

AD-A084 892

PARKER (LEE W) INC CONCORD MASS

F/6 9/2

TIME-DEPENDENT COMPUTER MODEL OF PLASMA SPACE CHARGE INTERACTIO--ETC(U)

DEC 79 L W PARKER

F19628-78-C-0149

NL

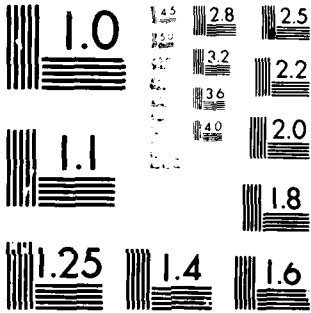
UNCLASSIFIED

AFGL-TR-80-0018

[ or ]

AD  
ACRLA ROL

END  
DATE  
FILMED  
6-80  
DTIC



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

# LEVEL II

154  
12

ADA 084892

## TIME-DEPENDENT COMPUTER MODEL OF PLASMA SPACE CHARGE INTERACTIONS WITH A FINITE-CYLINDRICAL SPACECRAFT

Lee W. Parker

Lee W. Parker, Inc.  
252 Lexington Road  
Concord, Massachusetts 01742

31 December 1979

DTIC  
ELECTED  
MAY 30 1980  
D  
C

Final Report  
1 April 78 - 30 September 79

AIR FORCE GEOPHYSICS LABORATORY  
AIR FORCE SYSTEMS COMMAND  
UNITED STATES AIR FORCE  
HANSCOM AFB, MASSACHUSETTS 01731

THIS DOCUMENT IS BEST QUALITY PRACTICES.  
THE COPY FURNISHED TO DDC CONTAINED A  
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT  
REPRODUCE LEGIBLY.

This document has been approved  
for public release and sale; its  
distribution is unlimited.

FILE COPY

80 5 27 291

Qualified requestors may obtain additional copies from the Defense Documentation Center. All others should apply to the National Technical Information Service.

## **DISCLAIMER NOTICE**

**THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.**

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

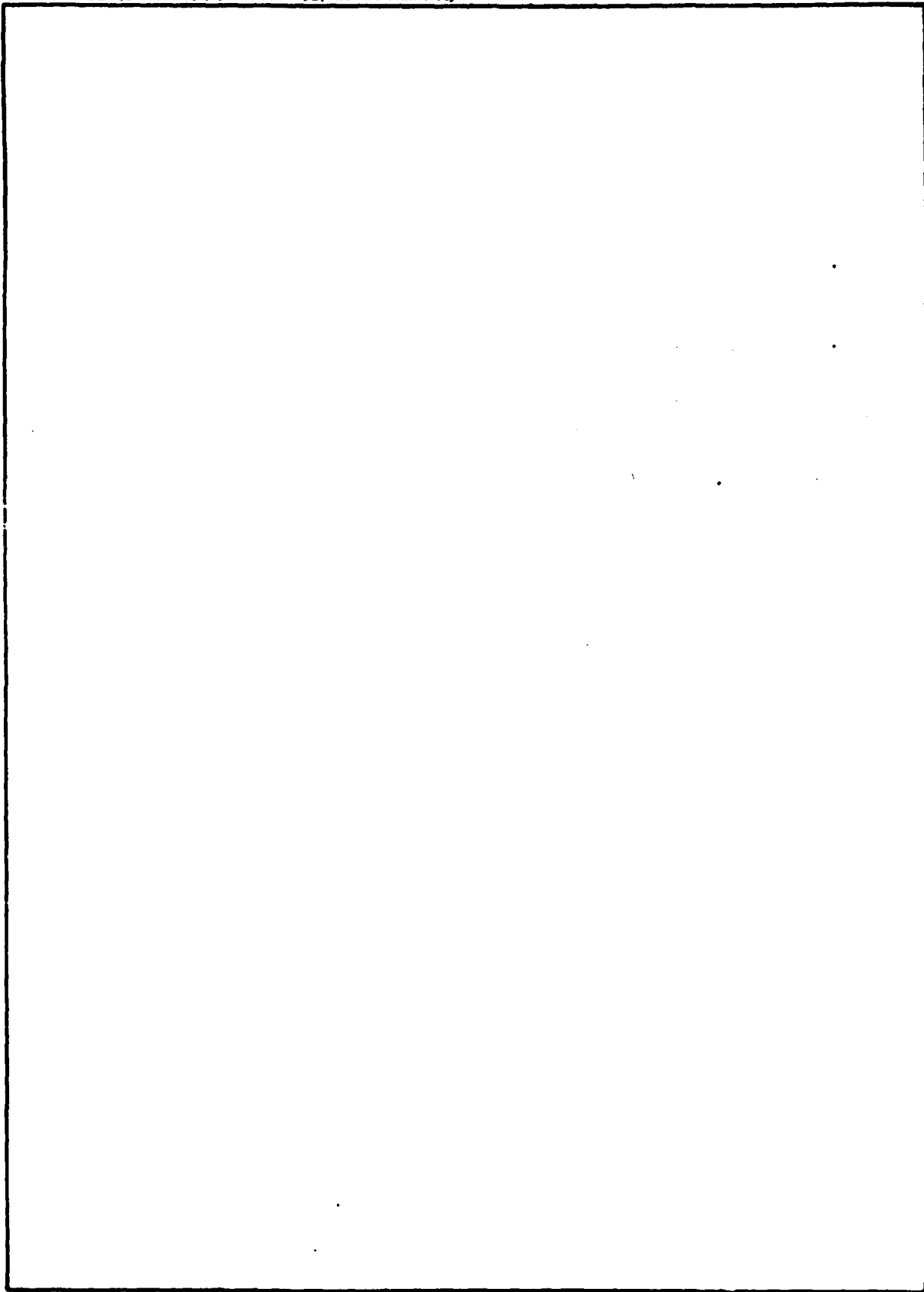
19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
18 1. REPORT NUMBER AFGL TR-80-0018	2. GOVT ACCESSION NO. AD-A084 892	3. RECIPIENT'S CATALOG NUMBER 9
6 4. TITLE (and Subtitle) TIME-DEPENDENT COMPUTER MODEL OF PLASMA SPACE CHARGE INTERACTIONS WITH A FINITE-CYLINDRICAL SPACECRAFT.	5. TYPE OF REPORT & PERIOD COVERED Final Report, 1 Apr 78 - 30 Sep 79	
10 7. AUTHOR(s) Lee W. Parker	8. CONTRACT OR GRANT NUMBER(s) F19628-78-C/0149	15
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lee W. Parker, Inc. 252 Lexington Road Concord, Massachusetts 01742	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62101F 766107AE 17 07	
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Geophysics Laboratory Hanscom AFB, Massachusetts 01731 Monitor/Allen G. Rubin/PHG	12. REPORT DATE 31 Dec 79	11
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	13. NUMBER OF PAGES 85	12 86
	15. SECURITY CLASS. (of this report) Unclassified	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) computer model spacecraft charging time dependent cylindrical spacecraft space charge		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A computer program has been developed for the study of the time-depend- ent behavior of the sheaths and charging effects of spacecraft with finite-cylin- der (pillbox) geometry. The behavior of ions and electrons is simulated by following representative computer particles in time. During each time step the particles are moved, their space charge is computed, and the electric potential distribution is updated. The development is original. A number of sample runs show that current collection and space charge behavior can be computed as functions of time.		

DD FORM 1 JAN 73 1473

Unclassified 391800

net

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)



SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

## INTRODUCTION

This project is concerned with the numerical simulation of the time-dependent behavior of spacecraft sheaths and charging effects. The objectives are to develop a time-dependent simulation model for plasma-spacecraft interactions, and to apply this model to a geometrical representation of the SCATHA spacecraft. An appropriate representation is that of a short right-circular cylinder (or "pillbox"), of comparable length and diameter (see Fig. 1), with azimuthal symmetry so that the potential and charge distributions can be defined on a grid in  $r$ - $z$  space.

A computer program has been developed for the study of the time-dependent sheath. A grid is used to define the spatial distributions of potential and charge density in the space around the spacecraft. Figure 2 illustrates the nature of the  $r$ - $z$  grid representation used. The geometry is axially-symmetric, with the axis shown as the vertical dotted boundary line on the left, labelled "West". The boundary condition representing the condition on the potential at infinity is applied to the other boundary lines of the grid. The inner boundary represents the satellite surface, on the grid points of which the surface potentials are defined. Associated with each spatial grid point is a volume of revolution in the shape of a torus of rectangular cross-section (shown as shaded boxes surrounding some of the grid points). Only 24 grid points are shown in Fig. 2, for the purpose of clarity. In an actual problem many more points are used.

An important feature of this grid representation is that the zoning is nonuniform. This allows for fine zoning in the regions of interest, e.g., where there are large gradients, and coarse zoning elsewhere, and has the advantage of optimum use of a given number of grid points, and therefore computer efficiency in large problems.

The plasma electrons and ions are simulated by a number of discrete "computer particles" injected through the outer grid boundaries. These particle trajectories are followed step by step through the grid. Similarly, emitted electrons and/or ions are injected from the inner (spacecraft) surface. Each simulation particle represents a definite number of real particles, retaining the charge-to-mass ratio of the real particles. The density

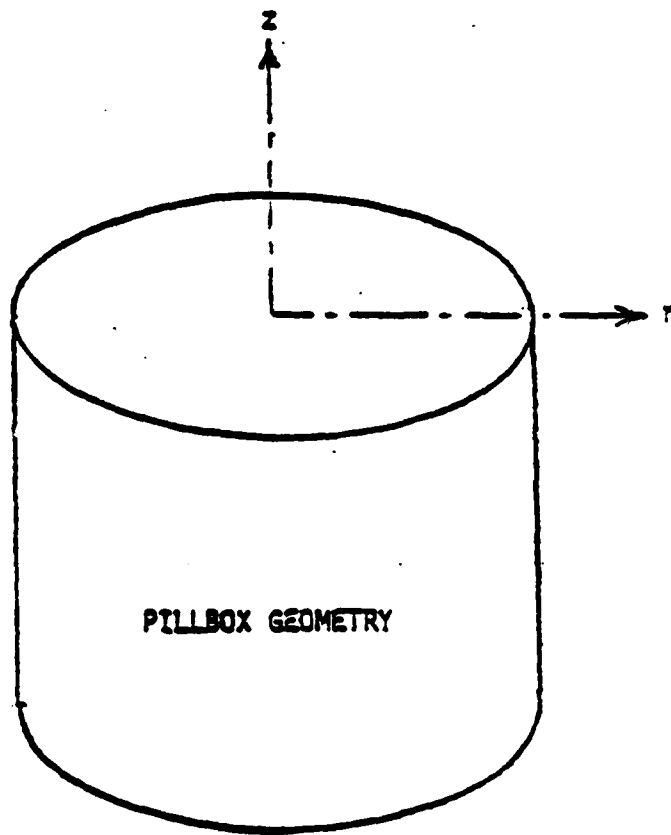


FIG 1. SPACECRAFT GEOMETRY

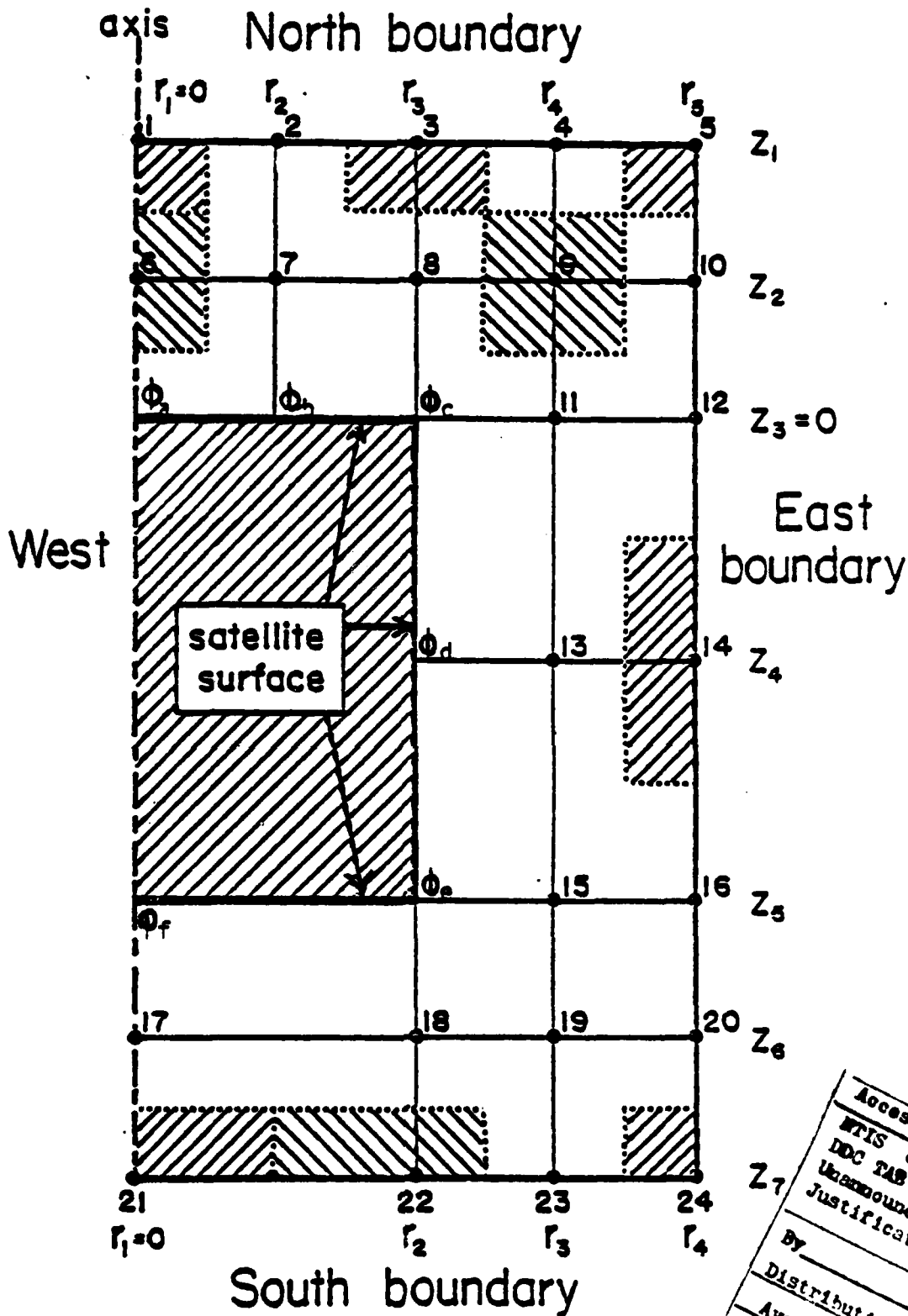


FIG. 2. GRID REPRESENTATION

Accession For	
DTIC	ORNL
DDC TAB	Unannounced
Justification	
By	
Distribution/	
Availability/	
Dist	Availability/ or special
A	23

of computer particles associated with any grid point is determined at any instant of time by the number of these particles in the box surrounding that grid point at that time. The potential distribution is found for the particular charge distribution and boundary conditions at any given time by solving the discretized analog of the Poisson equation. The particles are "pushed" in the calculated electric field during a time step in a given cycle and, based on their positions at the end of the time-step new charge densities are assigned to the grid points. At the beginning of the next cycle a new potential distribution is calculated from these densities. The calculation thus simulates the time-behavior of the system step by step.

In the following sections we describe the particle injection algorithms, trajectory calculation method, and technique for solving the Poisson equation. Following this we discuss the results of a number of sample runs. These show that the code is capable of handling arbitrary densities and temperatures with acceptable fluctuations. Current collection and space charge distributions can be computed as functions of time. Finally the program listing of PARKTDC (Parker Time Dependent Charging) is presented.

The present program is an original development. Although there is a considerable numerical simulation literature, it applies to other geometries than the present one. Sources available to the author give little or no information on methods of simulating axially-symmetric time-dependent problems involving finite cylinders in space plasmas. Yet the axially-symmetric 3-dimensional geometry requires techniques significantly different from those of say, planar, infinitely-long cylinder, or cartesian systems. Therefore no references are cited.

## PARTICLE INJECTION ALGORITHMS

Particles are injected through the outer grid boundary according to the following velocity-distribution options: (a) unidirectional monoenergetic beam, (b) isotropic monoenergetic, and (c) isotropic Maxwellian.

### Selection from isotropic monoenergetic velocity distribution:

The outer boundary surface of the grid is a cylindrical "box" of finite length  $H$  and radius  $R_0$ . This means that in the process of statistical selection of particle injection sites, one of the three areas must be selected, namely the top (A), side (B), or bottom (C). Consider a distribution of particle beams distributed in polar angle  $\theta$  (with respect to the cylinder axis), at a fixed energy. For an isotropic angle distribution, choose  $\theta$  from  $\cos\theta$  uniformly distributed in the range  $(-1, 1)$ . Define the projections of the cylinder areas onto a plane perpendicular to the beam. There are two cases:

#### Case $\theta > \pi/2$

Define  $A = \pi R_0^2 |\cos\theta|$  and  $B = 2R_0 H \sin\theta$ , and choose random number to select A or B.

#### Case $\theta < \pi/2$

Define  $C = \pi R_0^2 \cos\theta$  and  $B = 2R_0 H \sin\theta$ , and choose random number to select B or C.

#### With angle $\theta$ and areas selected:

If the selected surface is A or C, choose azimuthal angle  $\phi$  uniformly from  $(0, \pi)$ , and choose  $r^2$  uniformly from  $(0, R_0^2)$ . This gives the position of injection on the surface. If the surface selected is B, then choose  $z$  uniformly from  $(0, H)$ , and choose  $h$  uniformly from  $(0, R_0)$ . Then obtain  $\phi$  from  $\arcsin(h/R_0)$ .

It should be noted that alternatively one could have first selected the areas A and B (or A and C) on the basis of a uniform distribution of values of the integral

$$\frac{1}{\pi} \int_0^{\pi} \frac{\sin\theta |\cos\theta| d\theta}{|\cos\theta| + (2H/\pi R_0) \sin\theta} \quad (1)$$

which represents the average over angles  $\theta$  of the ratio of A to A+B, but which does not seem to be evaluable in analytical terms. Following this selection one would then select a value for  $\theta$ . This method would be equivalent to the one used.

Components of velocity at injection point:

For a given speed  $v$ , and for selected angles  $\theta$  and  $\phi$ , we have the components of particle velocity at the point of injection:

$$\begin{aligned}\dot{x} &= v \sin\theta \cos\phi \\ \dot{y} &= v \sin\theta \sin\phi \\ \dot{z} &= v \cos\theta\end{aligned}\tag{2}$$

Selection from Maxwellian velocity distribution:

Let the velocity distribution  $f(\vec{v})$  be described by

$$f(\vec{v}) d^3\vec{v} = e^{-v_{\perp}^2} v_{\perp} dv_{\perp} \cdot e^{-v_z^2} dv_z\tag{3}$$

in terms of dimensionless velocity components, where  $v_z$  is the axial component of velocity, and  $v_{\perp}$  is the perpendicular component  $\sqrt{\dot{x}^2 + \dot{y}^2}$ . Then choose  $v_{\perp}$  from  $\text{RAN}_1 = \exp(-v_{\perp}^2)$ , and  $v_z$  from  $\text{RAN}_2 = \text{erf}(v_z)$ , where  $\text{RAN}_1$  denotes a random number uniformly distributed in the unit interval, while  $\text{RAN}_2$  denotes a random number uniformly distributed in the range  $(-1,1)$ . For  $v_z$  we need the inverse ( $=\text{erf}^{-1}$ ) of the error function  $\text{erf}$ . (The inverse function was constructed by suitably modifying a fitting formula given by Abramowitz and Stegun.) Then the angle  $\theta$  and speed  $v$  are given by  $\tan\theta = v_{\perp}/v_z$ , and  $v = \sqrt{v_{\perp}^2 + v_z^2}$ . Having the angle  $\theta$ , we then select the injection point as given earlier, and following this the velocity components  $\dot{x}$ ,  $\dot{y}$ , and  $\dot{z}$ . If there is also a drift velocity  $M$  (= Mach number) in the axial direction, then  $v_z$  can be chosen from  $\text{RAN}_2 = \text{erf}(v_z - M)$ . This method of selection thus leads to a simplification when dealing with a drifting Maxwellian distribution.

PARTICLES INJECTED IN A TIME STEP

The actual charge of particles entering a surface of area  $A$  in time  $\Delta t$  is  $e j_0 A \Delta t$ , where  $j_0$  is the random thermal flux of incident particles

and  $e$  is particle charge. Let  $j_0$  be given by  $n_0 v_{\perp}$ , where  $n_0$  is the plasma density, and  $v_{\perp}$  is the mean velocity component of incident real particles normal to the surface. For a choice of time interval  $\Delta t = \Delta z / v_{\perp}$ , where  $\Delta z$  is a chosen thickness of transit (say, of the order of a mesh interval), we may determine the number  $N$  of real particles injected during the time step:

$$N = j_0 A \Delta t = n_0 A \Delta z \quad (4)$$

If  $N'$  is the chosen number of computer particles injected in a time step, say 10 to 1000, the charge  $e'$  per computer particle is given by

$$e' = \frac{Ne}{N'} = \frac{en_0 A \Delta z}{N'} \quad (5)$$

In the problem of isotropic injection,  $A$  may be evaluated as  $2\pi R_0^2 (1 + H/R_0)$ . The value of  $e'$  is used to calculate the charge assigned to a grid point through its product with the number of computer particles found at a given instant of time to be associated with that grid point. It is similarly used to compute the charge incident on and absorbed by the body.

Typical values of the parameters which have been used in tests are the following:

$$\begin{aligned} R_0 &= 100 \text{ cm} & N' &= 10 \text{ to } 1000 \\ H &= 100 \text{ cm} \\ \Delta z &= 2 \text{ cm} \\ n_0 &= 10^3/\text{cm}^3 \text{ to } 10^5/\text{cm}^3 \end{aligned}$$

For  $n_0 = 10^3$ ,  $N = 2.51 \times 10^8$  is the number of real particles injected per time step. Each computer particle then represents  $N/N'$  real particles, or  $2.51 \times 10^5$  to  $2.51 \times 10^7$ .

## TRAJECTORY CALCULATION

The particles are moved in accord with the equations of motion, which may be represented as follows.

Let  $X$  denote the dimensional vector position ( $X$  stands for any one of the 3 cartesian coordinates). Let  $T$  denote the dimensional time, and let  $V$  denote the dimensional potential (in volts) at the position  $X$ . Then over a short step  $\Delta T$  with constant acceleration,  $X$  changes in accord with

$$X = X_0 + \dot{X}_0 \Delta T - \frac{q}{2m} \frac{dV}{dX} (\Delta T)^2 \quad (6)$$

where  $q$  and  $m$  are the particle charge and mass, and where  $X_0$  is the initial value of  $X$  at the beginning of the time step. Let  $dV/dX$  be in volts/cm, and let  $\Delta T$  be expressed as

$$\Delta T = \frac{(\Delta Z)_{\min} \Delta t}{v_{oi}} \quad (7)$$

where  $(\Delta Z)_{\min}$  is the minimum zone thickness,  $v_{oi}$  is the scale velocity of the ions, and  $\Delta t$  is a dimensionless time, called "DELTA" in the program. Let  $m$  be the mass of the particle expressed in units of the ion mass. Then we may write

$$X = X_0 + \dot{X}_0 \frac{(\Delta Z)_{\min} \Delta t}{v_{oi}} - \frac{(\Delta Z)_{\min}^2 (\Delta t)^2}{4(m/m_i)} \frac{d}{dX} \left( \frac{qV}{E_i} \right) \quad (8)$$

where  $E_i$  (ev) is the scale energy of the ions ( $= m_i v_{oi}^2 / 2$ ) called "TVIONS" in the program, and  $qV$  is the potential energy in ev. Similarly we obtain the velocity components:

$$\dot{X} = \dot{X}_0 - v_{oi} \frac{(\Delta Z)_{\min} (\Delta t)}{2(m/m_i)} \frac{d}{dX} \left( \frac{qV}{E_i} \right) \quad (9)$$

Hence, assuming electrons and one species of ion, the program

(a) reads in:

$\Delta t = \text{"DELTA"}$

$E_i$  (ev) = "TVIONS" = ion energy

$E_e$  (ev) = "TVELEC" = electron energy

$m_i/m_e = \text{"XMASS"}$  = mass ratio

and

(b) constructs the scale velocity

$$v_0 = \text{"SPEED"} = 5.93 \times 10^7 \sqrt{\text{TVIONS/XMASS}}$$

= ion speed in cm/sec

and

(c) finds the minimum zone thickness  $(\Delta Z)_{\min}$ .

