORMAT(' SOURCE RATE =',E11.3,' PER SEC',4X,'W-VALUE(LOCAL  )=',
              XF10.3, 3X,'SOURCE RED.FACT=',F13.4,//,' W-VALUE=',
              XF10.3,4X,'CUM AVG WV1=',F9.3,3X,'CUM AVG WV2=',F9.3,
              1      //,' AVERAGE ENERGY OF ESCAPED PARTICLES=',F12.5,3X,'CUMULAT
              XIVE AVG=',F11.4,//,' AVERAGE ENERGY OF ABSORBED PARTICLES=',F12.5,
              X3X, 'CUMULATIVE AVG=',F11.4,/ )
0454            CKENE=SUMNE+ANTPO
0455            WRITE(6,189)CKENE, CESUM, TNION, TNIO2
0456      189 FORMAT(' ','THE SN # CHECK =',F12.4,3X,'THE CN # CHECK=',F12.6,//,
```

```
FORTRAN IV G LEVEL 18                    MAIN                    DATE = 72125          01/28/29

                     1' CALCULATED N* USE P=',E14.6,3X,'CALC. N* USE CN =',E14.6,3X, 'AR
                     XTIF CN*=',E14.6,//)
0457                 IF ((MOD(NT,5).NE.0).OR.(MOD(NT,10).EQ.0)) GO TO 3
0458                 WRITE(9) EAVG, DAVG, TME, NTW, NT, NTP, NION,DDEG, NSW,SWF, E, U
0459                 WRITE(9) Z, WT ,ANIOS, SFS
0460                 ENDFILE 9
0461                 REWIND 9
0462               3 CONTINUE
0463                 IF (MOD(NT,10).NE.0) GO TO 4
0464                 WRITE(10) EAVG, DAVG, TME, NTW, NT, NTP, NION, DDEG, NSW,SWF, E, U
0465                 WRITE(10) Z, WT ,ANIOS, SF3
0466                 ENDFILE 10
0467                 REWIND 10
0468               4 CONTINUE
0469                 IF (MOD(NT,NCUM).NE.0) GO TO 6666
0470                 NSTP=0
0471                 NTPS=0
0472                 DSAVS=0.0
0473                 NAST=0
0474                 NITNT=0
0475                 RUNLE=0.0
0476                 RUNSS=0.0
0477                 CSF1=0.0
0478                 CNIO2=0.0
0479                 CNSIZ=0.0
0480                 CNSS=0.0
0481                 CNMET=0.0
0482                 CNEXC=0.0
0483                 CESAVG=0.0
0484                 CEASG=0.0
0485                 DO 54 M=1,JZ
0486                 CNREZ(M)=0.0
0487              54 CEZ(M)=0.0
0488                 DO 55 M=1,JE
0489                 TNE(M)=0.
0490              55 CNE(M)=0.0
0491                 IF (MOD(NT,10).NE.0) GO TO 6666
0492                 DTAVS=0.0
0493                 CETEP=0.0
0494                 CETAS=0.0
0495                 CNES=0.0
0496                 CNAS=0.0
0497           6666 CONTINUE
0498                 NT= NT + 1
0499                 DO 56 M=1,500
0500                 DIZN(M)=0.0
0501                 IRR(M)=0
0502                 IES(M)=0
0503              56 IAS(M)=0
0504                 DO 51 N=1,50
0505                 WSE(N)=0.0
0506              51 ESE(N)=0.0
0507                 I=1
```

```
FORTRAN IV G   REL 18          MAIN          DATE = 72125          01/28/29

      C  REINITIALIZE THE OUTPUT QUANTITIES
0508                DO  52  M=1, JE
0509                FE(M)=0.
0510             52 NE(M)= 0.0
0511                DO  53  M=1,JZ
0512                NREZ(M)= 0.0
0513                EZ(M)=0.0
0514                DO  53  N=1,JE
0515                EKZ(N,M)=0.0
0516             53 NEZ(N,M)=0.0
0517                NIZN=0
0518                NOZN=0
0519                NIN=0
0520                NES=0
0521                NSE=0
0522                NCOL=0
0523                NAS=0
0524                DTOT=0.0
0525                NMET=0
0526                NEXC=0
0527                WNAS=0.0
0528                WNSF=0.0
0529                WNES=0.0
0530                WNIZN=0.0
0531                WNMET=0.0
0532                WNEXC=0.0
0533                ETAS=0.0
0534                ETEP=0.0
0535                ETOT=0.0
0536                NUP=0
0537                NDN=0
0538                NONP=0
0539                INIV=INIV+ 1100
0540                CALL   RNPINZ(INIV)
0541                WRITE(6,721)
0542                IF (NT.LE.NTC)  GO  TO  7777
0543             9999 STOP
0544                DEBUG  SUBCHK
0545                END
```

VITA

Benjamin Shaw-hu Wang was born in [redacted]. He is the son of Mr. C. H. Wang, the former Chinese Army General now the Congressman of the Republic of China. He received the B.S. degree in Electrical Engineering from the National Taiwan University of Taipei, Taiwan, China in 1960, he then served as an ensign in Chinese Naval Academy for one and a half years. He came to the United States in September, 1962 and enrolled in the University of Missouri at Rolla, Missouri. After he received the M.S. degree in Electrical Engineering in 1964, he joined the Westinghouse Research Laboratories and served as a research engineer for three and a half years.

In September 1967, he came to the University of Illinois at Urbana-Champaign to undertake studies for a Ph.D. in Computer Science Department. He has been a research assistant and teaching assistant in the Department of Computer Science and a research assistant in the Nuclear Engineering Program. He is the member of Kappa Mu Epsilon and Pi Mu Epsilon the two national honor societies of mathematics.