The particles are moved in subroutine "TRACK" (called by "DENSTY"), with given

$$X, \dot{X}/\text{SPEED}, \text{ and } "DT" = (\Delta Z)_{\min} \times \text{DELTA}$$

With the potential grid replaced by  $\phi = (m_i/m)qV/E_i$ , the new values of  $X$  and  $\dot{X}$  are given by:

$$X = X_0 + \dot{X}_0 \frac{DT}{v_0} - \frac{(DT)^2}{4} \frac{d\phi}{dX} \quad (10)$$

$$\dot{X} = \dot{X}_0 - v_0 \frac{DT}{2} \frac{d\phi}{dX} \quad (11)$$

### Interpolation

The required components of  $d\phi/dX$  in the latter equations are obtained by double linear interpolation within the boxes of the grid. Let  $r$  and  $z$  be located in the ranges  $r_j \leq r < r_{j+1}$  and  $z_i \leq z < z_{i-1}$ . Then the interpolated values of  $\partial\phi/\partial r$  and  $\partial\phi/\partial z$  are given by:

$$\frac{\partial\phi}{\partial r} = [\phi(i, j+1) - \phi(i, j) + (z-z_i)G/D_z]/D_r \quad (12)$$

$$\frac{\partial\phi}{\partial z} = [\phi(i-1, j) - \phi(i, j) + (r-r_j)G/D_r]D_z \quad (13)$$

where

$$G = \phi(i-1, j+1) + \phi(i, j) - \phi(i-1, j) - \phi(i, j+1) \quad (14)$$

and

$$D_r = r_{j+1} - r_j, \quad D_z = z_{i-1} - z_i \quad (15)$$

The interpolated potential itself is given by

$$\begin{aligned} \phi = & \phi(i, j) + (r-r_j)[\phi(i, j+1) - \phi(i, j)]/D_r \\ & + (z-z_i)[\phi(i-1, j) - \phi(i, j)]/D_z \\ & + (r-r_j)(z-z_i)G/D_r D_z \end{aligned} \quad (16)$$

### THE POISSON PROBLEM: POISSON DIFFERENCE EQUATIONS

In the present problem the electrostatic field is axially symmetric and is defined on a mesh of spatial grid points, such that at any point (including grid points) the potential and electric field can be obtained by interpolation.

Assume that the space charge density is known at the grid points. Consider a group of interior grid points, forming a portion of the overall grid as shown in Fig. 3. In this figure, the vertical and horizontal directions are the  $z$  and  $r$  directions, respectively, where  $z$  and  $r$  denote the cylindrical axial and cylindrical radial coordinates, respectively. Three horizontal grid lines, of constant  $z$ -values  $z_{i-1}$ ,  $z_i$ , and  $z_{i+1}$ , and three vertical grid lines, of constant  $r$  values  $r_{j-1}$ ,  $r_j$ , and  $r_{j+1}$ , are shown in the figure. (Note that the index ( $i$ ) of  $z$  increases as  $z$  decreases.) The set of grid lines intersect at 9 grid points, or nodes, as shown. Each point may be considered to be associated with a volume of space, and to have a group of four neighboring points which "interact" with it. Thus, consider the central point of the group, labeled  $C$  in the figure, which may be identified with one of the grid points in Fig. 2. Associated with this point is a volume of revolution (a torus) whose cross-section is rectangular and is shown by the rectangular shaded area surrounding Point  $C$ . The shaded area is defined by connecting the mid-points of the surrounding mesh rectangles. Let  $\tau$  denote the volume of the torus, and let the neighboring points (above, below, to the right of, and to the left of  $C$ ) be labeled  $N$ ,  $S$ ,  $E$  and  $W$  (north, south, east and west, respectively).

Let the Poisson equation be written

$$\nabla^2 \phi = - \rho / \epsilon \quad (17)$$

where  $\rho$  denotes the space charge density, and  $\epsilon$  denotes the dielectric constant.

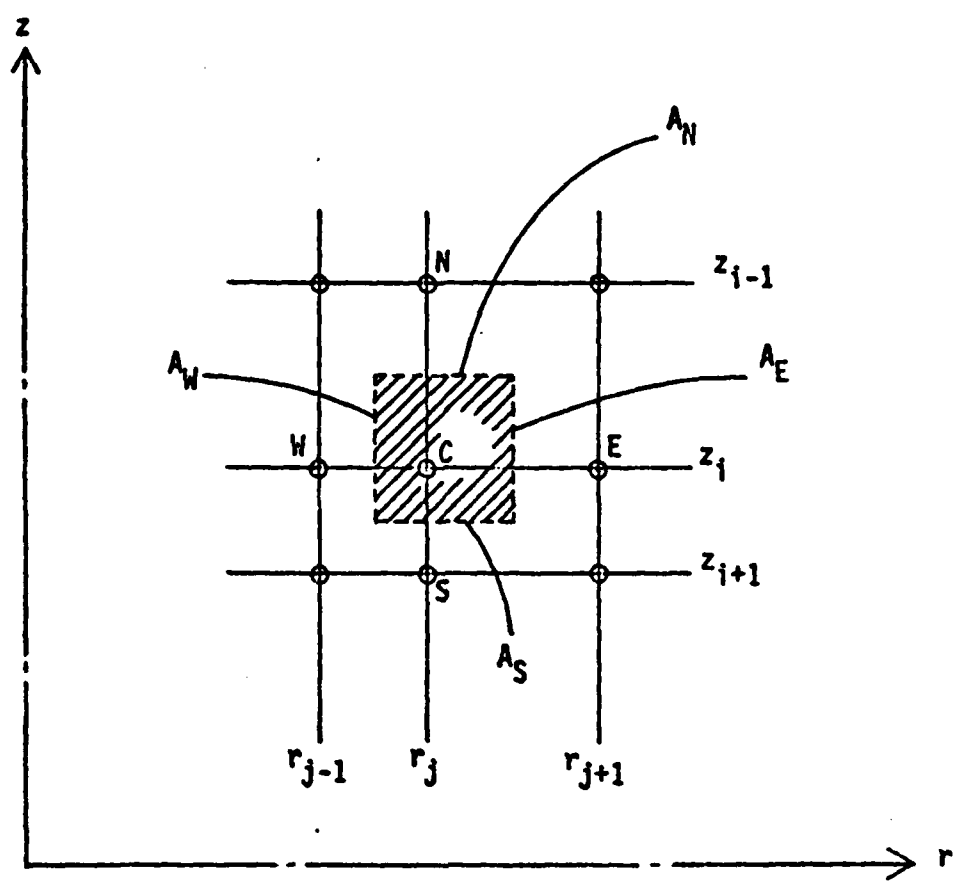


FIG. 3. GROUP OF INTERIOR GRID POINTS IN  $r, z$  GRID

The grid lines may be considered to be arbitrarily chosen so that the mesh intervals are nonuniform. In this case the Poisson difference equations may be obtained by integrating Eq. (17) over the volume  $\tau$  of the torus associated with Point C:

$$\iiint_{\tau} \nabla^2 \phi \, d\tau = - \iiint_{\tau} \rho \, d\tau - Q_C / \epsilon \quad (18)$$

where  $Q_C$  is known at the grid point C. The right-hand side has been approximated as shown since  $\tau$  is small in principle, and  $Q_C$  is the net charge in the box at Point C. By the divergence theorem, the left-hand side becomes

$$\begin{aligned} \iint_{\Sigma} \frac{\partial \phi}{\partial n} \, d\Sigma \approx \\ A_N \left( \frac{\partial \phi}{\partial n} \right)_N + A_S \left( \frac{\partial \phi}{\partial n} \right)_S + A_E \left( \frac{\partial \phi}{\partial n} \right)_E + A_W \left( \frac{\partial \phi}{\partial n} \right)_W \end{aligned} \quad (19)$$

where  $\Sigma$  denotes the surface of the torus;  $\partial\phi/\partial n$  is the component of  $\nabla\phi$  in the outward normal direction at the surface;  $A_N$ ,  $A_S$ ,  $A_E$ , and  $A_W$  denote the areas of the north, south, east, and west surfaces, respectively; and the quantities  $(\partial\phi/\partial n)_{N,S,E,W}$  denote values of  $\partial\phi/\partial n$  taken to be constant on the corresponding surfaces.

$(\partial\phi/\partial n)_{N,S,E,W}$  may be approximated by difference quotients, namely,

$$\left( \frac{\partial \phi}{\partial n} \right)_N \approx \frac{(\phi_N - \phi)}{(z_{i-1} - z_i)} \quad \left( \frac{\partial \phi}{\partial n} \right)_S \approx \frac{(\phi_S - \phi)}{(z_i - z_{i+1})} \quad (20)$$

$$\left( \frac{\partial \phi}{\partial n} \right)_E \approx \frac{(\phi_E - \phi)}{(r_{j+1} - r_j)} \quad \left( \frac{\partial \phi}{\partial n} \right)_W \approx \frac{(\phi_W - \phi)}{(r_j - r_{j-1})}$$

where  $\phi$  denotes the potential at Point C and  $\phi_N$ ,  $\phi_S$ ,  $\phi_E$ ,  $\phi_W$  denote the neighboring potentials. If Point C is an interior grid point, the areas  $A_N$ ,  $A_S$ ,  $A_E$ , and  $A_W$  are given by

$$\begin{aligned}
 A_N &= \frac{\pi}{4} [(r_{j+1} + r_j)^2 - (r_j + r_{j-1})^2] \\
 A_S &= A_N \\
 A_E &= \frac{\pi}{2} (r_{j+1} + r_j)(z_{i-1} - z_{i+1}) \\
 A_W &= \frac{\pi}{2} (r_j + r_{j-1})(z_{i-1} - z_{i+1})
 \end{aligned} \tag{21}$$

and the volume  $\tau$  is given by

$$\tau = \frac{A_N}{2} (z_{i-1} - z_{i+1}) \tag{22}$$

Thus we obtain the difference equation in the form

$$C_N \phi_N + C_S \phi_S + C_E \phi_E + C_W \phi_W - C \phi = - Q_C / \epsilon \tag{23}$$

where

$$C = C_N + C_S + C_E + C_W \tag{24}$$

and

$$\begin{aligned}
 C_N &= \frac{A_N}{(z_{i-1} - z_i)} & C_S &= \frac{A_S}{(z_i - z_{i+1})} \\
 C_E &= \frac{A_E}{(r_{j+1} - r_j)} & C_W &= \frac{A_W}{(r_j - r_{j-1})}
 \end{aligned} \tag{25}$$

This shows how to form the difference equations used for the Poisson problems of this report. Equation (24) holds only for an "interior" point of the grid, that is, a point surrounded by neighbors on all four sides. If Point C has a known neighboring potential (for example, if Point C is adjacent to the spacecraft surface), then the corresponding term on the left-hand side of Eq. (23) is transferred to the right-hand side as a known quantity.

The boundary conditions for the potentials in the Poisson problem are as follows. At points representing the body surface, the normalized potentials are fixed at the chosen values. At the external (boundary) points of the grid, where "infinity" is represented on the computer, a "floating" condition is optionally used, namely, a linear relation between  $\phi$  and  $\partial\phi/\partial n$ , the normal component of  $\nabla\phi$ . The exact relation of  $\phi$  to  $\partial\phi/\partial n$  is not important when the external boundary of the grid is sufficiently far away. (For the calculations to be reported, the assumed relation was the same as for a Coulomb potential.) In any case, either the fixed condition  $\phi = 0$  or the floating condition will give the same results, provided the grid boundary is moved sufficiently far out. The effects of various types of boundary conditions representing "infinity" have been studied by the author.

In general, the floating condition appears to be computationally more efficient than the fixed one. Of course, the floating condition becomes ideal when the true relation between  $\phi$  and  $\partial\phi/\partial n$  is used, but this requires that the asymptotic form of the solution be known in advance.

The boundary conditions at the outer grid surfaces can be combinations of fixed and floating conditions.

Consider a Point C on the outer boundary of the grid where a floating boundary condition is chosen. If the potential is assumed to satisfy the linear law

$$\frac{\partial\phi}{\partial n} = \frac{\partial\phi}{\partial z} = -\alpha\phi \quad (26)$$

on the z-boundary (North or South), and

$$\frac{\partial\phi}{\partial n} = \frac{\partial\phi}{\partial r} = -\beta\phi \quad (27)$$

on the r-boundary (East only;  $\beta=0$  on the West), then the corresponding "neighbor term" on the left-hand side of Eq. (23) vanishes, and the corresponding "neighbor coefficient" on the right-hand side of Eq. (24) is replaced by  $\alpha A$  or  $\beta A$ , where  $A$  is the appropriate area. The quantities  $\alpha$  and  $\beta$  depend on the position and on the assumed model for the variation of the potential at large distances.

Once the coefficients of all of the equations (corresponding to the grid points where the potentials are unknown) are computed, the system of linear equations of the form of Eq. (23) may be solved by iteration. Point-successive over-relaxation is a well-known process and has been found to be effective in the present problem. For the relaxation process, one rearranges the equations, so that the "diagonal" term is alone on the left-hand side, while all the other terms are on the right-hand side with the known charge-density term. Thus, Eq. (23) becomes

$$C\phi = C_N\phi_N + C_S\phi_S + C_E\phi_E + C_W\phi_W + Q_C/\epsilon \quad (28)$$

First, an initial guess is made for the values of all the potentials. Then new values are obtained from the left-hand sides of all of the equations (28), using previous values on the right-hand sides. One "sweeps" through the equations successively, replacing the potentials on the right-hand sides with updated values as they become available from preceding equations. This procedure is usually stable and leads to convergence. "Over-relaxation" is the process of mixing successive potential iterates in such a way as to enhance the rate of convergence.

It is convenient to express all potentials in volts and all lengths in centimeters. Then if the charges are all expressed in picocoulombs, we need simply to replace  $1/\epsilon$  by  $0.9 \times 4\pi$  or  $3.6\pi$ , in vacuum.

## SAMPLE RESULTS

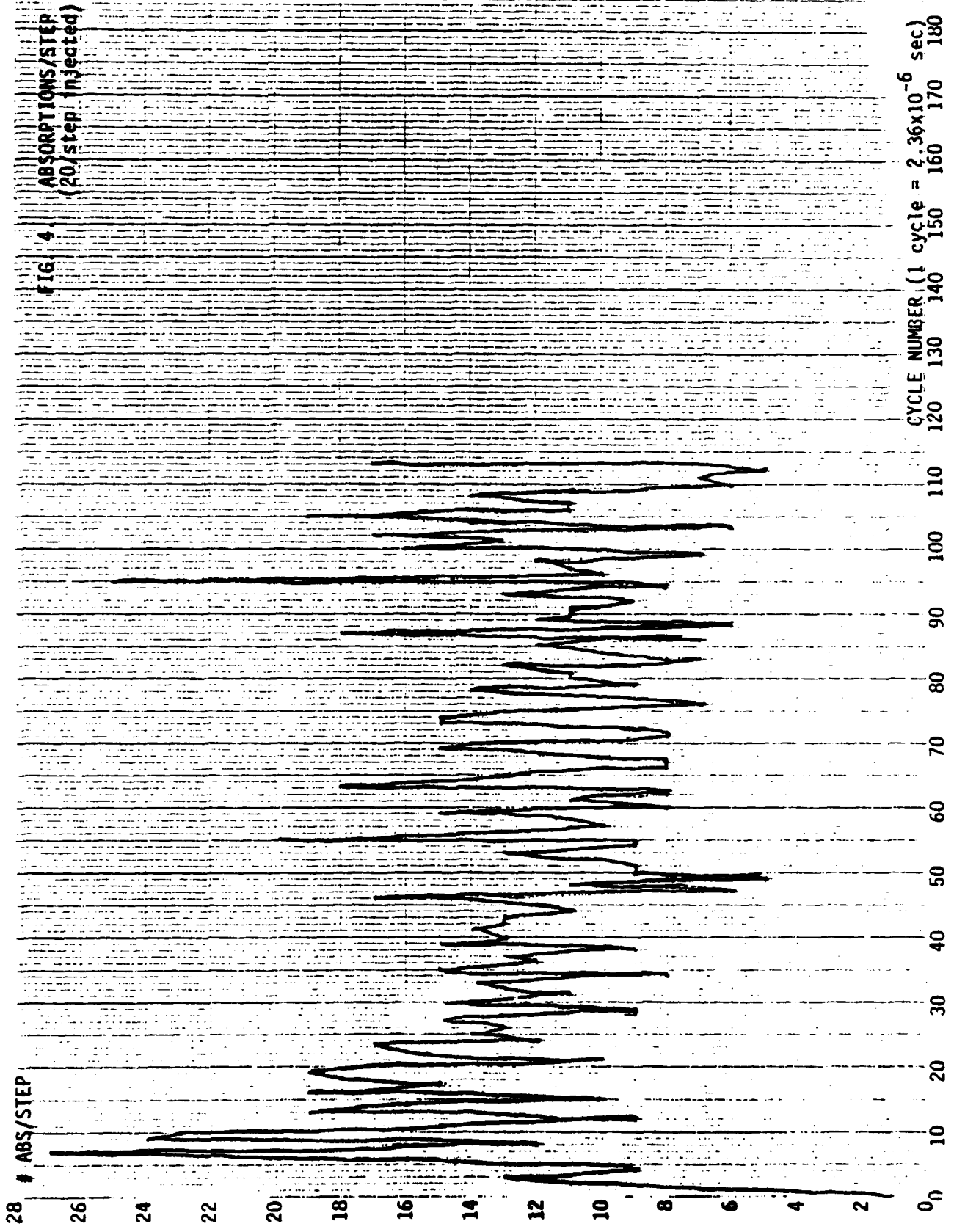
One of the most vexing problems of numerical time simulation is that of noise, wherein fluctuations not representative of the actual plasma are generated by the relatively small numbers of particles and grid points used. This problem becomes more severe as the plasma density increases, that is, as the Debye length becomes small compared with the spacecraft size. Since it was deemed essential that the code be able to cope with dense as well as tenuous plasmas, as part of the code checkout sample runs were made with the parameters, temperature  $T=0.1$  ev, and density  $n_0$  varying from  $10^3/\text{cm}^3$  to as high a limit as practicable.

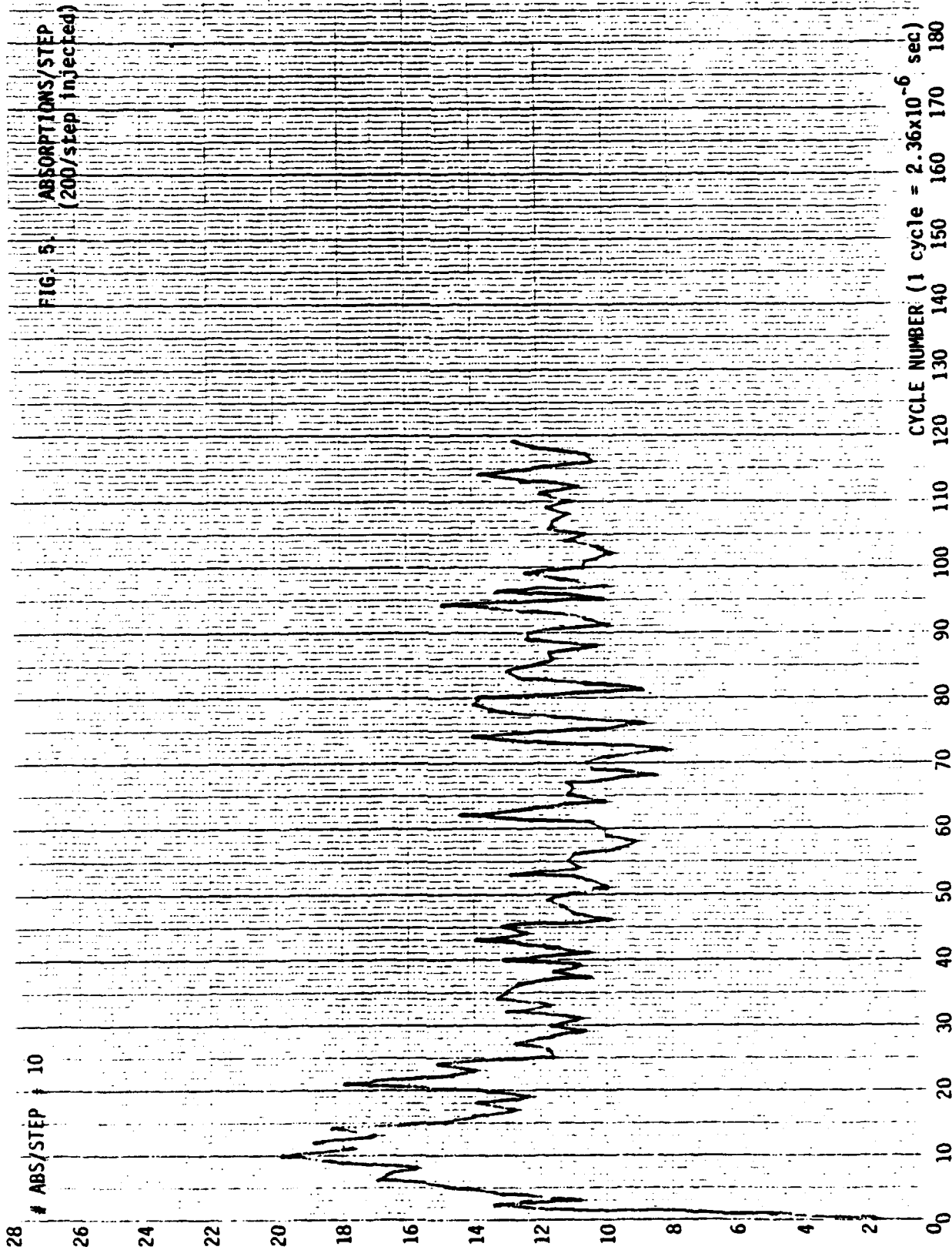
Among the quantities of interest are the collected fluxes of ions and electrons. Fluctuations in these are a measure of those in the plasma. In the first problem to be discussed the spacecraft is modeled by a disk, of diameter one meter. The plasma has density  $1000/\text{cm}^3$ , and is Maxwellian with  $T=0.1$  ev. In this case the Debye length is 7.43 cm, so that the Debye number is 0.149. The electron and ion plasma periods are  $3.5 \times 10^{-6}$  sec and  $1.8 \times 10^{-5}$  sec, respectively, for an assumed ion mass of 25 electron masses. (The use of an unphysically small ion mass is common in simulations, especially if the steady state is wanted. In the steady state the solution normally does not depend on the ion mass.)

The disk is assumed to "turn on" at  $t=0$  with a fixed potential of  $-10$  v. Time steps of length  $2.36 \times 10^{-6}$  sec are chosen, with 100 electrons and 20 ions injected per step. (The ratio 100 to 20 is the same as the ratio of electron-to-ion random thermal fluxes.) A coarse grid of 25 points is used.

The number of ions absorbed per step represents the collected ion current. This quantity is shown in Fig. 4 as a function of cycle (step) number. It is strongly fluctuating, but quite clearly centered about 12/step. Increasing the number injected per step to 1000 electrons and 200 ions results in a smaller fluctuation as illustrated in Fig. 5. When 10-cycle averages are plotted instead of the raw data, one obtains a "quieter" current as shown in Fig. 6.

In further runs to be discussed, the plasma density varies from  $10^4/\text{cm}^3$  to  $10^5/\text{cm}^3$ . The corresponding values of Debye length, Debye number, electron





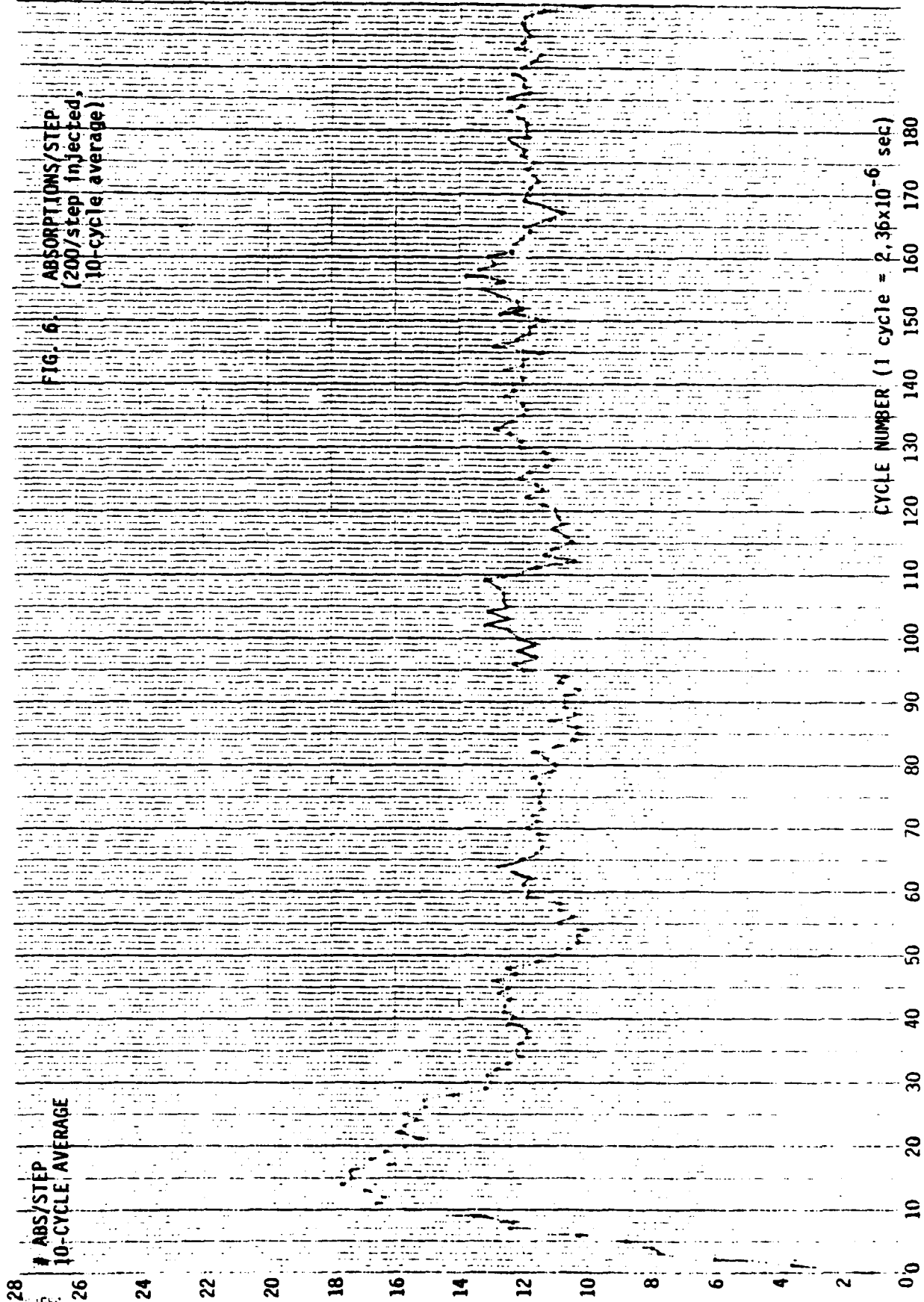


FIG. 6  
ABSORPTIONS/STEP  
(200/step injected,  
10-cycle average)

CYCLE NUMBER (1 cycle = 2.36x10<sup>-6</sup> sec)

plasma period  $\tau_e$ , and the ion plasma period  $\tau_i$ , are given in the following table.

Run	$n_0$ ( $\text{cm}^{-3}$ )	$\lambda_D$ (cm)	$\lambda_D/50$ --	$\tau_e$ (s)	$\tau_i$ (s)
A	$10^4$	2.35	.047	$1.11 \times 10^{-6}$	$5.55 \times 10^{-6}$
B	$2.5 \times 10^4$	1.49	.030	$7.02 \times 10^{-7}$	$3.51 \times 10^{-6}$
C	$5.0 \times 10^4$	1.05	.021	$4.96 \times 10^{-7}$	$2.48 \times 10^{-6}$
D	$7.5 \times 10^4$	0.86	.017	$4.05 \times 10^{-7}$	$2.03 \times 10^{-6}$
E	$10^5$	0.74	.015	$3.51 \times 10^{-7}$	$1.76 \times 10^{-6}$

In the following we will discuss Runs A and D in detail. Runs B and C gave results intermediate. Run E was so strongly fluctuating that it was not pursued further. The density of  $7.5 \times 10^4$  seemed to be the largest (at  $T=0.1$  ev) that could be treated at the present boundary condition (20 cm from the spacecraft surface). Higher densities will require that the boundary be moved inward.

Figure 7 (Figs. 7a-7g) shows results of Run A ( $n_0=10^4$ ). This is a printer plot of the variations with cycle number (one cycle =  $9.5 \times 10^{-8}$  sec) of

- A = ion absorptions per step (scaled from 0 to 30)
- B = ion population (scaled from 2500 to 5000)
- C = electron population (scaled from 2500 to 5000)
- D = maximum potential (scaled from 0 volts to 25 volts)

The significances of A, B, C, D, are as follows. We inject 100 electrons and 20 ions per step.

A represents, through the ratio of the number absorbed to the number injected, multiplied by the ratio of the area of collection to the area of injection, the collected current relative to the ambient thermal current available.

B and C represent the ion and electron populations (numbers of particles active throughout the grid). These should become constant as the steady state is approached. In order to arrive at the steady state as quickly as possible, a "quiet start" approach is used, wherein the space is previously

PRINTED PLOT SUMMARY OF DATA AS A FUNCTION OF TIME. DEGCC = 1.0000E+04      FIG. 7a.      RUN A  
 COARSE HERREST GYRO POINT MODEL      (Density 10<sup>4</sup>/cm<sup>3</sup>)

SYMBOL CORRESPONDENCE  
 A = ION ABSORPTIONS      SCALED FROM 0.0 TO 30.0  
 R = ION POPULATION      SCALED FROM 2500.0 TO 5000.0  
 C = ELFCYROM POPULATION      SCALED FROM 2500.0 TO 5000.0  
 D = MAXIMUM POTENTIAL      SCALED FROM 0.0 TO 25.0

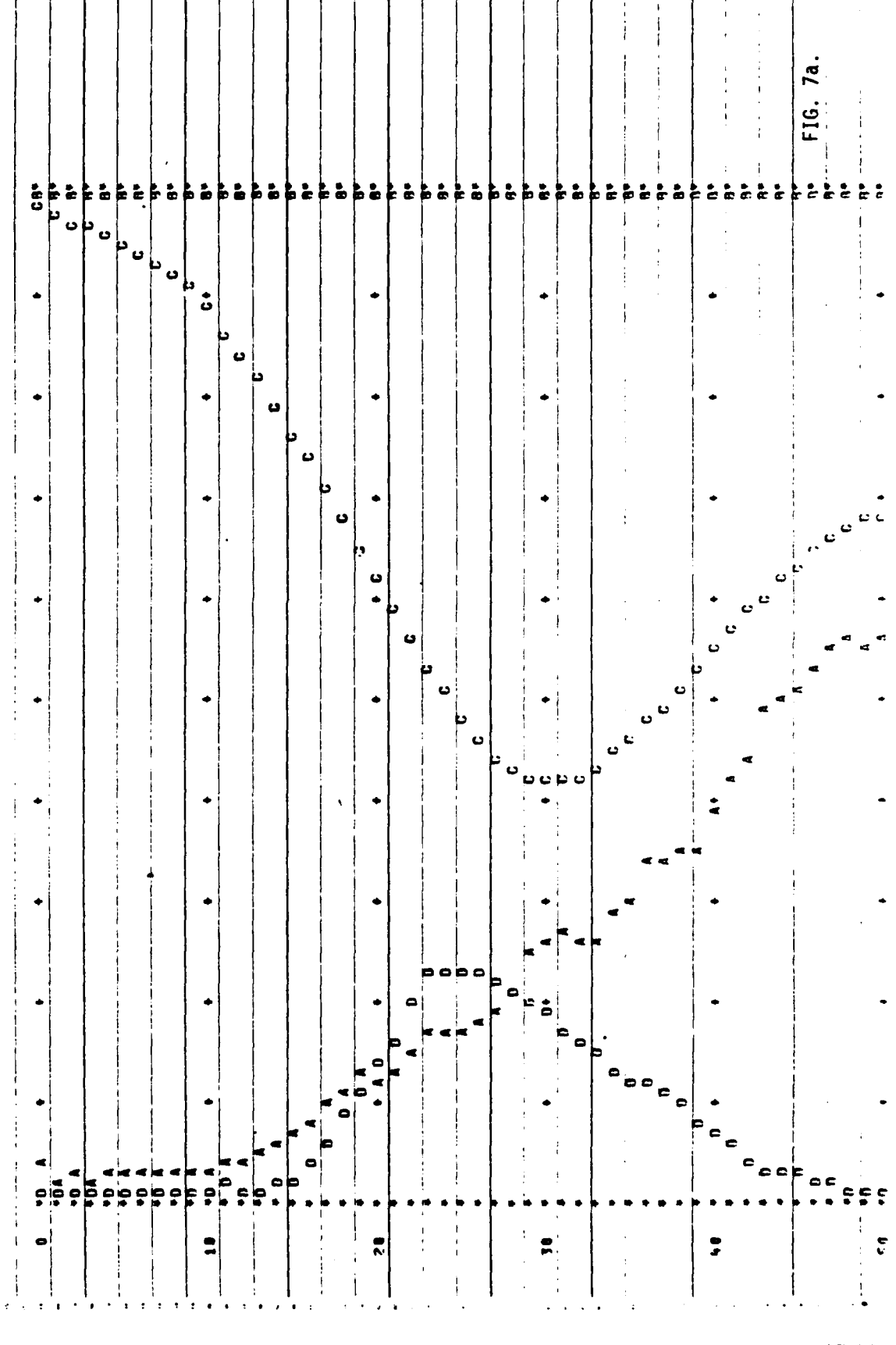


FIG. 7a.

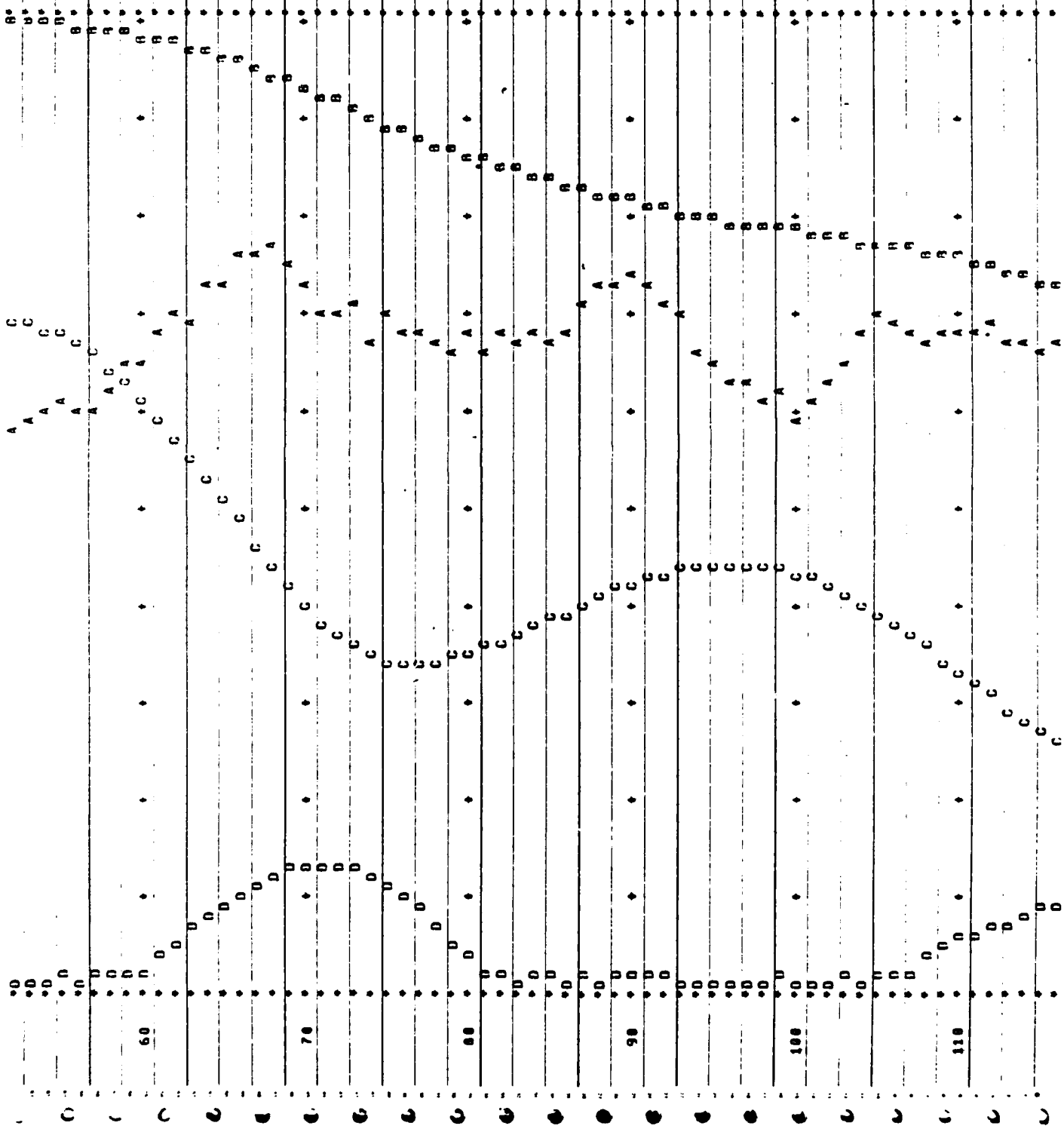


FIG. 7b.

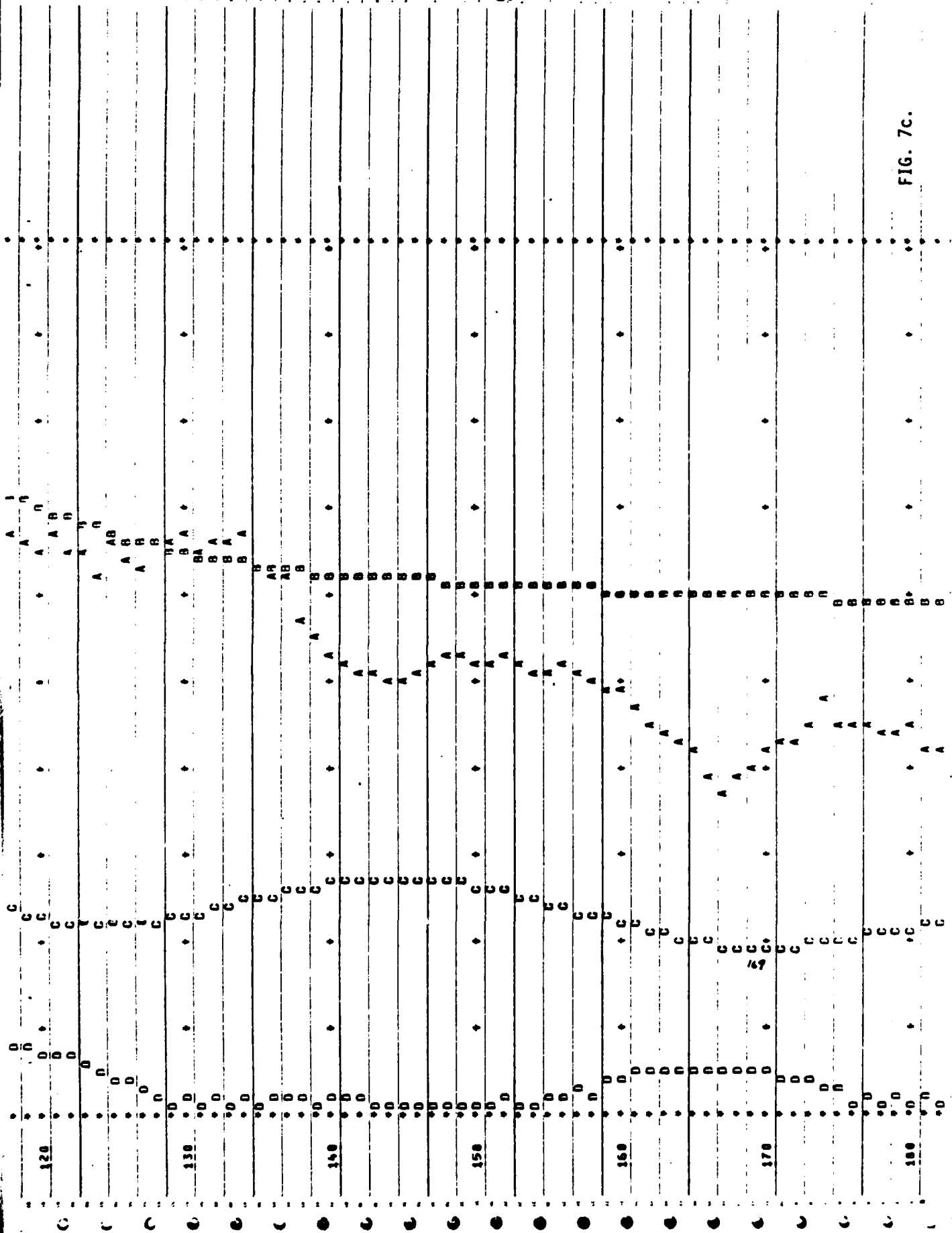


FIG. 7c.

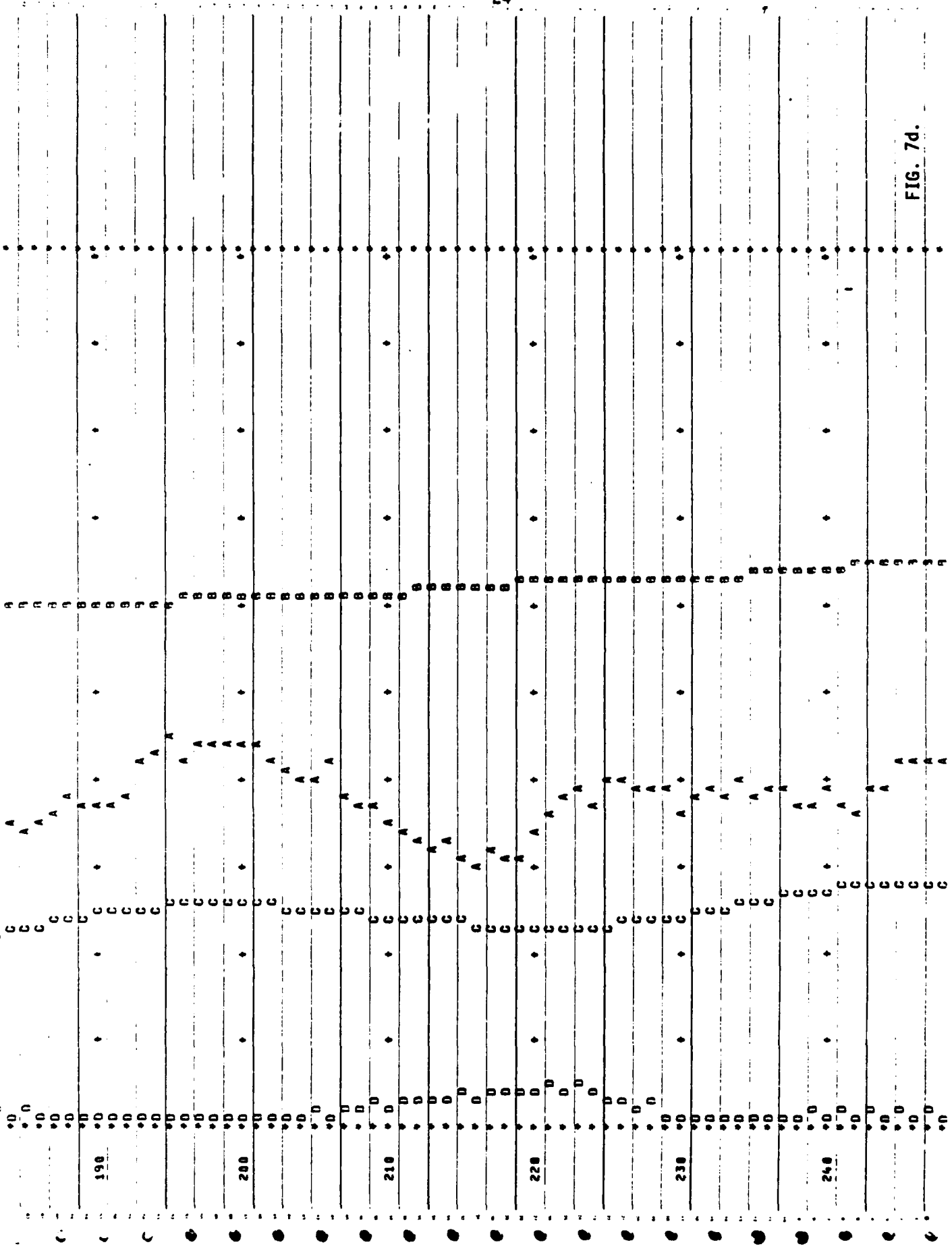


FIG. 7d.

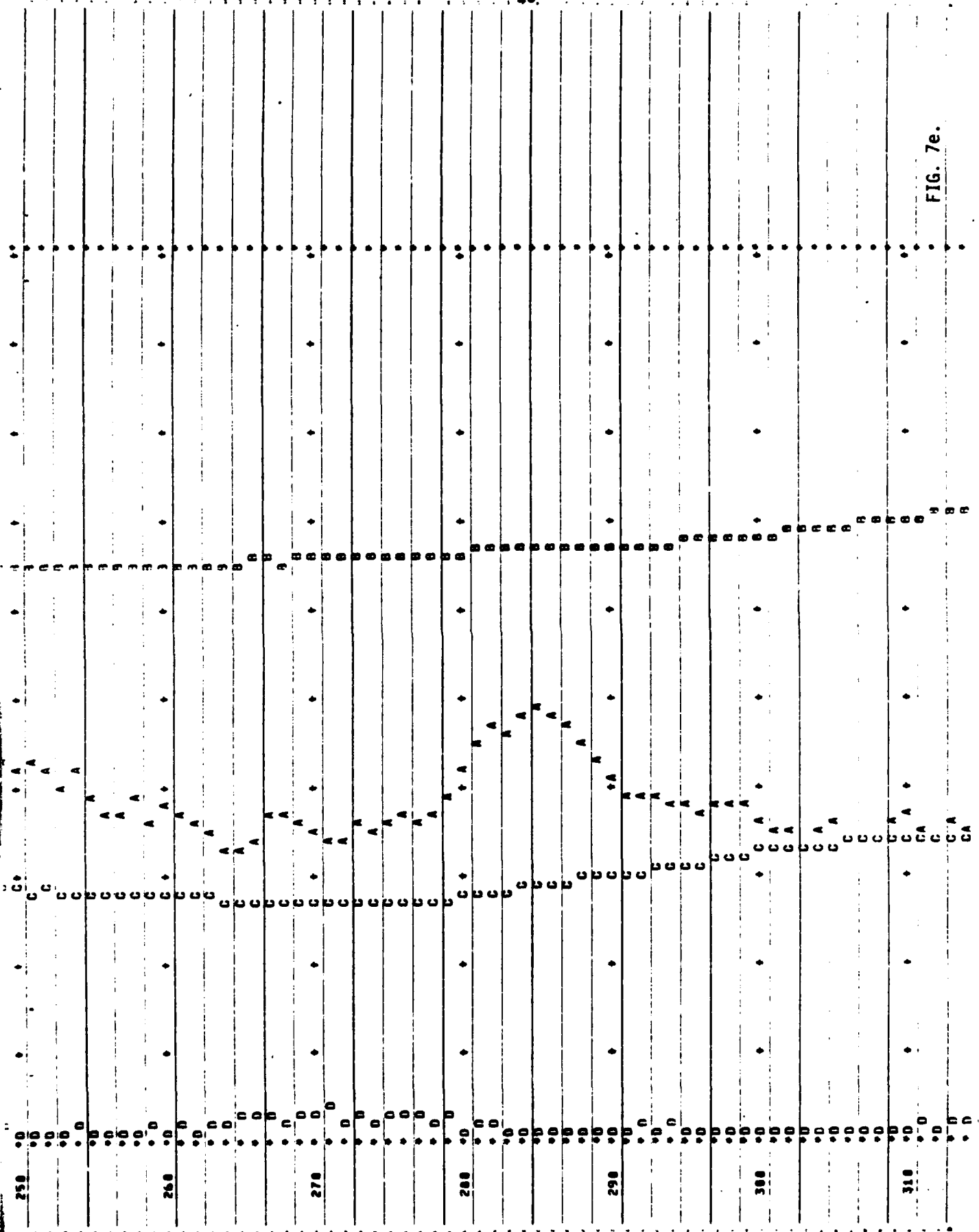


FIG. 7e.

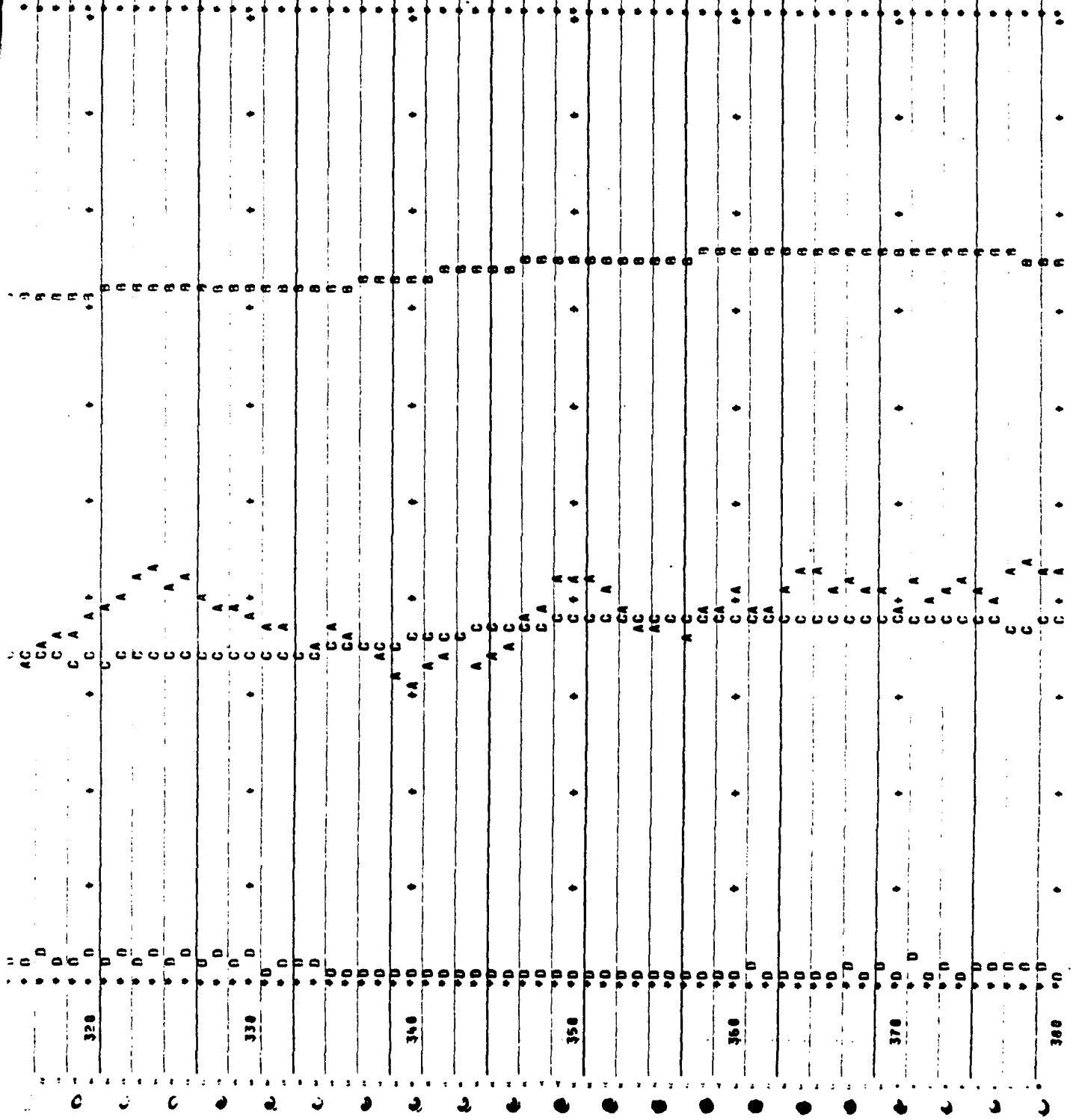


FIG. 7f.



filled with particles so as to represent normal density. In the present case 5000 ions and 5000 electrons are used, to represent the unperturbed populations.

D is a measure of the fluctuation intensity. Since the boundary potentials are allowed to float, they are susceptible to oscillations. In particular, when the instability is severe the boundary potentials tend to become positive, with the maximum increasing as the severity increases. Hence the maximum is monitored.

The behavior of A, B, C, and D as functions of time are shown by Fig. 7 as follows. Large transients occur in all 4 quantities during the first 150 cycles ( $t=1.4 \times 10^{-5}$  sec). Both the electron (C) and ion (B) populations decrease from their unperturbed values, the electrons because they are repelled away, and the ions because they are accelerated and spend less time in any given volume (a consequence of Langmuir probe theory). The electrons (C) oscillate while approaching their final value, the oscillations having diminishing amplitude. The period of the oscillations is roughly 50 cycles, or  $4.7 \times 10^{-6}$  sec, about 4 times the electron plasma period (see table), approximately equal to the ion plasma period. It is not clear whether the oscillations represent physical behavior; their smoothness suggests that they do. However, the effects of numerical "aliasing", wherein high-frequency sources can drive low-frequency oscillations through the finite grid spacing, must certainly be present because of the relatively large grid intervals (of order 10 cm).

After about 220 cycles ( $2.1 \times 10^{-5}$  sec) the electron population has settled down and subsequently rises slowly toward its asymptotic value, about 3500 out of the original 5000. The ions (B), on the other hand, do not oscillate, but drop monotonically toward a shallow minimum at about 180 cycles ( $1.7 \times 10^{-5}$  sec), and then rise slowly toward their asymptotic population, about 4400 out of the original 5000.

The ion absorption rate (A) rises initially, overshooting and subsequently decaying to an asymptotic value of about 11 absorbed per step. A higher-frequency oscillation of relatively small amplitude and a period of about 20 cycles ( $1.9 \times 10^{-6}$  sec) is superimposed. The ratio of the current collected to

the unperturbed current that would be collected in the absence of electric fields is given by dividing the number absorbed per step by the number injected per step, multiplied by the ratio of injection area to collection area, and divided by DELTA, a time-step input defined earlier.

Figure 8 (Figs. 8a-8g) shows results of Run D ( $n_0 = 7.5 \times 10^4$ ). All inputs are the same as in Fig. 7. However, now we see that there are severe fluctuations in curve D, the boundary potential. The ion and electron populations (B and C) both descend almost monotonically to lower values than they had in Fig. 7. They are closer to each other (about 3300 and 3000) than they were. The ion absorption rate (A) initially overshoots to a large value, but more quickly settles to a "steady" state, with some fluctuations though not severe, about 6 absorbed per step, roughly half of that in Fig. 7.

During the transient period in absorptions (first 70 cycles or about  $6.6 \times 10^{-6}$  sec), the electron population (C) oscillates with relatively small amplitude and smaller period (about 25 cycles or  $2.4 \times 10^{-6}$  sec) than in Fig. 7. This period is again about the same as the ion plasma period (table). Hence this oscillation, which appears regular, may be physical. It is interesting that the electron population fluctuations (the ions have little or none) are much less severe than and apparently uncorrelated with the boundary potential fluctuations (D).

It should be stated that a "cloud-in-cell" model was also developed and used, without significantly reducing the noise as had been hoped. It also gave slightly different (higher) values for the steady state absorption rate. The runs made with the cloud model will not be discussed.

Figure 9 shows a potential contour plot for Run D, generated by the computer printer, at the "steady" state.

PRINTED PLOT SUMMARY OF DATA AS A FUNCTION OF TIME. DENCC = 7.5000E+04  
 CNAPE NCP MODEL

FIG. 8a. RUN D

(Density  $7.5 \times 10^4 / \text{cm}^3$ )

SYMBOL CORRESPONDENCE

A = ION ABSORPTIONS  
 R = ION POPULATION  
 C = ELECTRON POPULATION  
 D = MAXIMUM POTENTIAL

0.0 TO 30.0  
 2500.0 TO 5000.0  
 2500.0 TO 5000.0  
 0.0 TO 25.0

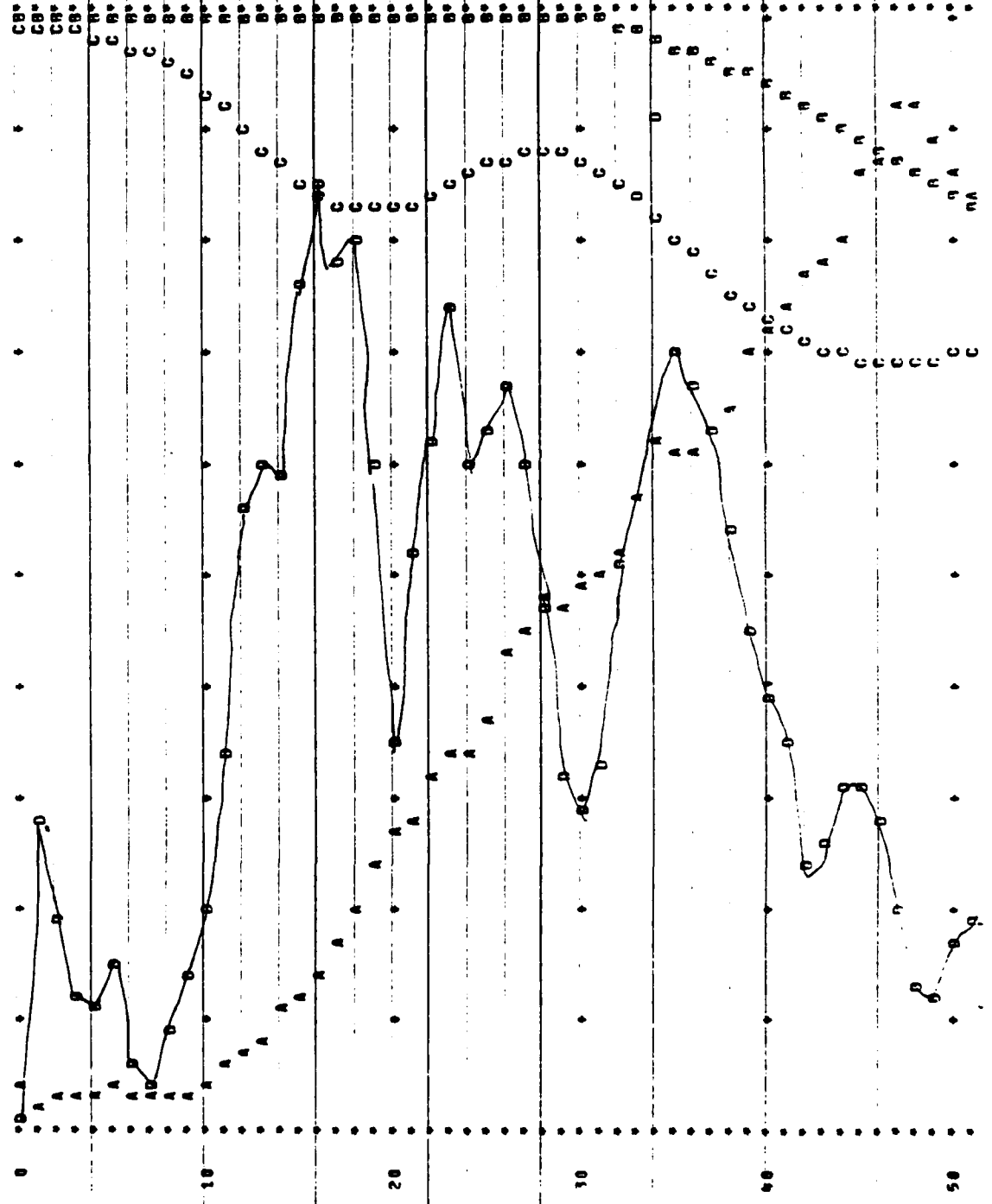


FIG. 8a.

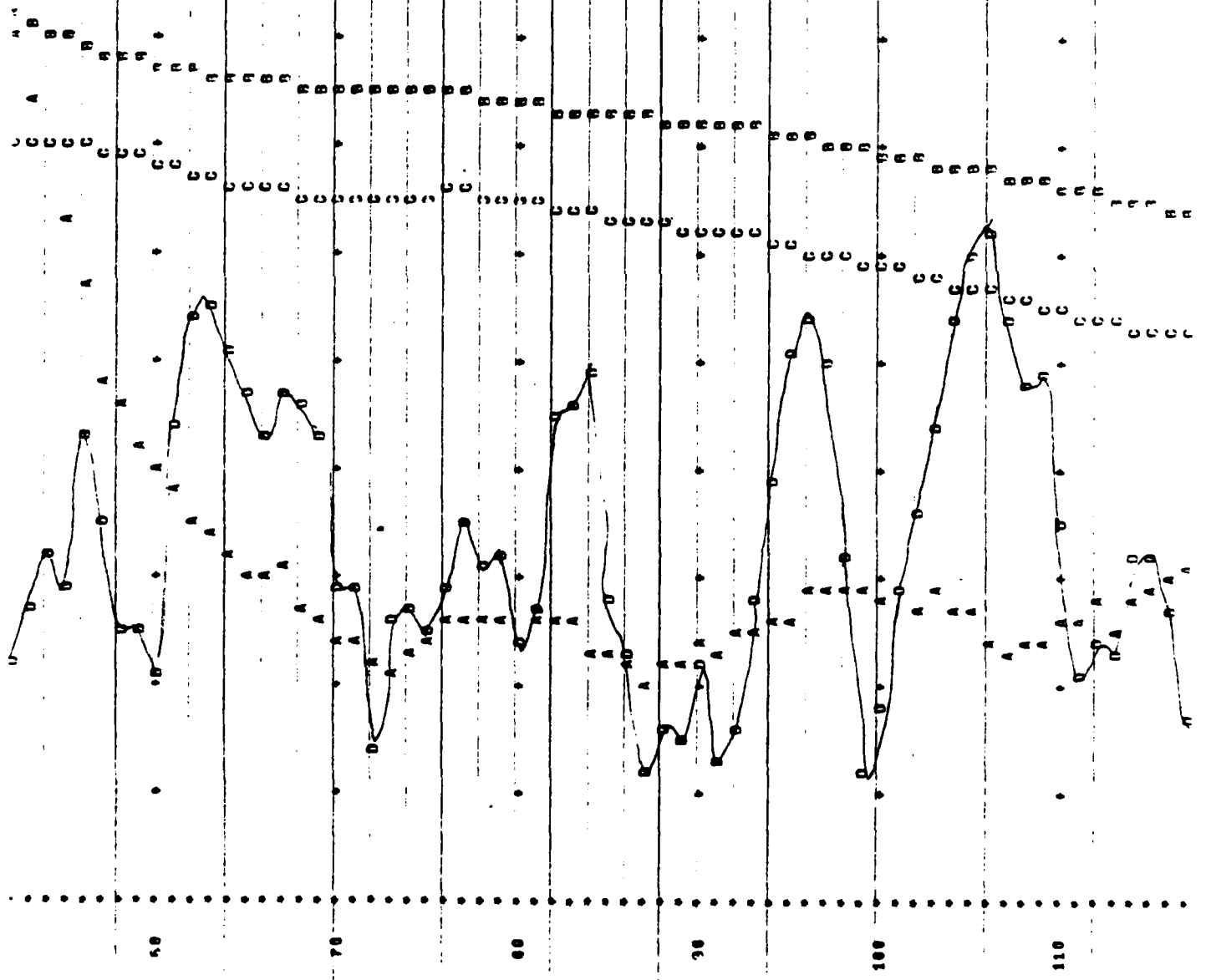


FIG. 8b.

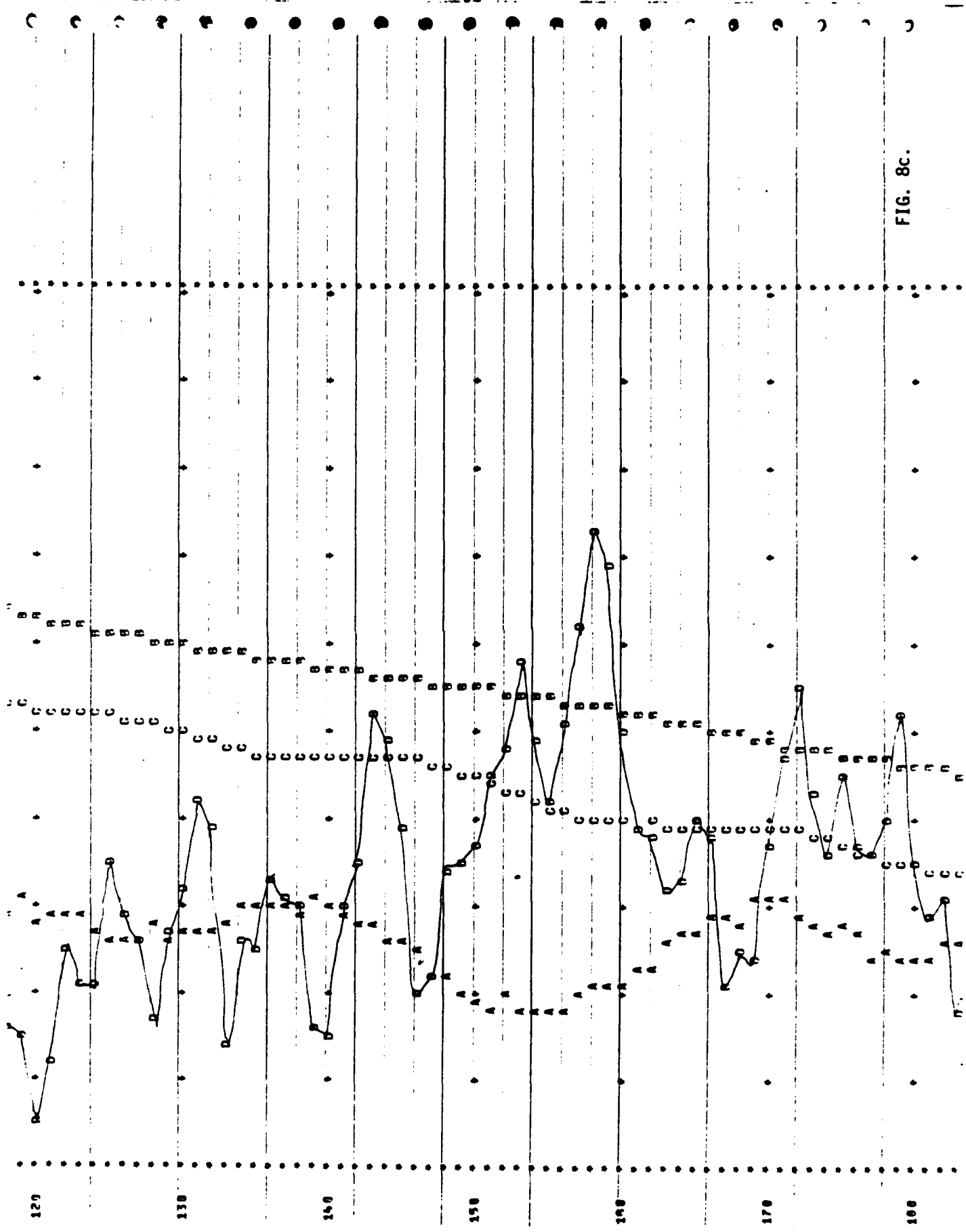


FIG. 8c.

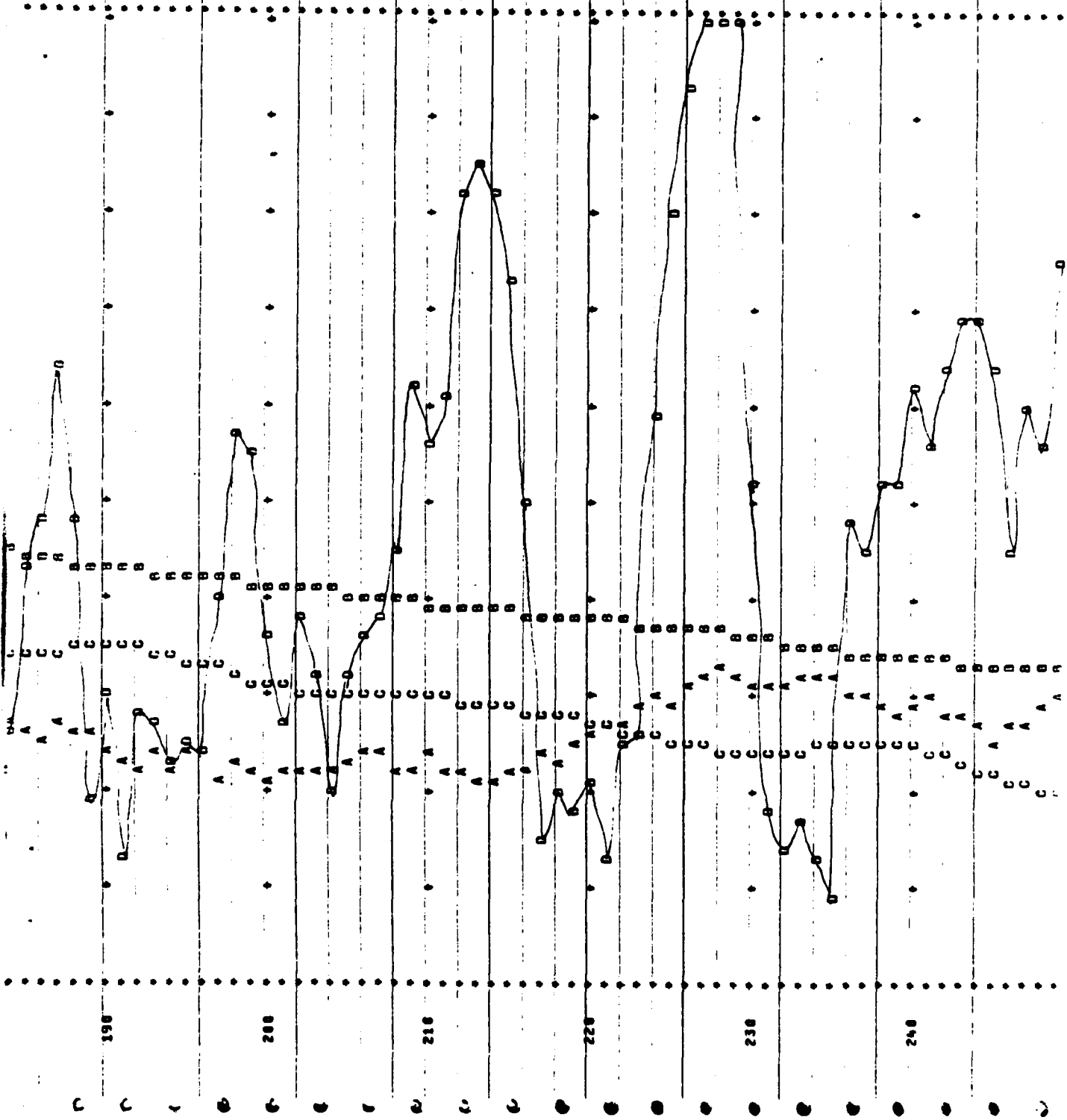


FIG. 8d.

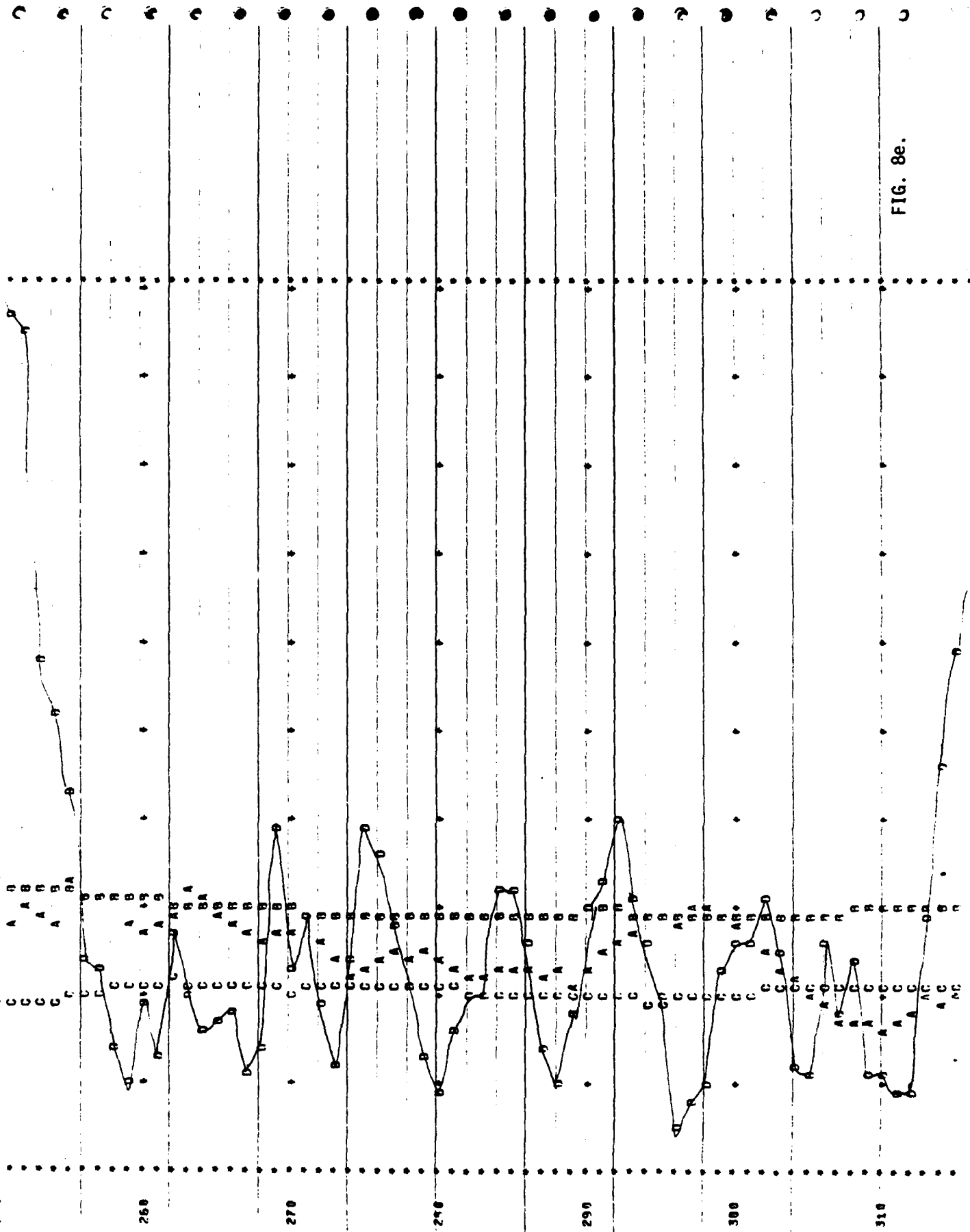


FIG. 8e.

FIG. 8f.

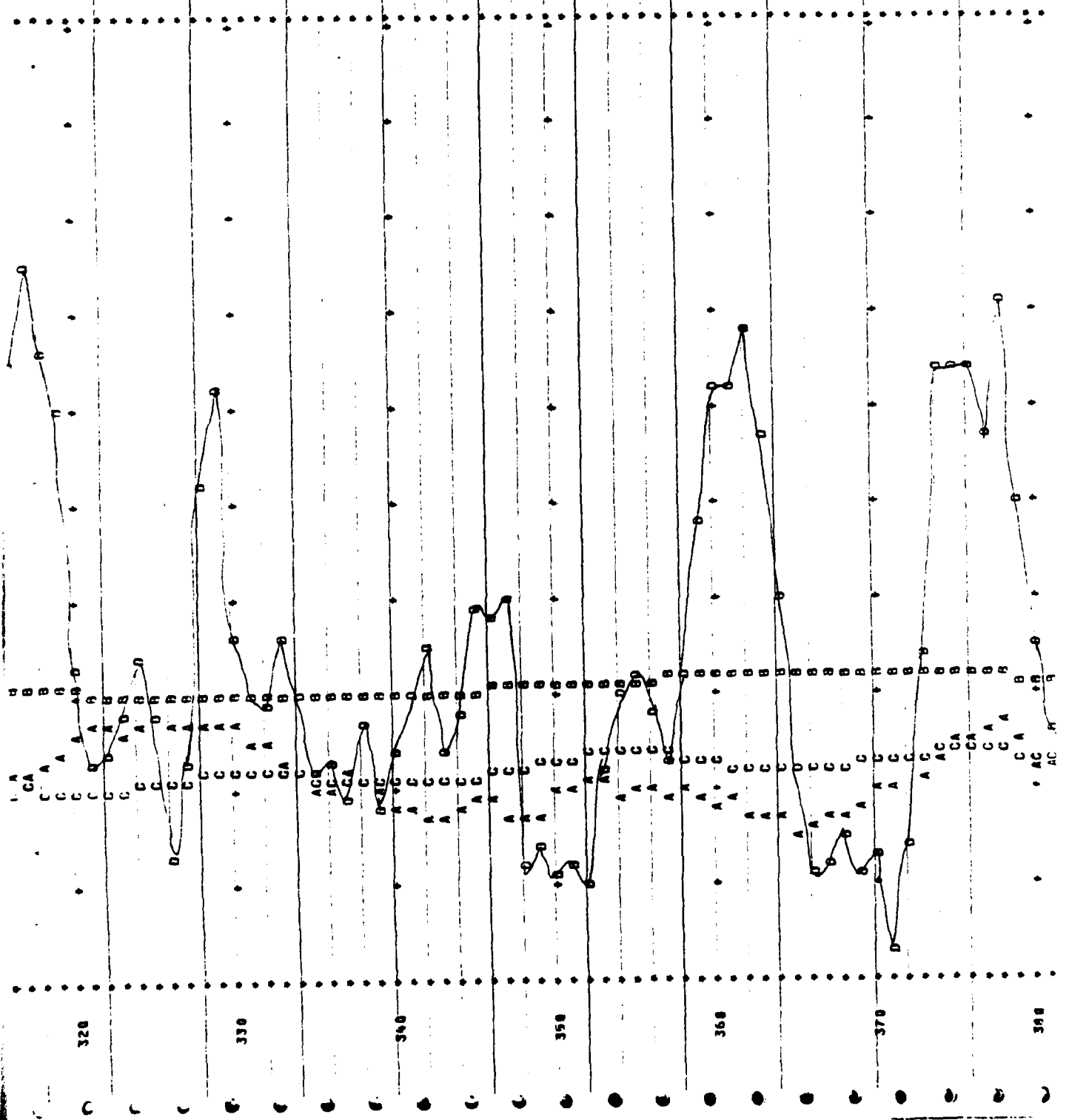




FIG. 9.  
EQUIPOTENTIAL CONTOURS

23.00 \*

JJJJJJJJJ \* JJJJJJJJJJ  
JJJJJJJJJ  
JJJJJJJJJ  
JJJJJJJJJ  
JJJJJJJJJ  
JJJJJJJJJ

J = 0 to +1

L = -1 to -2

JJ

JJ

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

N = -3 to -4

JJ

JJ

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

P = -5 to -6

JJ

JJ

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

R = -7 to -8

JJ

JJ

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

T = -9 to -10 volts

JJ

JJ

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

J

9-6-81-76 2) 4 2.61 3 5 1.4 6.1 7.0 13.0 19.0 21.0 21.6 2994.3 1243.9 .300

PARKTDC PROGRAM

The listing for the PARKTDC computer program is given in the following pages. (PARKTDC = Parker Time Dependent Charging.)



```

PROGRAM PARKTDC(INPUT=65,OUTPUT=513,TAPE1=513,TAPE2=513,      10
1 TAPE3=513,TAPE5=INPUT,TAPE6=OUTPUT)                          11
C*****
C                                                                    13
C   TIME-DEPENDENT CHARGING AND SPACECRAFT SHEATH IN           14
C   R=2 GEOMETRY                                                15
C                                                                    16
C   LEE W. PARKER / / / ERNEST G. HOLEMAN.   LEE W. PARKER, INC.
C   252 LEXINGTON ROAD, CONCORD MA. 01742                      18
C*****
C   VERSION -- E --.   MODE SELECTIBLE MODEL.                  19
C                                                                    20
C   MODE = 0 ==>  STANDARD CALCULATION MODEL                    21
C   MODE = 1 ==>  CLOUD MODEL                                    22
C   MODE = 2 ==>  WEIGHTED PARTICAL MODEL                       23
C                                                                    24
C   E. HOLEMAN.  FROGFAMMER.  LAST ALTERED  2 MAR 79           25
C*****
C*****
C   COMMON N, IV, IPFIRST, JFIRST, JEDGE, NR, NZA, NZ, NTOT, PI, IT, IR, IZ,
1  MONMAX, TIME, IPRINT, RADIUS, RR(11), ZZ(11), X(51), Q(51) ,   30
2  AREAM(11), DELZ(11), RI(11), ZCI(11)
COMMON/DEN/RFT, ZPT, AL1, BE1, EV, TVOLTS, DENST, NPTS, SPEED0, CURR,
1  IPART, PARTCL(2), DELTA, SPEED, QDISK, DZMIN, DTSEC, NINJ,    33
2  VSAVE(10,100), TVCHG, NRAN, NCESC, NCABS, NCACT, NCHGE(11), NCHGI(11),
3  KOUNT(51), XKOUN(51), MODE, WLOUD                               35
COMMON/FLO/DEBYE, DIELEC, IRELAX, PHI(11), DENCC                36
DIMENSION OATE(20)                                              37
DIMENSION ZA(11), ZB(11), ZAI(11), ZBI(11)                       38
DIMENSION PART1(2), PART2(2), X1(11), NETCH(51)                  39
DIMENSION KHISTI(51,12), KHISTE(51,12), AVGI(51), RMSDI(51),   40
1  AVGE(51), RMSDE(51), KHABSI(12), KHACTI(12),
2  KHABSE(12), KHACTE(12), LINE(14)                              42
DATA PART1/5H IONS, 5H /, PART2/5H ELEC, 5HTRONS/              43
NF(IZ, JR) = JR + NR*(IZ-1)                                     44
L=5                                                                45
M=6                                                                46
IV=1                                                                47
JV=2                                                                48
KV=3                                                                49
PI=3.1415926535898                                              50
EV=0.                                                             51
ELK=1H                                                            52
NRAN=0                                                            53
INIT=0                                                            54
NR0=80                                                            55
NZ0=55                                                            56
C                                                                    57
C*****
C   22 JUN 78 TEMPORARIES FOR INTERACTIVE MODE
C   CALL SSWTCH(6, ISW6)                                         58
                                                                    59
                                                                    60

```

	IF(ISW6.NE.1) GO TO 1800	61
	CALL DISCON(M)	62
	CALL DISCON(L)	63
	REWIND M	64
	REWIND L	65
	REWIND KV	66
C****		67
C		68
1800	READ(L,9999) DATE	69
9999	FORMAT(20A4)	70
	IF(EOF(L)) 99,1801	71
1801	WRITE(M,9998) DATE	72
9998	FORMAT(1H1,2EH TIME DEPENDENT CHARGING. ,20A4)	73
C		74
C	GEOMETRIC PARAMETERS	75
C		76
	READ(L,1111) JEDGE,NR,NZA,NZB,IRELAX,MODE	77
	NZ=NZA + NZB - 1	78
	NTOT=NZ*NR	79
	READ(L,2222) (RR(J),J=1,NR)	80
	READ(L,2222) (ZA(I),I=1,NZA)	81
	READ(L,2222) (ZB(I),I=1,NZB)	82
C		83
	READ(L,2222) (PHI(J),J=1,JEDGE)	84
	JEDGEP=JEDGE+1	85
	RADIUS=.5*(RR(JEDGEP) + RR(JEDGE))	86
	WRITE(M,9990) JEDGE,NR,NZA,NZB,IRELAX	87
	IF(MODE.EQ.0) WRITE(M,9991) MODE	88
	IF(MODE.EQ.1) WRITE(M,9992) MODE	89
	IF(MODE.EQ.2) WRITE(M,9993) MODE	90
	WRITE(M,9980) RADIUS,(J,RR(J),J=1,JEDGE)	91
	WRITE(M,9970) (J,RR(J),J=JEDGEP,NR)	92
	WRITE(M,9960) (I,ZA(I),I=1,NZA)	93
	WRITE(M,9950) (I,ZB(I),I=1,NZB)	94
	WRITE(M,9940) (J,PHI(J),J=1,JEDGE)	95
C		96
C	ITERATION AND SPACE CHARGE OPTIONS	97
C		98
	IT=0	99
	READ(L,1111) NPRINT,NPTS,IT,ITS,NIINJ,ISKIP,JSKIP,MASS	100
	WRITE(M,9930)NPRINT,NPTS,IT,ITS,NIINJ,ISKIP,JSKIP,MASS	101
	IF(IT.GT.0) READ(L,3333) NRAM	102
	READ(L,2222) DELTA,TIME,TVIIONS,TVELEC,DENCC,WCLOUD	103
	IF(IT.GT.0) READ(L,1111) (NCHGI(I),I=1,JEDGE)	104
	IF(IT.GT.0) READ(L,1111) (NCHGE(I),I=1,JEDGE)	105
	IF(IT.GT.0) READ(L,2222) (Q(I),I=1,NTOT)	106
	ITMAX=IT+ITS	107
	DIELEC=1.	108
C	DEBYE=0.	109
C	DEBYE= 743.39*SQRT(TVELEC/DENCC)/RADIUS	110
C		111

	IF (NPTS.EQ.0) READ(L,2222) RPT,ZPT,AL1,BE1,EV	112
	IF (NPTS.EQ.0) WRITE(M,9888) RPT,ZPT,AL1,BE1,EV	113
	IF (NPTS.EQ.0) TIME=0.	114
	DO 200 I=1,NZA	115
200	ZZ(I)=ZA(I)	116
	DO 210 I=2,NZB	117
	I1=NZA+I-1	118
210	ZZ(I1)=ZZ(I)	119
C		120
	WRITE(M,9870) (I,ZZ(I),I=1,NZ)	121
C		122
C	CALCULATE INTERSTITIAL GEOMETRY	123
C		124
	IR=NR+1	125
	IZA=NZA+1	126
	IZB=NZB+1	127
	IZ=NZA+NZB	128
	IZZ=1	129
C		130
	RI(I)=RR(I)	131
	RI(IR)=RR(NR)	132
	DO 300 J=2,NR	133
300	RI(J)=.5*(RR(J-1)+RR(J))	134
		135
	ZAI(1)=ZA(1)	136
	ZAI(IZA)=ZA(NZA)	137
	ZCI(1)=ZZ(1)	138
	ZCI(IZ)=ZZ(NZ)	139
	DO 350 I=2,NZA	140
	ZAI(I)=.5*(ZAI(I-1)+ZAI(1))	141
	IZZ=IZZ+1	142
	ZCI(IZZ)=ZAI(I)	143
350	CONTINUE	144
C		145
	ZBI(1)=ZB(1)	146
	ZBI(IZB)=ZB(NZB)	147
	DO 400 I=2,NZB	148
	ZBI(I)=.5*(ZBI(I-1)+ZBI(1))	149
	IZZ=IZZ+1	150
	ZCI(IZZ)=ZBI(I)	151
400	CONTINUE	152
C		153
	WRITE(M,9860) (J,RI(J),J=1,IR)	154
	WRITE(M,9890) (I,ZAI(I),I=1,IZA)	155
	WRITE(M,9840) (I,ZBI(I),I=1,IZB)	156
	WRITE(M,9835) (I,ZCI(I),I=1,IZZ)	157
C		158
	DO 450 N=1,NTOT	159
	KOUNT(N)=0	160
	XKOUNT(N)=0	161
	IF (IT.EQ.0) X(N)=0.	162

	IF(IT.EQ.0) Q(N)=0.	163
450	CONTINUE	164
C		165
	DO 500 J=1, NR	166
	R1=RI(J)	167
	IF(J.LT.NR) R2=RI(J+1)	168
	IF(J.EQ.NR) R2=RR(NR)	169
	AREA(J)=PI*(R2**2 - R1**2)	170
	NCHGE(J)=0	171
	IF(J.GT.JEDGE) GO TO 500	172
	IF(IT.EQ.0.AND.IRELAX.GT.0) PHI(J)=0.	173
500	CONTINUE	174
C		175
	DZMIN=10.**5	176
	DO 525 I=1, NZA	177
	Z2=ZAI(I)	178
	IF(I.LT.NZA) Z1=ZAI(I+1)	179
	IF(I.EQ.NZA) Z1=ZAI(IZA)	180
	DELZ(I)= Z2 - Z1	181
	DZMIN=AMIN1(DZMIN, DELZ(I))	182
525	CONTINUE	183
B		184
	XMASS=MASS	185
	IZZ=NZA	186
	DO 550 I=1, NZB	187
	IZZ=IZZ+1	188
	Z2=ZBI(I)	189
	IF(I.LT.NZB) Z1=ZBI(I+1)	190
	IF(I.EQ.NZB) Z1=ZBI(IZB)	191
	DELZ(IZZ)= Z2 - Z1	192
	DZMIN=AMIN1(DZMIN, DELZ(IZZ))	193
550	CONTINUE	194
C		195
	IFIRST=0	196
	JFIRST=0	197
C		198
C	MONMAX=0 FOR MONOENERGETIC BEAM	199
C	MONMAX=1 FOR ISOTROPIC MONOENERGETIC	200
C	MONMAX=2 FOR ISOTROPIC MAXWELLIAN	201
C		202
	MONMAX=2	203
	DENOM=1.	204
	IF(MONMAX.EQ.1) DENOM=4.	205
	IF(MONMAX.EQ.2) DENOM=2.*SQRT(PI)	206
C		207
	PCON=PI*DENOM*DELTA*DZMIN*RR(NR)**2	208
	IF(MONMAX.GT.0) PCON=PCON*(1.+(ZZ(1)-ZZ(NZ))/RR(NR))*2.	209
	CHCON=1.6E-7*PCON	210
	XIINJ=NIINJ	211
	COMPAR=PCON/XIINJ	212
	SPEEDE=5.93E7*SQRT(TVELEC)	213

	SPEEDI=5.93E7*SQRT(TVIONS/XMASS)	214
	CURRI=1.6E-7*DENCC*SPEEDI/DENOM	215
	CURRE=1.6E-7*DENCC*SPEEDI/DENOM	216
	PLASPI=1.11E-4/SQRT(DENCC/XMASS)	217
	PLASPE=1.11E-4/SQRT(DENCC)	218
	DEEYE=743.59*SQRT(TVELEC/DENCC)	219
C		220
C	SPEED IS SCALE VELOCITY	221
C		222
	SPEED=SPEEDI	223
	DTSEC=DELTA*QZMIN/SPEED*DENOM	224
C		225
	WRITE(N,9920) DELTA,DTSEC,TIME,TVIONS,TVELEC,DENCC,PLASPI,PLASPE,	
	1 DEEYE,PCON,COMPAR	227
	IF(MODE.EQ.1) WRITE(N,9921) WLCLOUD	228
C		229
	NPPP=(NTOT/300)+1	230
	DO 600 IPAGE=1,NPPP	231
	WRITE(N,9830)	232
	CALL LIST(2,IPAGE)	233
600	CONTINUE	234
C		235
C	INITIALIZE MATRICES FOR ANALYSIS OF VARIANCES	236
C		237
	DO 610 I=1,NTOT	238
	AVGI(I)=0.	239
	AVGE(I)=0.	240
	RMSDI(I)=0.	241
	RMSDE(I)=0.	242
	DO 610 J=1,12	243
	KHISTI(I,J)=0	244
	KHISTE(I,J)=0	245
610	CONTINUE	246
	DO 620 I=1,12	247
	KHABSI(I)=0	248
	KHACTI(I)=0	249
	KHABSE(I)=0	250
	KHACTE(I)=0	251
620	CONTINUE	252
C		253
	IF(IT.GT.0) GO TO 650	254
C		255
C	INITIALIZE SPACE CHARGE MATRIX	256
C	ASSUME QUIET START	257
C		258
	INIT=1	259
	VOL=PI*RR(NR)**2*(ZZ(1)-ZZ(NZ))	260
	RPART=DENCC*VOL	261
	NINJ=RPART/CCMPAR	262
	CALL CONNEN(N)	263
	WRITE(N,*) " INITIAL NINJ,RPART,COMPAR = ",NINJ,RPART,COMPAR	

	CALL DISCON(M)	265
	DO 630 IDUM=1,2	266
	IPART=IDUM	267
	SPEED0=SPEED1	268
	IF(IPART.EQ.2) SPEED0=SPEEDE	269
	CALL DENSITY(INIT)	270
	IFIRST=IFIRST+1	271
630	CONTINUE	272
	INIT=0	273
C		274
650	CONTINUE	275
B		276
C	COMPUTE IPRINT	277
O		278
	IPRINT=0	279
	IF(IPRINT.EQ.1) IPRINT=1	280
	IF(MOD(IT,JSKIP).LE.ISKIF-1) IPRINT=1	281
	IF(JFIRST.LE.9.OR.ITMAX-IT.LE.1) IPRINT=2	282
	IF(IT.EQ.0) IPRINT = 3	283
B		284
	IF(JFIRST.GT.0) TIME=TIME+DTSEC	285
	CALL FIELD	286
	IF(IPRINT.GT.1) WRITE(M,9790) IT	287
	IF(IT.GE.ITMAX.AND.ISHG.EQ.1) CALL CONNEG(M)	288
	CALL PLOT(X,RR,ZZ,NR,NZ,NRC,NZO,RADIUS,XMAX,IT)	289
G		290
	IF(IT.GE.ITMAX) GO TO 1080	291
C		292
C		293
O	IPART LOOP IPART=1,2, AND 3 FOR IONS, ELECTRONS, AND PHOTOELECTR	295
C		296
	DO 900 IDUM=1,2	296
	IPART=IDUM	297
	IF(IPART.EQ.1) TVOLTS=TVIONS	298
	IF(IPART.EQ.2) TVOLTS=-TVIONS/XMASS	299
	IF(IPART.EQ.1) SPEED0=SPEED1	300
	IF(IPART.EQ.2) SPEED0=SPEEDE	301
	IF(IPART.EQ.1) TVCHG=TVIGNS	302
	IF(IPART.EQ.2) TVCHG=-TVELEC	303
	IF(IPART.EQ.1) CURR=CURRI	304
	IF(IPART.EQ.2) CURR=CURRE	305
	IF(IPART.EQ.1) PARTCL(1)=PART1(1)	306
	IF(IPART.EQ.1) PARTCL(2)=PART1(2)	307
	IF(IPART.GT.1) PARTCL(1)=PART2(1)	308
	IF(IPART.GT.1) PARTCL(2)=PART2(2)	309
O		310
C****		311
C	CALL DENSITY TO GENERATE NEW PARTICLES AND TAKE NEXT TIME STEP.	313
C****		314
C		315
	IF(IPRINT.GT.1) WRITE(M,9750) PARTCL,SPEED0,CURR	315

C		316
	NINJ=NIINJ	317
	IF(IPART.EQ.2) NINJ=XIINJ*SPEEDE/SPEEDI+.1	318
C		319
	CALL DENSTY(INIT)	320
C		321
C	CDAMB=2.677*TCENCC*SQRT(TVELEC)	322
C		323
	JVAR=MOD(JFIRST/2,10)+1	324
	DO 700 N=1,NTOT	325
	IF(IPART.EQ.2) GO TO 690	326
	Q(N)=XKOUNT(N)*CHCON/XIINJ	327
	NETCH(N)=KOUNT(N)	328
	KHISTI(N,11)=KHISTI(N,11)+KOUNT(N)-KHISTI(N,JVAR)	329
	KHISTI(N,12)=KHISTI(N,12)+KOUNT(N)*KOUNT(N)-	330
	1 KHISTI(N,JVAR)*KHISTI(N,JVAR)	331
	KHISTI(N,JVAR)=KOUNT(N)	332
	GO TO 700	333
690	CONTINUE	334
	Q(N)=Q(N)-XKCON(N)*CHCON/XIINJ	335
	NETCH(N)=NETCH(N)-KOUNT(N)	336
	KHISTE(N,11)=KHISTE(N,11)+KOUNT(N)-KHISTE(N,JVAR)	337
	KHISTE(N,12)=KHISTE(N,12)+KOUNT(N)*KOUNT(N)-KHISTE(N,JVAR)*	
	1 KHISTE(N,JVAR)	339
	KHISTE(N,JVAR)=KOUNT(N)	340
700	CONTINUE	341
C		342
C	ADD TO REMAINING VARIANCE SUMS	343
C		344
	IF(IPART.EQ.2) GO TO 720	345
	NIABS=NCABS	346
	NIAGT=NCACT	347
	NIESC=NCESC	348
	KHABSI(11)=KHABSI(11)+NCABS-KHABSI(JVAR)	349
	KHACTI(11)=KHACTI(11)+NCACT-KHACTI(JVAR)	350
	KHABSI(12)=KHABSI(12)+NCABS*NCABS-KHABSI(JVAR)*KHABSI(JVAR)	
	KHACTI(12)=KHACTI(12)+NCACT*NCACT-KHACTI(JVAR)*KHACTI(JVAR)	
	KHABSI(JVAR)=NCABS	353
	KHACTI(JVAR)=NCACT	354
	GO TO 730	355
720	CONTINUE	356
	NEABS=NCABS	357
	NEACT=NCACT	358
	NEESC=NCESC	359
	KHABSE(11)=KHABSE(11)+NCABS-KHABSE(JVAR)	360
	KHACTE(11)=KHACTE(11)+NCACT-KHACTE(JVAR)	361
	KHABSE(12)=KHABSE(12)+NCABS*NCABS-KHABSE(JVAR)*KHABSE(JVAR)	
	KHACTE(12)=KHACTE(12)+NCACT*NCACT-KHACTE(JVAR)*KHACTE(JVAR)	
	KHABSE(JVAR)=NCABS	364
	KHACTE(JVAR)=NCACT	365
730	CONTINUE	366

C		367
	IFIRST=IFIRST+1	368
	JFIRST=JFIRST+1	369
900	CONTINUE	370
	NCHGSE=0	371
	NCHGSI=0	372
	DO 710 J=1,JEDGE	373
	NCHGSE=NCHGSE+NCHGE(J)	374
	NCHGSI=NCHGSI+NCHGI(J)	375
710	CONTINUE	376
C		377
	QDISK=FLAG*(NCHGSI-NCHGSE)*CHOOON/PI*INJ	378
C		379
	IF(IRELAX.EQ.0) GO TO 920	380
	DO 910 J=1,JEDGE	381
C		382
C	PHI RELATION WILL BE ESTABLISHED LATER. THIS IS INCORRECT.	
C		384
	PHI(J)=PI/2.*.9*QDISK/RADIUS	385
910	CONTINUE	386
920	CONTINUE	387
	N1=0	388
	N2=0	389
	IF(IPRINT.GT.0) WRITE(M,9740) (BLK,I,I=1,NR)	390
	DO 930 I=1,NZ	391
	N1=N2+1	392
	N2=N2+NR	393
	IF(IPRINT.GT.0) WRITE(M,9730) I,(NETCH(J),J=N1,N2)	394
930	CONTINUE	395
C		396
C	BEGIN CALCULATION OF VARIANCES	397
C		398
	IF(ISW6.EQ.1.AND.JFIRST.EQ.2) CALL CONNec(M)	399
	DELT=NINO*(JFIRST/2,10)	400
	AVIABS=FLOAT(KHABSI(11))/DELT	401
	AVIACT=FLOAT(KHACTI(11))/DELT	402
	RMIABS=SQRT(FLOAT(KHABSI(12))/DELT-AVIABS*AVIABS)	403
	RMIACT=SQRT(FLOAT(KHACTI(12))/DELT-AVIACT*AVIACT)	404
	AVEABS=FLOAT(KHABSE(11))/DELT	405
	AVEACT=FLOAT(KHACTE(11))/DELT	406
	RMEABS=SQRT(FLOAT(KHABSE(12))/DELT-AVEABS*AVEABS)	407
	RMEACT=SQRT(FLOAT(KHACTE(12))/DELT-AVEACT*AVEACT)	408
	IF(IPRINT.GT.0) WRITE(M,3002)	409
	IF(ISW6.EQ.1) CALL CONNec(M)	410
3002	FORMAT(///25X,21HSUMMARY FOR THIS STEP,16X,19HTEN-STEP RMS ERRORS,	
1	17X,17HTEN-STEP AVERAGES/22X,4HIONS,12X,9HELECTRONS,1X,2(3X,217X,	
2	1HI,7X,1HE))/1X,5HCYCLE,3X,4HTIME,2(4X,14HABS ESC ACT),	
3	2(8X,3HABS,5X,3HACT,5X,3HACT),2X,4HMAX)	414
	IPLAST=IPRINT	415
	WRITE(M,3001) IT,TIME,NIABS,NIESC,NIACT,NEABS,NEESC,NEACT,	
1	RMIABS,RMEABS,RMIACT,RMEACT,AVIABS,AVEABS,AVIACT,AVEACT,XMAX	

```

WRITE(KV,3001) IT,TIME,NIABS,NIE SC,NIACT,NEABS,NEESC,NEACT,
I RNIABS,RHEABS,RHIACT,RHEACT,AVIABS,AVEABS,AVIACT,AVEACT,XMAX
3001 FORMAT(1X,I4,1PE18.3,1X,I3,2I6,3X,I3,2I6,3X,0P4F8.1,3X,4F8.1,F8.3)
IF(IISW6.EQ.1) CALL DISCON(M)
CD 421
IT=IT+1 422
GO TO 650 423
C 424
99 CONTINUE 425
C 426
C**** 427
C TRANSFER SUMMARY FROM FILE=KV TO OUTPUT 428
C**** 429
CALL DISCON(M) 430
WRITE(M,5555) 431
REWIND KV 432
WRITE(M,3002) 433
940 CONTINUE 434
READ(KV,4444) LINE 435
IF(EOF(KV)) 960,950 436
950 CONTINUE 437
WRITE(M,4444) LINE 438
GO TO 940 439
960 CONTINUE 440
C 441
S CREATE RESTART FILE ON TAPE1 442
C 443
IF(IV.EQ.2) GO TO 1010 444
IV=1 445
JV=2 446
REWIND IV 447
REWIND JV 448
C 449
C COPY TAPE2 TO TAPE1 450
C 451
1020 CONTINUE 452
READ(JV) INJA,((VSAVE(I,J),I=1,10),J=1,INJA) 453
IF(EOF(JV)) 1010,1030 454
1030 CONTINUE 455
WRITE(IV) INJA,((VSAVE(I,J),I=1,10),J=1,INJA) 456
GO TO 1020 457
C 458
1010 CONTINUE 459
C 460
C CREATE RECORD TO BE COPIED TO INPUT FILE FOR RESTART 461
C 462
REWIND JV 463
WRITE(JV,9999) DATE 464
WRITE(JV,1111) JEDGE,NR,NZA,NZB,IRELAX,MODE 465
WRITE(JV,2222) (RR(J),J=1,NR) 466
WRITE(JV,2222) (ZA(I),I=1,NZA) 467

```

WRITE(JV,2222) (Z9(I),I=1,NZ9)	469
WRITE(JV,2222) (PHI(J),J=1,JEDGE)	470
WRITE(JV,1111) NPRINT,NPTS,IT,ITS,NIINJ,ISKIF,JSKIP,MASS	471
CALL RANGET(NRAN)	472
WRITE(JV,3333) NFAN	473
WRITE(JV,2222) DELTA,TIME,TVIONS,TVELEC,DENCC,WGLOUO	474
WRITE(JV,1111) (NCHGI(I),I=1,JEDGE)	475
WRITE(JV,1111) (NCHGE(I),I=1,JEDGE)	476
WRITE(JV,2222) (Q(I),I=1,NTOT)	477
<del>D</del>	478
STOF11	479
1122 FORMAT(2I5,1F7E10.3/( 0E10.3))	480
1111 FORMAT(16I5)	481
2222 FORMAT(8E10.3)	482
3333 FORMAT(1X,020)	483
4444 FORMAT(13A10,A6)	484
5555 FORMAT(1H1)	485
9990 FORMAT(/1X,I3,33H COLUMNS (R-VALUES) WITHIN RADIUS /	486
1 1X,I3,25H COLUMNS (R-VALUES) TOTAL /	487
2 1X,I3,46H ROWS (Z-VALUES) ABOVE AND INCLUDING Z=0 PLANE /	488
3 1X,I3,46H ROWS (Z-VALUES) BELOW AND INCLUDING Z=0 PLANE /	489
5 1X,I3,44H = IRELAX (RELAXATION, =0 IF BODY POTENTIAL,	490
6 10H IS FIXED,,48H =1 IF BODY POTENTIAL IS FLOATING, AND EQUILIBR	491
7 31HUM VALUE IS TO BE CALCULATED,))	492
9991 FORMAT(/1X,I3,41H = MODE, SELECTS NORMAL CALCULATION MODE)	493
9992 FORMAT(/1X,I3,29H = MODE, SELECTS GLOUO MODEL)	494
9993 FORMAT(/1X,I3,41H = MODE, SELECTS WEIGHTED PARTICAL MODEL)	495
9980 FORMAT(/124H R-VALUES WITHIN RADIUS=1PE15.4,3H CM/(1X,	496
1 I3,1PE15.4,3H CM))	497
9970 FORMAT(/124H R-VALUES OUTSIDE RADIUS/(1X,I3,1PE15.4,3H CM))	498
9960 FORMAT(/134H Z-VALUES POSITIVE ABOVE Z=0 PLANE/(1X,I3,1PE15.4,	499
1 3H CM))	500
9950 FORMAT(/134H Z-VALUES NEGATIVE BELOW Z=0 PLANE/(1X,I3,1PE15.4,	501
1 3H CM))	502
9940 FORMAT(/127H SURFACE POTENTIALS ON BODY/(1X,I3,1PE15.4,	503
1 6H VOLTS))	504
9930 FORMAT(/138H INTEGER INPUTS FOR SHEATH CALCULATION /1X,	505
1 I3,37H = NPRINT (TO PRINT TRAJECTORY STEPS)/1X,	506
2 I3,39H (TO COMPUTE ONE OR MORE POINTS)/1X,	507
4 2I5,44H = IT AND ITS (ITERATION INDEX AND NUMBER OF,	508
5 12H ITERATIONS)/1X,	509
6 I5,30H IONS GENERATED PER TIME STEP/1X,	510
7 I5,35H PAGES OF SUMMARY PRINTED FOR EACH,I5,6H STEPS,	511
8 I3,30H = MASS (MASS OF COMPUTER ION))	512
9920 FORMAT(/127H EMISSION AND PLASMA INPUTS /	513
1 1X,F10.3,30H = DELTA = INITIAL TIME STEP (MEANS,	514
2 1PE15.4,3X,8HSECONDS)/	515
3 1X,0FF10.3,24H = TIME = INITIAL TIME/	516
4 1X,F10.3,49H = TVIONS = AMBIENT-ION TEMPERATURE IN VOLTS/	517
5 1X,F10.3,49H = TVELEC = AMBIENT-ELECTRON TEMPERATURE IN VOLTS/	518
6 1X,F10.3,50H = DENCC = AMBIENT-ELECTRON DENSITY IN NO. PER CC /	519

7	20X,25H(ION PLASMA PERIOD =,1PE15.4,3X,8HSECONDS)/	520
8	20X,25H(ELECTRON PLASMA PERIOD =,E15.4,3X,8HSECONDS)/	521
9	20X,25H(ELECTRON DEBYE LENGTH =,E15.4,2X,3HCM)/	522
A	1X,1PE15.4,50H = PCON = NUMBER OF REAL IONS INJECTED PER CYCLE/	
B	1X,E15.4,53H = COMPAR = NUMBER OF REAL IONS/ELECTRONS PER COMPUTE	
C	IONR PARTICLE)	525
9921	FORMAT(1X,1PE15.4,29H = WLOUD = CLOUD HALF WIDTH)	52E
9880	FORMAT(/1X,29H INPUTS FOR SINGLE TRAJECTORY/	527
1	1X,F10.3,19H = RPT = R-POSITION/1X,F10.3,19H = ZPT = Z-POSITION/	
2	1X,F10.3,30H = AL1 = POLAR ANGLE (DEGREES)/	529
3	1X,F10.3,34H = BE1 = AZIMUTHAL ANGLE (DEGREES)/	530
4	1X,F10.3,24H = EV = ENERGY IN VOLTS)	531
9870	FORMAT(/13H ALL Z-VALUES/(1X,I3,1PE15.4))	532
9860	FORMAT(/1X,21HINTERSTITIAL R-VALUES/(1X,I3,1PE15.4))	533
9850	FORMAT(/1X,30HINTERSTITIAL Z-VALUES POSITIVE/(1X,I3,1PE15.4))	
9840	FORMAT(/1X,30HINTERSTITIAL Z-VALUES NEGATIVE/(1X,I3,1PE15.4))	
9835	FORMAT(/1X,43HINTERSTITIAL Z-VALUES NEGATIVE AND POSITIVE/	
1	(1X,I3,1PE15.4))	537
9830	FORMAT(1H1/6X,1HN,5X,4HR(N),4X,4HZ(N)/)	538
9790	FORMAT(/1X,I3,16H ORDER POTENTIAL)	539
9780	FORMAT(/1X,I3,25H ORDER FLUXES AND CHARGES)	540
9750	FORMAT(/1X,23HGOING INTO DENSITY WITH ,2A5,5X,	541
1	13H WITH SPEED =,1PE15.4,7H CM/SEC,5X,22H AND CURRENT DENSITY =,	
2	E15.4,14H PICOAMP/CM**2)	543
9740	FORMAT(/5X27HSUMMARY OF NET SPACE CHARGE//12X,9(A2,	544
1	2HR(,I1,1H),11(A1,2HR(,I2,1H)))	545
9730	FORMAT(/5X,2HZ(,I2,1H),20I6)	546
	END	547

	SUBROUTINE LIST(LST,IP)	548
C		549
C	PRINT ARRAYS	550
C		551
	COMMON M,IV,IFIRST,JFIRST,JEDGE,NR,NZA,NZ,NTOT,PI,IT,IR,IZ,	
	1 NONMAX,TIME,IPRINT,RADIUS,RR(11),ZZ(11),X(51),Q(51),	553
	2 AREA(11),DELZ(11),RI(11),ZCI(11)	554
	DIMENSION KOUT(5),ROUT(5),ZOUT(5)	555
C		556
	DO 500 LINE=1,60	557
	DO 200 NP=1,5	558
	KP=LINE + (NF-1)*60 + (IF-1)*300	559
	IF(KP.GT.NTOT.AND.NP.EQ.1) RETURN	560
	IF(KP.GT.NTOT) GO TO 300	561
	NMAX=NP	562
	KOUT(NP)=KP	563
	IKP=(KP-1)/NF+1	564
	JKP=MOD(KP-1,NR)+1	565
C		566
	IF(LST.EQ.1) ROUT(NP)=Q(KP)	567
	IF(LST.EQ.1) ZOUT(NP)=X(KP)	568
	IF(LST.EQ.2) ROUT(NP)=RR(JKP)	569
	IF(LST.EQ.2) ZOUT(NP)=ZZ(IKP)	570
		571
5		572
200	CONTINUE	573
C		574
300	CONTINUE	575
	IF(LST.NE.1) GO TO 450	576
C\$1300	GO TO (400,450),LST	577
5		578
400	WRITE(M,1000) (KOUT(NP),ROUT(NP),ZOUT(NP),NP=1,NMAX)	579
1000	FORMAT(5(I5,1P2E11,3))	580
	GO TO 500	581
C		582
450	WRITE(M,3000) (KOUT(NP),ROUT(NP),ZOUT(NP),NP=1,NMAX)	583
3000	FORMAT(5(I8,2F8,3))	584
500	CONTINUE	585
	RETURN	586
	END	

	SUBROUTINE FIELD	587
C		588
C	COMPUTE ELECTRIC FIELD FROM GIVEN FIXED POTENTIALS OR CHARGES ON S	
C	CONSTRUCT COEFFICIENTS IN LINEAR DIFFERENCE EQUATIONS. SOLVE BY RE	
C		591
	COMMON N, IV, IPFIRST, JPFIRST, JEDGE, NR, NZA, NZ, NTOT, PI, IT, IR, IZ,	
	1 MONMAX, TIME, IPRINT, RADIUS, RR(11), ZZ(11), X(51), Q(51),	593
	2 AREAH(11), DELZ(11), RI(11), ZCI(11)	594
	COMMON/FLO/DEBYE, DIELEC, IRELAX, PHI(11), DENGCC	595
	COMMON/ARRAYC/ INDX(51,4), COEF(51,5)	596
	COMMON/CCC/N, I, J, NN, NS, NE, NH, NSEW, CN, CS, CE, CH, CC, C, AREA, R, Z	
	1 , VOLUME, CHARG	598
	REAL KKK1, KKK2	599
	INTEGER DIA	600
	DATA KKK1/5HALPH= /	601
	DATA KKK2/5HEET = /	602
	DATA CN/5HNORTH/	603
	DATA CS/5HSOUTH/	604
	DATA CE/5HEAST /	605
	DATA CH/5HWEST /	606
C		607
C	ASSUME ASYMPTOTIC BEHAVIOR AT INFINITY (MONOPOLE+DIPOLE, ETC.)	
C		609
	$ALP(RRR, ZZZ) = ABS((A+3.*B*ZZZ/RS) * ZZZ/RS - B/RS) / (A+B*ZZZ/RS)$	611
C		612
	$BEP(RRR, ZZZ) = ABS((A+3.*B*ZZZ/RS) * RRR/RS - (A+B*ZZZ/RS))$	613
C		614
C	DIA=POSITIVE INTEGER FOR DIAGNOSTIC OUTPUT	
	DIA=1	615
	DIA=0	616
	IF (JPFIRST.EQ.0) WRITE (N, 9000) IT, TIME	617
	9000 FORMAT(1H1/10H0FIELD CALCULATION, 5X, 4HIT =, I3, 5X, 9HAT TIME =,	
	1 1PE10.3////1X, 17HCOEFFICIENT ARRAY)	619
C		620
C	PURE DIPOLE	621
	A=0.	622
	B=1.	623
C		624
C	PURE MONOPOLE	625
	A=1.	626
	B=0.	627
C		628
	ZOLD=ZZ(1)	629
	N=0	630
	DO 1000 IDM=1, NZ	631
	DO 1000 JDM=1, NR	632
	N=N+1	633
	R=RR(JDM)	634
	Z=ZZ(IDM)	635
	RS=R**2 + Z**2	636
	IF (IDM.LT.NZA) VOLUME=AREAH(JDM)*DELZ(IDM)	637

```

IF(IOM.GT.NZA) VOLUME=AREA*(JOM)*DELZ(IOM+1) 638
IF(IOM.EQ.NZA) VOLUME=AREA*(JOM)*(DELZ(NZA)+DELZ(NZA+1)) 639
CHARG=Q(N) 640
IF(Z.GE.ZOLD.AND.N.GT.1) GO TO 100 641
ZOLD=Z 642
IF(JFIRST.EQ.0) WRITE(N,8800) IOM,Z 643
8800 FORMAT(/,1X,2HZ(,I2,2H)=,F10.3/ 644
1 20X,1HN,17X,1HW,17X,1HC,17X,1HE,17X,1HS,15X,6I,CHARGE, 645
2 4X,13HNORMAL CHARGE) 646
C 647
100 CC=0. 648
I=IOM 649
J=JOM 650
DO 200 JOUM=1,2 651
NSEW=JOUM 652
C=0. 653
CALL CNSEW(1) 654
C 655
IF(NSEW.EQ.1) OO=ON 656
IF(NSEW.EQ.2) OO=OS 657
IF(DIA.GT.0) WRITE(N,8888) N,I,J,OO,AREA,C 658
8888 FORMAT(1X,16HN,I,J,OO,AREA,C,14,2X,2I3,1X,A5,1P2E16.4) 659
C 660
GG=GG+C 661
IF(C.GT.0.) GO TO 150 662
C 663
ALPH=0. 664
IF(Z.EQ.ZZ(1).OR.Z.EQ.ZZ(NZ)) ALPH=ALF(R,Z) 665
ALBE=KKK1 666
C 667
IF(DIA.GT.0) WRITE(N,7777) ALBE,N,I,J,OO,AREA,ALPH 668
7777 FORMAT(1X,14HN,I,J,OO,AREA,,A5,14,2X,2I3,1X,A5,1P2E16.4) 669
C 670
IF(ALPH.GT.0.) GG=GG + AREA*ALPH 671
C 672
150 IF(NSEW.EQ.1) GN=C 673
IF(NSEW.EQ.2) CS=C 674
200 CONTINUE 675
C 676
DO 300 JOUM=1,2 677
NSEW=JOUM 678
C=0. 679
CALL CNSEW(2) 680
IF(NSEW.EQ.1) OO=OE 681
IF(NSEW.EQ.2) OO=OW 682
IF(DIA.GT.0) WRITE(N,8888) N,I,J,OO,AREA,C 683
C 684
GG=GG+C 685
IF(C.GT.0.) GO TO 350 686
C 687
BET=0. 688

```

	IF(R.EQ.RR(NR)) BET=BEF(R,Z)	689
	ALBE=KKKZ	690
C		691
	IF(DIA.GT.0) WRITE(W,7777) ALBE,N,I,J,00,AREA,BET	692
	IF(BET.GT.0.) CC=CC + AREA*BET	693
C		694
350	IF(NSEW.EQ.1) CE=C	695
	IF(NSEW.EQ.2) CW=C	696
300	CONTINUE	697
	CALL ARRAYS	698
1000	CONTINUE	699
C		700
	CALL RELAX	701
	RETURN	702
	END	703

	SUBROUTINE CNSEN(IGO)	704
C		705
C	COEFFICIENTS NORTH AND SOUTH	706
C		707
	COMMON N, IV, IFIRST, JFIRST, JEDGE, NR, NZA, NZ, NTOY, PI, IT, IR, IZ,	
1	HONMAX, TIME, IPRINT, RADIUS, RR(11), ZZ(11), X(51), O(51),	709
2	AREAH(11), DELZ(11), RI(11), ZCI(11)	710
	COMMON/FLO/BESVE, DIELEC, IRELAX, PHI(11), DENGC	711
	COMMON/CCC/N, I, J, NN, NS, NE, NW, NSEW, CN, CS, CE, CW, CC, C, AREA, R, Z	
1	VOLUME, CHARG	713
	NF(IZ, JR) = JR + NR*(IZ-1)	714
C		715
	IF(IGO.GT.1) GO TO 550	716
C		717
	AREA=0.	718
	BZ=0.	719
	IF(Z.EQ.0.) GO TO 100	720
	IF(Z.GT.0.) AREA=AREAH(J)	721
	IF(Z.LT.0.) AREA=DIELEC*AREAH(J)	722
	GO TO 200	723
C		724
100	IF(NSEW.EQ.1) AREA=AREAH(J)	725
	IF(NSEW.EQ.2) AREA=DIELEC*AREAH(J)	726
C		727
200	IF(NSEW.EQ.1) GO TO 300	728
	IF(NSEW.EQ.2) GO TO 400	729
	RETURN	730
C		731
300	NN=0	732
	IF(I.EQ.1) RETURN	733
	NN=NF(I-1, J)	734
	BZ=ZZ(I-1) - Z	735
	GO TO 500	736
C		737
400	NS=0	733
	IF(I.EQ.NZ) RETURN	739
	NS=NF(I+1, J)	740
	BZ=Z - ZZ(I+1)	741
C		742
500	IF(BZ.GT.0.) G=AREA/BZ	743
	RETURN	744
C		745
C		746
550	CONTINUE	747
C		748
	AREA=0.	749
	DR=0.	750
	IF(Z.EQ.0.) AREA=DELZ(NZA) * DIELEC*DELZ(NZA+1)	751
	IF(Z.GT.0.) AREA=DELZ(I)	752
	IF(Z.LT.0.) AREA=DIELEC*DELZ(I+1)	753
C		754

	IF(NSEW.EQ.1) GO TO 600	755
	IF(NSEW.EQ.2) GO TO 700	756
	RETURN	757
C		758
600	NE=0	759
	IF(J.EQ.NR) AREA=2.*PI*RR(J)*AREA	760
	IF(J.EQ.NR) RETURN	761
	NE=NF(I,J+1)	762
	DR=RR(J+1) - R	763
	PERIM=PI*(RR(J+1) + R)	764
	GO TO 800	765
C		766
700	NW=0	767
	IF(J.EQ.1) AREA=0.	768
	IF(J.EQ.1) RETURN	769
	NW=NF(I,J-1)	770
	DR=R - RR(J-1)	771
	PERIM=PI*(R + RR(J-1))	772
C		773
800	AREA=PERIM*AREA	774
	IF(DR.GT.0.) C=AREA/DR	775
	RETURN	776
	END	777

	SUBROUTINE ARRAYS	778
C		779
C	SAVE COEFFICIENTS AND INDICES IN ARRAYS	780
C		781
	COMMON M, IV, IFIRST, JFIRST, JEDGE, NR, NZA, NZ, NTOT, PI, IT, IR, IZ,	
	1 NONMAX, TIME, IPRINT, RADIUS, RR(11), ZZ(11), X(51), Q(51),	783
	2 AREA(11), DELZ(11), RI(11), ZCI(11)	784
	COMMON/FLO/DEBYE, DIELEC, IRELAX, PHI(11), DENG	785
	COMMON/ARRAYC/ INDX(51,4), COEF(51,5)	786
	COMMON/CCG/N, I, J, NN, NS, NE, NW, NSEW, CN, CS, CE, CW, CG, C, AREA, R, Z	
	1, VOLUME, CHARG	788
B		789
C		790
	IF(CG.EQ.0.) CALL ABNOR(10H STOP111)	791
	IF(CC.EQ.0.) STOP111	792
C	CN=NORTH (=+Z) NEIGHBOR = COEF1*CC	793
C		794
	COEF(N,1)=CN/CC	795
		796
C	CS=SOUTH (= -Z) NEIGHBOR = COEF2*CC	797
C		798
	COEF(N,2)=CS/CC	799
		800
C	CE=EAST (=+R) NEIGHBOR = COEF3*CC	801
C		802
	COEF(N,3)=CE/CC	803
		804
C	CW=WEST (= -R) NEIGHBOR = COEF4*CC	805
C		806
	COEF(N,4)=CW/CC	807
		808
C	WHERE CC=CENTRAL-POINT COEFFICIENT	809
C		810
	COEF(N,5)=.9*4.*PI*CHARG/CC	811
C58	IF(DEBYE.GT.0.) COEF(N,5)=CH(N)/CC/DEBYE**2*.9	812
C		813
C	CH=NET POSITIVE CHARGE IN N-TH BOX (FACTOR .9 FOR CH IN FICOCCULOM	
C	POTENTIAL IN VOLTS, LENGTH IN CM)	815
C		816
	INDX(N,1)=NN	817
	INDX(N,2)=NS	818
	INDX(N,3)=NE	819
	INDX(N,4)=NW	820
	CHARGO=1.6E-7*VOLUME*DENG	821
C		822
C		823
	IF(JFIRST.EQ.0) WRITE(M,1000) NN,CN,NW,CW,N,CC,NE,CE,NS,CS,CHARG,	
	1 CHARGO	825
1000	FORMAT(/13X,5(1X,1H(,14,2H)=,1PE10.4),2E14.4)	826
	RETURN	827
	END	828

```

SUBROUTINE RELAX                                829
C
C POINT-SUCCESSIVE RELAXATION TO SOLVE EQUATIONS 830
C
COMMON H, IV, IFIRST, JFIRST, JEDGE, NR, NZA, NZ, NTOT, PI, IT, IR, IZ,
1  NOMAX, TIME, IPRINT, RADIUS, RR(11), ZZ(11), X(51), Q(51), 834
2  AREA(11), DELZ(11), RI(11), ZCI(11) 835
COMMON/FLO/DBYE, DIELEC, IRELAX, PHI(11), DENGCC 836
COMMON/ARRAYC/ INDX(51,4), COEF(51,5) 837
C
OMEGA=1.9 838
EPS=1.E-9 839
ITMAX=2000 840
ITR=0 841
IPROLD=0 842
IGO=1 843
IF(JFIRST.GT.0.AND.IT.GT.0) GO TO 200 844
DO 100 K=1,NTOT 845
100 X(K)=0. 846
C
200 CONTINUE 847
ITR=ITR+1 848
DELTAM=0. 849
C
DO 500 N=1,NTOT 850
X1=X(N) 851
I=(N-1)/NR+1 852
J=MOD(N-1,NR)+1 853
IF(J.GT.JEDGE.OR.ZZ(I).NE.0.) GO TO 400 854
C
C SET SURFACE POTENTIALS 855
C
X(N)=PHI(J) 856
GO TO 500 857
400 CONTINUE 858
C
SUM=COEF(N,5) 859
DO 300 KK=1,4 860
INDEX=INDX(N, KK) 861
IF(INDEX.GT.0) SUM=SUM + COEF(N, KK) * X(INDEX) 862
300 CONTINUE 863
C
XIN)=SUM 864
X(N)=OMEGA*X(N) + (1.-OMEGA)*X1 865
DELTA=ABS(X(N)-X1) 866
IF(X1.NE.0.) DELTA=ABS((X(N)-X1)/X1) 867
IF(DELTA.GT.DELTAM) DELTAM=DELTA 868
C
500 CONTINUE 869
IF(ITR.GT.ITMAX) WRITE(M,8888) ITR 870

```

	IF(ITR.GT.ITMAX) GO TO 700	880
880	FORMAT(//15H MORE THAN, I4, 11H ITERATIONS)	881
C		882
	IPR=ITR/500	883
	IF(IPR.LE.IPROLD) GO TO 600	884
	IPROLD=IPR	885
	GO TO 880	886
C		887
600	IF(DELTAM.GT.EPS) GO TO 200	888
C		889
C	ITERATION CONVERGED. PRINT AND EXIT.	890
C		891
700	IGO=2	892
	IF(IPRINT.EQ.0) GO TO 1000	893
800	NFPP=(NTOT/300) + 1	894
	DO 900 IPAGE=1,NFPP	895
	WRITE(M,7777) IT,TIME,ITR,EPS,DELTAM,OMEGA	896
	CALL LIST(1,IPAGE)	897
900	CONTINUE	898
	7777 FORMAT(1H1/15H8FIELD CALCULATION,5X,4HIT =, I3,5X,9HAT TIME =, 1 1PE10.3//15H SOLUTION AFTER, I6,2X,25HITERATIONS WITH TOLERANCE, 2 0PF12.0, 0X,10HMAXIMUM DIFFERENCE, F12.0, 0X,6HOMEGA =, F8.5/4X, 3 1HN,2X,4HQ(N),7X,4HX(N)//)	902
C		903
	IF(IGO.EQ.1) GO TO 600	904
1000	CONTINUE	905
	RETURN	906
	END	907

	SUBROUTINE INTERP(R,Z,JG,IG,PHIC,RI,ZI,IR,IZ,INT)	908
C		909
C	INTERPOLATE IN R-Z GRID TO FIND POTENTIAL AND GRADIENT	910
C		911
	COMMON/TT/MINT,X,Y,XDOT,YDOT,ZDOT,PHI,PHIR,PHIZ,DT	912
	DIMENSION ZI(11),RI(11),PHIC(IR,IZ)	913
	IF(INT.EQ.0) IG=0	914
	IF(INT.EQ.0) JG=0	915
	NCH=0	916
C	LOCATION OF Z IN ARRAY	917
C		918
C	ASSUME ZI(IZ) LESS THAN OR EQUAL TO Z LESS THAN OR EQUAL TO ZI(1).	919
C		920
	IF(Z.EQ.ZI(1)) IG=2	921
	IF(Z.EQ.ZI(1)) GO TO 103	922
	IF(INT.EQ.1) GO TO 100	923
C		924
	DO 10 I=2,IZ	925
	IG=IZ-I+2	926
	IF(Z.LT.ZI(IG-1)) GO TO 103	927
10	CONTINUE	928
	GO TO 999	929
C	RETURN WITHOUT LOCATING Z	930
C		931
C	ACCEPT IF ZI(IG) LESS THAN OR EQUAL TO Z LESS THAN ZI(IG-1).	933
C		934
100	IF(Z.GE.ZI(IG-1)) GO TO 102	935
C	IG TOO LARGE. USE DECREMENT LOOP	936
C		937
	IF(Z.GE.ZI(IG)) GO TO 104	938
101	IG=IG+1	939
	IF(Z.LT.ZI(IG)) GO TO 101	940
	GO TO 103	941
102	IG=IG-1	942
	IF(Z.GE.ZI(IG-1)) GO TO 102	943
103	NCH=1	944
C	IG DETERMINED	945
C		946
104	CONTINUE	947
C		948
C	LOCATION OF R IN ARRAY	949
C		950
C	ASSUME RI(1) LESS THAN OR EQUAL TO R LESS THAN OR EQUAL TO RI(IR).	951
C		952
	IF(R.EQ.RI(1)) JG=IR-1	953
	IF(R.EQ.RI(IR)) GO TO 153	954
	IF(INT.EQ.1) GO TO 150	955
C		956
	DO 15 J=2,IR	957
	JG=J-1	958
	IF(R.LT.RI(J)) GO TO 153	959

15	CONTINUE	959
	GO TO 999	960
C	RETURN WITHOUT LOCATING R	961
C		962
C	ACCEPT IF RI(JG) LESS THAN OR EQUAL TO R LESS THAN RI(JG+1)	964
		965
150	IF(R.GE.RI(JG+1)) GO TO 152	966
	IF(R.GE.RI(JG)) GO TO 154	967
151	JG=JG-1	968
	IF(R.LT.RI(JG)) GO TO 151	969
	GO TO 153	970
152	JG=JG+1	971
	IF(R.GE.RI(JG+1)) GO TO 152	972
153	NGH=1	973
154	CONTINUE	974
C		975
	IF(IG.LE.0.OR.JG.LE.0) GO TO 999	976
C		977
C	RETURN IF IG AND JG AND ALL THAT WERE REQUESTED	978
C		979
	IF(INT.EQ.2) GO TO 310	980
C		981
C	SET UP FRACTIONS	982
		983
	DELZ=ZI(IG-1)-ZI(IG)	984
	BELR=RI(JG+1)-RI(JG)	985
	FZ=(Z-ZI(IG))/DELZ	986
	FR=(R-RI(JG))/BELR	987
C		988
C	DEFINE POTENTIALS AT CORNERS OF BOX	989
C		990
	P22=PHI(IG(JG+1,IG-1))	991
	P21=PHI(IG(JG,IG-1))	992
	P12=PHI(IG(JG+1,IG))	993
	P11=PHI(IG(JG,IG))	994
	IF(NGH.EQ.0) GO TO 300	995
C		996
C	SKIP IF PHI-BOX IS NOT CHANGED	997
C		998
	D1=(P22-P12)/DELZ	999
	D2=(P21-P11)/DELZ	1000
	D3=(P22-P21)/BELR	1001
	D4=(P12-P11)/DELR	1002
C		1003
300	CONTINUE	1004
C		1005
C	INTERPOLATE TO FIND POTENTIAL AND COMPONENTS OF GRADIENT	1006
C		1007
	PHIZ=D2 + FR*(D1-D2)	1008
	PHIR=D4 + FZ*(D3-D4)	1009
	PHI=P11 + FR*(P12-P11) + FZ*(P21-P11) + FR*FZ*(P22-P21-P12+P11)	

310	CONTINUE	1010
	RETURN	1011
C		1012
C		1013
C	CANNOT LOCATE EITHER R OR Z OR BOTH	1014
C		1015
999	WRITE(MINT,9999) R,Z,IG,JG	1016
	CALL ABNOR(10H STOP22)	1017
	STOP22	1018
C		1019
9999	FORMAT(///1X,32HCANNOT LOCATE R OR Z. R,Z,IG,JG=,1P2E12.4,2I5)	1020
	END	1021

	SUBROUTINE TRACK(R,Z)	1022
C		1023
C	ADVANCE X,Y,Z,XDOT,YDOT,ZDOT BY CONSTANT ACCELERATION. WITH STEP C	1025
C	COMMON/TT/MINT,X,Y,XDOT,YDOT,ZDOT,PHI,PHIR,PHIZ,DT	1026
C		1027
	IF(R.EQ.0.) PHIX=0.	1028
	IF(R.EQ.0.) PHIY=0.	1029
	IF(R.EQ.0.) GO TO 100	1030
C		1031
	PHIX=PHIR*X/R	1032
	PHIY=PHIR*Y/R	1033
C		1034
100	CONTINUE	1035
C		1036
	X=X + DT*(XDOT - .25*DT*PHIX)	1037
	Y=Y + DT*(YDOT - .25*DT*PHIY)	1038
	Z=Z + DT*(ZDOT - .25*DT*PHIZ)	1039
C		1040
	XDOT=XDOT - .5*DT*PHIX	1041
	YDOT=YDOT - .5*DT*PHIY	1042
	ZDOT=ZDOT - .5*DT*PHIZ	1043
C		1044
	RETURN	1045
	END	1046

	FUNCTION ERF(X,ERROR)	1047
C		1048
C	ERROR FUNCTION	1049
C		1050
	DIMENSION A(7)	1051
	DATA A,RPI /1.,8.70923078E-1,.42282012E-1,.92789272E-2,	1052
	1 .15201430E-3,.27656720E-3,.430638E-4,1.772453E/	1053
	S=X*X	1054
	RS=ABS(X)	1055
	IF(S.LT.4.8) GO TO 240	1056
	S=AMIN1(S,675.)	1057
	ERROR=1./ (RS*RPI)*(1.+5/S*(-1.+5/S*(S.+7.5/S)))	1058
	ERF=1.-ERROR*EXP(-S)	1059
	RETURN	1060
C		1061
C	PASTINGS APPROXIMATION (P.187)	1062
C		1063
240	SP=1.0	1064
	SM=1.0	1065
	DO 300 I=2,7	1066
	SP=SP*RS	1067
	SM=SM+A(I)*SP	1068
	IF(ABS(SP).LT.1.E-9) GO TO 350	1069
300	CONTINUE	1070
350	ERFC=1./SM**16	1071
	ERF=1.-ERFC	1072
	RETURN	1073
	END	1074

	FUNCTION ERFINV(R)	1075
C		1076
C	INVERSE ERF RATIONAL APPROXIMATION	1077
C	(SEE ABRAMOWITZ & STEGUN, P. 933)	1078
C		1079
	DATA C0,C1,C2/2.515517,.002053,.010320/	1080
	DATA D1,D2,D3/1.432788,.089269,.001308/	1081
	DATA ROUND,ROOT2,ROOTPI/1.E-18,1.41421356237309,1.77245385/	
	ERFINV=-2.	1083
	IF(R.GT.1..OR.R.LT.0.) CALL ABNOR(10H STOP 33)	1084
	IF(R.GT.1..OR.R.LT.0.) STOP 33	1085
	X=0.	1086
	IF(R.LE..09) X=R*ROOTPI/2.	1087
	IF(R.LE..09) GO TO 200	1088
	P=(1.-R)/2.	1089
	IF(P.LE.0.) GO TO 200	1090
	T=0.	1091
	RADIC=0.	1092
	IF(P.LE.ROUND) GO TO 150	1093
	IF(P.GT.ROUND) RADIC=-2.*ALOG(P)	1094
	IF(RADIC.GT.0.) T=SQRT(RADIC)	1095
	XN=C0+T*(C1+T*C2)	1096
	XD=1.+T*(D1+T*(D2+T*D3))	1097
	XP=T	1098
	IF(XD.GT.0.) XP=XP-XN/XD	1099
	X=XP/ROOT2	1100
	IF(X.LT.ROUND) X=0.	1101
	GO TO 200	1102
150	CONTINUE	1103
	IF(P.GT.0.) T=ALOG(.5/ROOTPI/P)	1104
	IF(T.GT.0.) RADIC=T - .5*ALOG(T)	1105
	IF(RADIC.GT.0.) X=SQRT(RADIC)	1106
200	CONTINUE	1107
	ERFINV=X	1108
	END	1109

```

SUBROUTINE PFLOT(U,RI,ZI,NRI,NZI,NRO,NZO,RADIUS,XMAX,IT) 1110
DIMENSION ILINE(122),JLINE(122),BNDY(23),RI(11),ZI(11), 1111
1 REXP(122),ISYM(26),U(NRI,NZI) 1112
COMMON /IT/ MINT,X,Y,XDOT,YDOT,ZDOT,PHI,PHIR,FFIZ,DT 1113
DATA ISYM /1HZ,1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL, 1114
1 1HM,1HN,1HO,1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY/ 1115
DATA BNDY /100.,10.,9.,8.,7.,6.,5.,4.,3.,2.,1.,0.,-1.,-2.,-3., 1116
1 -4.,-5.,-6.,-7.,-8.,-9.,-10.,-100./ 1117
MINT=6 1118
C 1119
C***** 1120
C DETERMINE MINIMUM AND MAXIMUM OF THE X MATRIX 1121
C***** 1122
XMIN=10.**10 1123
XMAX=-XMIN 1124
NROM=NRO-1 1125
NZOM=NZO-1 1126
DO 100 JJ=1,NRI 1127
DO 100 II=1,NZI 1128
XMAX=AMAX1(XMAX,U(JJ,II)) 1129
XMIN=AMIN1(XMIN,U(JJ,II)) 1130
100 CONTINUE 1131
C 1132
C***** 1133
C LIST SYMBOL CORRESPONDENCE 1134
C***** 1135
WRITE(MINT,3001) XMAX,XMIN,IT 1136
3001 FORMAT(1H1,5X,*SYMBOL TABLE CORRESPONDENCE FOR PRINTED PLOT*) 1138
1 6X,*TWENTY TWO INTERVALS ASSUMED FOR DATA. XMAX = *, 1138
2 1PE12.4,* TO XMIN = *,E12.4,* AT BEGINNING OF CYCLE IT = *,
3 I6// 24X,*INTERVAL*/8X,*SYMBOL*,5X,*FROM*,12X,*TO*) 1140
WRITE(MINT,3002) (I,ISYM(I),BNDY(I),BNDY(I+1),I=1,22) 1141
3002 FORMAT(5X,I3,2X,A1,1P2E15.5) 1142
WRITE(MINT,3003) 1143
3003 FORMAT(1H1//) 1144
C 1145
C***** 1146
C EXPAND R COORD AND PREPARE TO INTERPOLATE 1147
C***** 1148
REXP(1)=RI(1) 1149
DRR=(RI(NRI)-RI(1))/FLOAT(NROM) 1150
DO 200 II=2,NRO 1151
REXP(II)=REXP(II-1)+DRR 1152
200 CONTINUE 1153
Y=0. 1154
DZZ=(ZI(1)-ZI(NZI))/FLOAT(NZOM)-1.E-10 1155
Z=ZI(1)+DZZ 1156
IZCUR=1 1157
DO 350 KK=1,NZO 1158
Z=Z-DZZ 1159
IFLG=0 1160

```

```

IF(Z.LE.ZI(IZCUR)) IFLG=1      1161
JRCUR=1                        1162
INT=0                          1163
DO 300 II=1,NRO                1164
X=R=REXP(II)                   1165
JFLG=0                          1166
IF(X.GE.RI(JRCUR)) JFLG=1      1167
CALL INTERP(R,Z,JG,IG,U,RI,ZI,NRI,NZI,INT) 1168
INT=1                          1169
C                               1170
C DETERMINE BOX NUMBER FROM BNDY MATRIX 1171
C                               1172
DO 250 JJ=1,22                 1173
IBOX=JJ                        1174
TBOX=((BNDY(JJ)-PHI)*(PHI-BNDY(JJ+1))) 1175
IF(IBOX.EQ.22) ILINE(II)=ISYM(22) 1176
IF(TBOX.GT.0.) GO TO 750      1177
250 CONTINUE                   1178
750 CONTINUE                   1179
   ILINE(II)=1H               1180
   IF(MOD(IBOX,2).EQ.1) ILINE(II)=ISYM(MOD(
IBOX-1,26)+1)                 1181
C                               1182
C MARK CORNERS                 1183
C                               1184
   IF(IFLG.EQ.1.AND.JFLG.EQ.1) ILINE(II)=1H+ 1185
C                               1186
C MARK DISC                    1187
C                               1188
   IF(ZI(IZCUR).EQ.0..A.REXP(II).LE.RADIUS.A.IFLG.EQ.1) ILINE(II)=1H- 1189
   IF(JFLG.EQ.1) JRCUR=JRCUR+1 1190
C                               1191
300 CONTINUE                   1192
C                               1193
C *****                    1194
C HAVE WE CROSSED A Z BOUNDARY????? 1195
C *****                    1196
   IF(IFLG.EQ.1) GO TO 310     1197
C                               1198
C ELSE                          1199
   WRITE(MINT,3004) (ILINE(I),I=1,NRO) 1200
3004 FORMAT(9X,12Z A1)         1201
   GO TO 350                   1202
C                               1203
310 CONTINUE                   1204
   WRITE(MINT,3005) ZI(IZCUR), (ILINE(I),I=1,NRO) 1205
3005 FORMAT(1X,F7.2,1X,80A1)   1206
   IZCUR=IZCUR+1              1207
350 CONTINUE                   1208
   RETURN                      1209
   END                        1210

```

	SUBROUTINE DENSTY(INIT)	1211
C		1212
C	COMPUTE FLUXES AND CHARGE DENSITIES	1213
C		1214
	COMMON M, IV, IFIRST, JFIRST, JEDGE, NR, NZA, NZ, NTOT, PI, IT, IR, IZ,	
	1 NOKMAX, TIME, IPRINT, RADIUS, RR(11), ZZ(11), U(91), Q(91),	1216
	2 AREA(11), DELZ(11), RI(11), ZCI(11)	1217
	COMMON/DEN/RPT, ZPT, AL1, BE1, EV, TVOLTS, DENST, NPTS, SPEED0, CURR,	
	1 IPART, PARTCL(2), DELTA, SPEED, QDISK, OZMIN, OTSEC, NINJ,	1219
	2 VSAVE(10, 100), TVCHG, NRAN, NCESC, NCABS, NCACT, NCHGE(11), NCHGI(11),	
	3 KOUNT(51), XKOUN(51), MODE, WCLOUD	1221
	COMMON/TT/HINT, X, Y, XDOT, YDOT, ZDOT, PHI, PHIR, PHIZ, DT	1222
	01MENSION ENDESC(2), ENCABS(2), PHIC(51), WT(51), FHYP(51), PHYZ(51),	
	1 PHYALT(11), IPDSK(11)	1224
	DATA ENDESC/5H ESCA, 5HPES /, ENCABS/5H AESOR, 5HED /	1225
	NIJ(I, J) = NR * (I - 1) * J	1226
	IF(IPRINT.GT.0) WRITE(M, 1000) IT, PARTCL, SPEED0, CURR	1227
		1228
	JV=3-IV	1229
	IF(IFIRST.NE.0) GO TO 90	1230
	CALL RANSET(NRAN)	1231
	NPARTS=0	1232
	HITE=ZZ(1)-ZZ(NZ)	1233
	AREAA=PI/2.*RR(NR)*RR(NR)	1234
	AREAB=RR(NR)*HITE	1235
	ROOTPI=SQRT(PI)	1236
	NTOP=0	1237
	NBOT=0	1238
	NSID=0	1239
	IC=1	1240
	JG=1	1241
	HINT=M	1242
	N1=1	1243
	INJB=0	1244
	SLK=1H	1245
	REWIND IV	1246
	REWIND JV	1247
		1248
C	IF(IT.NE.0) GO TO 60	1249
		1250
C	DO 80 I=1, JEDGE	1251
	NCHGE(I)=0	1252
	NCHGI(I)=0	1253
80	CONTINUE	1254
	GO TO 90	1255
60	CONTINUE	1256
	READ(IV) INJA, ((VSAVE(I, J), I=1, 10), J=1, INJA)	1257
	IF(TEOP(IV)) 90, 70	1258
70	CONTINUE	1259
	WRITE(JV) INJA, ((VSAVE(I, J), I=1, 10), J=1, INJA)	1260
	NPARTS=VSAVE(9, INJA)+.1	1261

	GO TO 60	1262
C		1263
90	CONTINUE	1264
	IF (INIT.EQ.1) GO TO 350	1265
	NCABS=0	1266
	NCESC=0	1267
	NCACT=0	1268
	INTER=0	1269
	N=0	1270
	IF (IPRINT.GT.0.AND.IPART.EQ.2) WRITE(M,1010) (RR(J),J=1,NF)	
	IF (IPRINT.GT.0.AND.IPART.EQ.1) WRITE(M,1020) (RR(J),J=1,NF)	
	J1=1-NR	1273
	J2=8	1274
	DO 110 I=1,NZA	1275
	J1=J1+NR	1276
	J2=J2+NR	1277
	DO 100 J=1,NF	1278
	N=N+1	1279
	IF (TVOLTS.EQ.0) GO TO 110	1280
100	PHIC(N)=U(N)/TVOLTS	1281
	IF (IPRINT.GT.0) WRITE(M,1100) I,ZZ(I),(PHIC(J),J=J1,J2)	1282
110	CONTINUE	1283
C		1284
	IF (IPRINT.GT.0) WRITE(M,1200)	1285
	I1=NZA+1	1286
	DO 160 I=I1,NZ	1287
	J1=J1+NR	1288
	J2=J2+NR	1289
	DO 150 J=1,NF	1290
	N=N+1	1291
	IF (TVOLTS.EQ.0) GO TO 160	1292
150	PHIC(N)=U(N)/TVOLTS	1293
	IF (IPRINT.GT.0) WRITE(M,1100) I,ZZ(I),(PHIC(J),J=J1,J2)	1294
160	CONTINUE	1295
	IF (MODE.NE.1) GO TO 180	1296
C		1297
C	CALCULATE PHYR AND PHYZ MATRICES	1298
C		1299
	N=0	1300
	DO 170 I=1,NZ	1301
	IP=MING(I+1,NZ)	1302
	IM=MAX8(I-1,1)	1303
C		1304
	DO 170 J=1,NF	1305
	N=N+1	1306
	JP=MING(J+1,NR)	1307
	JM=MAX8(J-1,1)	1308
C		1309
	NP=NIJ(I,JP)	1310
	NM=NIJ(I,JM)	1311
	PHYR(N)=(PHIC(NP)-PHIC(NM))/(RR(JP)-RR(JM))	1312

```

IF(I.EQ.NZA.AND.J.EQ.JEDGE+1) PHYR(N)=(PHIC(NP)-PHIC(NP))/
I (ERR(JP)-RADIUS) 1314
C 1315
NH=NIJ(IN,J) 1316
NP=NIJ(IP,J) 1317
PHYZ(N)=(PHIC(NH)-PHIC(NP))/(ZZ(IN)-ZZ(IP)) 1318
IF(I.EQ.NZA.AND.J.LE.JEDGE) PHYZ(N)=(PHIC(NH)-PHIC(NP))/
I (ZZ(IN)-ZZ(IP)) 1319
IF(I.EQ.NZA.AND.J.LE.JEDGE) PHYALT(J)=(PHIC(N)-PHIC(NP))/
I (ZZ(I)-ZZ(IP)) 1322
170 CONTINUE 1323
180 CONTINUE 1324
C 1325
IF(NPTS.NE.0) GO TO 350 1326
C 1327
C SINGLE TRAJECTORY (NPTS=0) 1328
C 1329
ALPHA=AL1*PI/180. 1330
BETA=BE1*PI/180. 1331
WRITE(M,1500) AL1,BE1,ALPHA,BETA,EV 1332
CS1 GO TO 400 1333
C**** 1334
C 17 AUG 78. NOTE THAT SINGLE TRAJECTORY IS INCOMPLETE 1335
C BEGINNING HERE. TEMPORARILY THIS CASE IS JUST TERMINATED.
C**** 1337
CALL AENDR(IGH STOP400) 1338
STOP 400 1339
C INITIAL CONDITIONS OF TRAJECTORIES 1340
C 1341
350 CONTINUE 1342
C**** 1343
D INJECT NEW PARTICLES 1344
C ASSUMED FLUX = NINJ IONS PER CYCLE 1345
C**** 1346
C 1347
DO 370 I=1,NINJ 1348
RAN1=RANF(NRAN) 1349
RAN2=RANF(NRAN) 1350
RAN3=RANF(NRAN) 1351
RAN4=RANF(NRAN) 1352
RAN5=RANF(NRAN) 1353
RAN6=RANF(NRAN) 1354
NPARTS=NPARTS+1 1355
INJB=INJB+1 1356
C 1357
C SELECT VERTICAL AND PERPENDICULAR COMPONENTS OF VELOCITY 1358
C 1359
VPERP=SQRT(ABS(ALOG(RAN1))) 1360
C 1361
VVERT=ERFINV(ABS(1.-2.*RAN2)) 1362
IF(RAN2.GT..5) VVERT=-VVERT 1363

```

C		1364
C	CALCULATE THETA AND TRIG FUNCTIONS OF THETA	1365
C		1366
	AL1R=ATAN2(VPERP,VVERT)	1367
	CAL1=COS(AL1R)	1368
	SAL1=SIN(AL1R)	1369
C		1370
C	SELECT TOP/BOTTOM OR SIDE	1371
C		1372
	RAREAS=AREA*AOS(CAL1)/(AREA*SAL1+AREA*AOS(CAL1))	1373
	IF(RAN3.LE.RAREAS) GO TO 360	1374
C		1375
C	PARTICLE HITS SIDE. CALCULATE X,Y,Z AND PHI COORD.	1376
C		1377
	LOCAT=4HSIDE	1378
	S9E1=RAN4	1379
	9E1R=ASIN(S9E1)	1380
	C9E1=COS(9E1R)	1381
	NSID=NSID+1	1382
C		1383
	X=RR(NR)	1384
	IF(INIT.EQ.1) X=SQRT(RAN6)*RR(NR)	1385
	Y=0.	1386
	Z=RAN5*(HITE)+ZZ(NZ)	1387
C		1388
	GO TO 410	1389
360	CONTINUE	1390
C		1391
C	PARTICLE HITS TOP/BOTTOM. CALCULATE X,Y,Z AND PHI COORD.	1392
C		1393
	9E1R=2.*PI*RAN4	1394
	C9E1=COS(9E1R)	1395
	S9E1=SIN(9E1R)	1396
C		1397
	X=SQRT(RAN5)*RR(NR)	1398
	Y=0.	1399
	Z=ZZ(NZ)	1400
	IF(CAL1.LT.0.) Z=ZZ(1)	1401
	IF(INIT.EQ.1) Z=HITE*RAN6+ZZ(NZ)	1402
	LOCAT=6HBOTTOM	1403
	IF(CAL1.LT.0.) LOCAT=3HTOP	1404
	IF(CAL1.GE.0.) N9OT=N9OT+1	1405
	IF(CAL1.LT.0.) NTOP=NTOP+1	1406
410	CONTINUE	1407
C		1408
C	CALCULATE VELOCITY COMPONENTS	1409
C		1410
	XDOT=SPEED0*VPERP*C9E1	1411
	YDOT=SPEED0*VPERP*S9E1	1412
	ZDOT=SPEED0*VVERT	1413
C		1414

VSAVE(1, INJB)=X	1415
VSAVE(2, INJB)=Y	1416
VSAVE(3, INJB)=Z	1417
VSAVE(4, INJB)=XDOT	1418
VSAVE(5, INJB)=YDOT	1419
VSAVE(6, INJB)=ZDOT	1420
VSAVE(7, INJB)=DELTA*RANG	1421
IF(INIT.EQ.1) VSAVE(7, INJB)=DELTA	1422
IF(IPART.EQ.1) VSAVE(8, INJB)=1.	1423
IF(IPART.EQ.2) VSAVE(8, INJB)=-1.	1424
VSAVE(9, INJB)=FLOAT(NPARTS)+.1	1425
VSAVE(10, INJB)=1.	1426
AL1=AL1R*180./PI	1427
BEI=BEIR*180./PI	1428
C	1429
IF(INJB.LT.100) GO TO 370	1430
WRITE(JV) INJB, ((VSAVE(K, J), K=1, 10), J=1, INJB)	1431
INJB=0	1432
C	1433
370 CONTINUE	1434
CSS WRITE(M, *) = ATOBTOC = ", NTOP, NSID, NBOT	1435
IF(INJB.GT.0) WRITE(JV) INJB, ((VSAVE(I, J),	1436
1 I=1, 10), J=1, INJB)	1437
INJB=0	1438
IF(INIT.EQ.1) RETURN	1439
JV=3-JV	1440
IV=3-JV	1441
C	1442
IF(IPRINT.GT.8) WRITE(M, 3410) IT, TIME, PARTCL	1443
IF(IPRINT.GT.3) WRITE(M, 3415) DELTA, DTSEC	1444
C	1445
GO 820 I=1, NTOP	1446
KOUNT(I)=0	1447
XKOUNT(I)=0.	1448
820 CONTINUE	1449
IOPAR=IME	1450
IF(IPART.EQ.1) IOPAR=1HI	1451
REWIND JV	1452
REWIND IV	1453
890 CONTINUE	1454
READ(IV) INJA, ((VSAVE(I, J), I=1, 10), J=1, INJA)	1455
IF(2OP(IV)) 860, 870	1456
870 CONTINUE	1457
DO 910 INJ=1, INJA	1458
NPART=VSAVE(9, INJ)+.1	1459
IF(IPART.EQ.1.ANC.VSAVE(8, INJ).LT.0.) GO TO 860	1460
IF(IPART.EQ.2.ANC.VSAVE(8, INJ).GT.0.) GO TO 860	1461
NCACT=NCACT+1	1462
C	1463
C NEW = 1 IMPLIES NEWLY INJECTED PARTICLE	1464
C FAYT = 0., -1., -2., IMPLIES NORMAL, ESCAPING, OR ABSORBED	

```

C
NEW=0 1466
FAYT=0. 1467
FATE=1H 1468
ZOLD=VSAVE(3,INJ) 1469
X=VSAVE(1,INJ) 1470
Y=VSAVE(2,INJ) 1471
Z=VSAVE(3,INJ) 1472
R=SGRT(X*X+Y*Y) 1473
XDOT=VSAVE(4,INJ) 1474
YDOT=VSAVE(5,INJ) 1475
ZDOT=VSAVE(6,INJ) 1476
OT=VSAVE(7,INJ)*CZMIN 1477
INT=0 1478
IF(INTER.EQ.0) CALL INTERP(R,Z,JG,IG,PHIC,RR,ZZ,NR,NZ,INT)
IF(INTER.GT.0) CALL INTERP(R,Z,JG,IG,PHIC,RI,ZCI,IR,I?,INT)
E=PHI+SPEED*SPEED 1482
IF(VSAVE(7,INJ).LT.DELTA) NEW=1 1483
VEL=SQRT(XDOT*XDOT+YDOT*YDOT+ZDOT*ZDOT) 1484
IF(IPRINT.GT.3.AND.NEW.EQ.1) WRITE(M,2100) NPART,IDPAR,IG,JG,
1 N1,U(N1),X,Y,Z,R,XDOT,YDOT,ZDOT,FATE,VEL 1486
IF(VSAVE(7,INJ).LT.DELTA) VEL=VSAVE(7,INJ) 1487
C 1488
VELSQ=1. 1489
IF(NEW.EQ.0) GO TO 450 1490
PHIA=PHI*SPEED**2 1491
IF(X.EQ.RR(NF)) GO TO 420 1492
VELSQ=ZDOT**2-PHIA 1493
IF(VELSQ.GT.0.) ZDOT=SIGN(SQRT(VELSQ),ZDOT) 1494
IF(VELSQ.LE.0.) ZDOT=-ZDOT 1495
GO TO 430 1496
420 CONTINUE 1497
VELSQ=XDOT**2-PHIA 1498
IF(VELSQ.GT.0.) XDOT=SIGN(SQRT(VELSQ),XDOT) 1499
IF(VELSQ.LE.0.) XDOT=-XDOT 1500
430 CONTINUE 1501
IF(IPRINT.GT.3) WRITE(M,2100) NPART,IDPAR,IG,JG,
1 N1,U(N1),X,Y,Z,R,XDOT,YDOT,ZDOT,FATE,VEL 1503
IF(VELSQ.LE.0.) GO TO 500 1504
450 CONTINUE 1505
IF(MODE.NE.1) GO TO 440 1506
B 1507
C DISTRIBUTE FORCES INTO PHIR AND PHIZ 1508
G 1509
CALL CLOUD(R,Z,PHIC,IFDSK,WT,WCL OUD) 1510
PHIR=0. 1511
PHIZ=0. 1512
N=0 1513
DO 460 I=1,NZ 1514
DO 460 J=1,NF 1515
N=N+1 1516

```

	PHIR=PHIR+WT(N)*PHYR(N)	1517
C		1518
C	SIFT FOR SPECIAL HALF BOX MARKED BY IFDSK = -1	1519
C		1520
	IF(I.NE.NZA.CR.J.GT.JEDGE) GO TO 470	1521
	IF(IFDSK(J).NE.-1) GO TO 470	1522
C		1523
	PHIZ=PHIZ+WT(N)*PHYALT(J)	1524
	GO TO 460	1525
C		1526
470	CONTINUE	1527
	PHIZ=PHIZ+WT(N)*PHYZ(N)	1528
460	CONTINUE	1529
C		1530
440	CONTINUE	1531
	IF(MODE.NE.2) GO TO 465	1532
C		1533
C	CALCULATE WEIGHTS	1534
C		1535
	IF(MEN.NE.1.CR.PHI.GE.0.OR.IT.EQ.0) GO TO 465	1536
	PHY=-PHI*TVOLTS/TVCHG	1537
	XPHY=SQRT(PHY)	1538
	ERFC=1.-ERF(XPHY,DENA)	1539
	IF(XPHY.LT.2.) DENA=EXP(PHY)*ERFC	1540
	IF(MONMAX.LE.1) VSAVE(10,INJ)=1./(1.-SQRT(PHY/(1.+PHY)))	1541
	IF(MONMAX.EQ.2) VSAVE(10,INJ)=2./ROOTPI*XPHY*DENA	1542
	VSAVE(10,INJ)=VSAVE(10,INJ)**.25	1543
		1544
465	CONTINUE	1545
C		1546
	XDOT=XDOT/SPEED	1547
	YDOT=YDOT/SPEED	1548
	ZDOT=ZDOT/SPEED	1549
C		1550
	CALL TRACK(R,Z)	1551
C		1552
	XDOT=XDOT*SPEED	1553
	YDOT=YDOT*SPEED	1554
	ZDOT=ZDOT*SPEED	1555
	VSAVE(1,INJ)=X	1556
	VSAVE(2,INJ)=Y	1557
	VSAVE(3,INJ)=Z	1558
	VSAVE(4,INJ)=XDOT	1559
	VSAVE(5,INJ)=YDOT	1560
	VSAVE(6,INJ)=ZDOT	1561
	VSAVE(7,INJ)=DELTA	1562
	R=SQRT(X*X+Y*Y)	1563
	IF(R.GT.0.) ROOT=(X*XDOT+Y*YDOT)/R	1564
	IF(R.EQ.0.) ROOT=SQRT(XDOT*XDOT+YDOT*YDOT)	1565
C		1566
C	CHECK WHETHER TRAJECTORY IS TERMINATED	1567

C		1568
	IF(R.GT.RR(N9).OR.Z.GT.ZZ(1)) GO TO 500	1569
	IF(Z.LT.ZZ(NZ)) GO TO 500	1570
C		1571
	IF(Z*ZOLD.LT.0..AND.R.LT.RADIUS) GO TO 550	1572
C		1573
	IF(MODE.NE.1) GO TO 900	1574
C		1575
C	GENERATE CLOUD AND DISTRIBUTE CHARGE	1576
	CALL CLOUD(R,Z,PHIC,IFDSK,NT,WGLOUD)	1577
	DO 480 I=1,NTOT	1578
	XKOUN(I)=XKOUN(I)+WT(I)	1579
480	CONTINUE	1580
C		1581
	GO TO 900	1582
C		1583
C	PARTICLE ESCAPES	1584
C		1585
500	CONTINUE	1586
C		1587
	NCESC=NCESC+1	1588
	NCACT=NCACT-1	1589
	FATE=1HE	1590
	FAYT=1.	1591
C		1592
	GO TO 900	1593
C		1594
C	PARTICLE IS ABSORBED	1595
C		1596
550	CONTINUE	1597
C		1598
	NGABS=NGABS+1	1599
	NCACT=NCACT-1	1600
	FATE=1HA	1601
	FAYT=-2.	1602
C		1603
C		1604
900	CONTINUE	1605
C		1606
C	INTERPOLATE TO FIND IG,JG	1607
	INT=2	1608
	IF(FAYT.NE.-1.) CALL INTERP(R,Z,JG,IG,PHIC,RI,ZGI,IR,I7,INT)	1609
	N1=JG+NR*(IG-2)	1610
C		1611
C	DO SUMS FOR SUMMARY	1612
C		1613
	IF(FAYT.GE.0.) KOUNT(N1)=KOUNT(N1)+1	1614
	IF(FAYT.GE.0..AND.MODE.NE.1) XKOUN(N1)=XKOUN(N1)+VSAVE(10,INJ)	1615
	IF(IPART.EQ.2.AND.FAYT.EQ.-2.) NCHGE(JG)=NCHGE(JG)+1	1616
	IF(IPART.EQ.1.AND.FAYT.EQ.-2.) NCHGI(JG)=NCHGI(JG)+1	1617
C		1618

```

C      WRITE SUMMARY FOR CURRENTLY ACTIVE PARTICLE                1619
C      NEED TO ADD ROOT TO LISTINGS LATER                          1620
C                                                                    1621
      IF(IFPRINT.GT.3.AND.NEW.EQ.0) WRITE(M,2100) NPART,IOPAR,    1622
1  IG,JG,N1,U(N1),X,Y,Z,R,XDOT,YDOT,ZDOT,FATE,VEL                1623
      IF(IFPRINT.GT.3.AND.NEW.EQ.1) WRITE(M,2200) NPART,IOPAR,    1624
1  IG,JG,N1,U(N1),X,Y,Z,R,XDOT,YDOT,ZDOT,FATE,VEL                1625
C                                                                    1626
C      SAVE CURRENTLY ACTIVE PARTICLES                              1627
C                                                                    1628
      IF(FAYT.LT.0.) GO TO 910                                     1629
860  CONTINUE                                                       1630
      INJB=INJB+1                                                 1631
      IF(INJ.EQ.INJB) GO TO 910                                     1632
      DO 915 I=1,9                                                 1633
      VSAVEI,INJBT=VSAVE(I,INJ)                                    1634
915  CONTINUE                                                       1635
C                                                                    1636
910  CONTINUE                                                       1637
C                                                                    1638
      IF(INJB.GT.0) WRITE(JV) INJB,((VSAVE(I,J),I=1,10),J=1,INJB)
      INJB=0                                                       1640
      GO TO 890                                                    1641
C                                                                    1642
880  CONTINUE                                                       1643
C                                                                    1644
      IF(IPRINT.EQ.8) RETURN                                       1645
C                                                                    1646
      WRITE(M,3420) PARTCL,(BLK,I,I=1,JEDGE)                       1647
      IF(IPART.EQ.2) WRITE(M,3430) NZA,(INCHGE(I),                 1648
1  I=1,JEDGE)                                                       1649
      IF(IPART.EQ.1) WRITE(M,3430) NZA,(INCHGI(I),                 1650
1  I=1,JEDGE)                                                       1651
C                                                                    1652
      N1=0                                                         1653
      N2=0                                                         1654
      WRITE(M,3440) PARTCL,(BLK,I,I=1,NR)                          1655
      DO 920 I=1,NZ                                               1656
      N1=N2+1                                                       1657
      N2=N2+NR                                                      1658
      WRITE(M,3430) I,(KOUNT(J),J=N1,N2)                           1659
920  CONTINUE                                                       1660
C                                                                    1661
      RETURN                                                       1662
C                                                                    1663
1000 FORMAT(1H1/21H FLUXES AND DENSITIES,4X,4HIT =,13,4X,2A5,    1664
1  13H WITH SPEED =,1PE15.4,7H CM/SEC,4X,22H AND CURRENT DENSITY =,
2  E19.4,14H PICCA/P/CM**2)                                       1666
1010  FORMAT(/5X,41HDIMENSIONLESS POTENTIAL ARRAY - ABOVE Z=0   1667
1  5X,32H(IN UNITS OF MINUS TVIONS/XMASS)                          1668
2  //1X,3HR =,14F9.3/ (/4X,14F9.3))                                1669

```

```

1020 FORMAT(/5X,41HDIMENSIONLESS POTENTIAL ARRAY - ABOVE Z=0      1670
1 5X,20H(IN UNITS OF TVIONS)                                       1671
2 //1),3HR =,14F9.3/ (/4X,14F9.3))                                  1672
1100 FORMAT(/5H LINE,14,5X,2HZ=,F7.3/(/7F16.8))                    1673
1200 FORMAT(///1X, 41HDIMENSIONLESS POTENTIAL ARRAY - BELOW Z=0 )
1300 FORMAT(1X,17HSINGLE TRAJECTORY/1X,23ANGLES ALPHA AND BETA =,
12F15.8,11H DEGREES =,2F15.8,8H RADIANS/9H ENERGY =,F15.8,5HVOLTS)
2000 FORMAT(1X,I5,1P11E10.3,2I5)                                     1677
2100 FORMAT(1X,I5,1H-,A1,3I3,1P8E11.3,2X,A1,E11.3)                 1678
2200 FORMAT(1X,I5,1H-,A1,3I3,1P8E11.3,2X,A1,6H RAN=,E10.3)       1679
2700 FCRMAT(/1X,2A6,5X,16HENERGY CHANGES =, 1P2E10.2,
1 20H (WHOLE CROIT, MAX PER STEP) )                                1681
3400 FORMAT(/5X,33HABSORPTION SUMMARY FOR IT CYCLE =,I5,          1682
1 2X,9HAT TIME =,1PE10.3,2X,2A5,2X,5X,23HTOTAL NUMBER ABSORBED =,
2 I7/8X,27HNUMFER ABSORBED THIS STEP =,I5,8X,15HNUMBER ESCAPING,
3 12H THIS STEP =,I5,8X,25HNUMBER CURRENTLY ACTIVE =,I5)         1685
3410 FORMAT(1H1//,5X,38HPOPULATION SUMMARY FOR ITERATION IT =,
1 I5,5X,9HAT TIME =,1PE10.3,2X,2A5)                                1687
3415 FORMAT(5X,6HDELTA=,1PE11.3,5X,11H(MEANS DT =,E11.3,5H SEC)
1 //10X,7HI J N,2X,9HPOTENTIAL,6X,1HX,10X,1HY,10X,1HZ,10X,1HR,
2 8X,42HXDOT YDOT ZDOT VELOCITY, 1690
3 /20X,5HVOLTS,9X,2HCM,9X,2HCM,9X,2HCM,9X,2HCM,7X,6HCM/SEC,
4 6X,6HCM/SEC,4X,6HCM/SEC,6X,6HCM/SEC)                             1692
3420 FORMAT(/5X,31HSUMMARY OF ABSORBED CHARGES FOR,2X,2A5//12X,9(A2,
1 2HR(,I1,1H)),11(A1,2HR(,I2,1H)))                                1694
3430 FORMAT(/5X,2HZ(,I2,1H),20I6)                                  1695
3440 FORMAT(/5X,34HSUMMARY OF SPACE CHARGE MATRIX FOR,2X,2A5//12X,9(A2
1 ,2HR(,I1,1H)),11(A1,2HR(,I2,1H)))                                1697
END                                                                    1698

```

```

SUBROUTINE CLOUD(RCD,ZCD,PHIC,IFDSK,WT,WCLOUD) 1699
C***** 1701
C DISTRIBUTION OF FINITE-CLOUD PARTICLE AMONG GRID BOXES 1702
C (CHARGES AND FORCES) 1703
C 1704
C VERSION OF 15 JAN 79. LAST ALTERATION 2 MAR 79..... 1705
C***** 1706
C***** 1706
DIMENSION PHIC(51),WT(51),IFDSK(11) 1708
COMMON N,IV,IFIRST,JFIRST,JEDGE,NR,NZA,NZ,NTOT,PI,IT,IR,IZ,
1 MONMAX,TIME,IPRINT,RADIUS,RR(11),ZZ(11),U(51),Q(51),
2 AREAH(11),DELZ(11),RI(11),ZCI(11) 1710
NIJ(I,J)=NR*(I-2)+J 1711
NZAF=NZA+1 1712
C 1713
C 1714
C DETERMINE R AND Z COORDINATES OF CORNERS OF CLOUD 1715
C 1716
ZH=AMIN1(ZCD+WCLOUD,ZCI(11)) 1717
ZL=AMAX1(ZCD-WCLOUD,ZCI(12)) 1718
RH=AMIN1(RCD+WCLOUD,RI(IR)) 1719
RL=AMAX1(RCD-WCLOUD,0.) 1720
C 1721
C DETERMINE IG AND JG CORNERS OF GRID CONTAINED WITHIN CLOUD 1722
C 1723
CALL INTERP(RL,ZL,JGL,IGL,PHIC,RI,ZCI,IR,IZ,2) 1724
CALL INTERP(RH,ZH,JGH,IGH,PHIC,RI,ZCI,IR,IZ,2) 1725
C 1726
C DETERMINE VOLUME WEIGHTS OF EACH IG JG BOX CONTAINED IN CLOUD
C 1726
DO 80 II=1,NTOT 1729
WT(II)=0. 1730
80 CONTINUE 1731
C 1732
DO 90 II=1,NR 1733
IFDSK(II)=0 1734
90 CONTINUE 1735
C 1736
DO 10 I=IGH,IGL 1737
DO 20 J=JGL,JGH 1738
N1=NIJ(I,J) 1739
W2=AREAH(J) 1740
IF(I.LE.NZA) W1=DELZ(I-1) 1741
IF(I.EQ.NZAF) W1=DELZ(NZA)+DELZ(NZAF) 1742
IF(I.GT.NZAF) W1=DELZ(I) 1743
IF(I.EQ.IGH) W1=ZH-ZCI(IGH) 1744
IF(I.EQ.IGL) W1=ZCI(IGL-1)-ZL 1745
IF(IGH.EQ.IGL) W1=2.*WCLOUD 1746
C 1747
IF(J.EQ.JGL) W2=PI*(RI(JGL+1)**2-RL**2) 1748
IF(J.EQ.JGH) W2=PI*(RH**2-RI(JGH)**2) 1749

```

	IF(JGH.EQ.JGL) W2=2.*WCLOUD	1750
C		1751
	IF(ZL.GT.0..OR.ZH.LT.0..OR.J.GT.JEDGE) GO TO 3C	1752
C		1753
	IF(ZCD.LT.0..AND.I.LE.NZAP) GO TO 4B	1754
	IF(ZCD.GE.0..AND.I.GE.NZAP) GO TO 5B	1755
	GO TO 30	1756
C		1757
C	BOX IS IN SHADOW	1758
C		1759
50	CONTINUE	1760
C	BELOW DISC	1761
	W1=0.	1762
	IF(I.EQ.NZAP) W1=AMINI(DELZ(NZAP),ZH)	1763
	GO TO 30	1764
C		1765
40	CONTINUE	1766
C	ABOVE DISC	1767
	W1=0.	1768
	IF(I.NE.NZAP) GO TO 30	1769
	IFDSK(J)=-1	1770
	W1=AMINI(DELZ(NZAP),-ZL)	1771
C		1772
30	CONTINUE	1773
	IF(I.EQ.NZAP.AND.ZH.LT.0.) IFDSK(J)=-1	1774
	WT(N1)=W1*W2	1775
20	CONTINUE	1776
10	CONTINUE	1777
C		1778
C	COMPLETE WEIGHTS	1779
C		1780
	MVOL=0.	1781
	DO 60 I=1,NTCT	1782
	MVOL=MVOL+WT(I)	1783
60	CONTINUE	1784
	DO 70 I=1,NTCT	1785
	WT(I)=WT(I)/MVOL	1786
70	CONTINUE	1787
	RETURN	1788
	END	1789

```

SUBROUTINE AENOR(ERR)                                1790
C                                                     1791
C*****                                             1792
C                                                     1793
C  SUBROUTINE ABNOR IS A DUMMY ROUTINE TO FORCE A MODE 2 EXIT 1794
C  WHEN ANY NON STANDARD STOP CONDITION IS DETECTED      1795
C                                                         1796
C*****                                             1797
C                                                     1798
C  PRINT 3001,ERR                                       1799
3001 FORMAT(1X,'*AENORMAL STOP REQUESTED*,5X,A10)    1800
A=0.                                                  1801
C=9/A                                               1802
RETURN                                             1803
END                                               1804
*EOR                                             1805
MAXWELLIAN ISOTROPIC CASE  MARCH 1979           1806
  2   5   5   5   0   1                               1807
0.    40.   60.   80.   100.                       1808
50.   25.   0.                                     1809
1.    -25.  -5%.                                    1810
-10.  -10.                                       1811
  0   1   0   40  20   3  20  25                   1812
.07803 0.    .1    .1    75000.   15.             1813
*EOR                                             1814
MAXWELLIAN ISOTROPIC CASE  OCTOBER 1978        1815
 24  50  26  26  0  0                               1816
0.    2.    6.    9.    12.   14.   16.   18.   19.   1817
20.   22.   24.   26.   28.   30.   32.   34.   34.   1818
36.   38.   40.   42.   44.   46.   48.   49.   49.   1819
50.5  52.   54.   56.   58.   60.   62.   64.   64.   1820
66.   68.   70.   72.   74.   76.   78.   80.   80.   1821
82.   84.   86.   88.   90.   92.   94.   96.   96.   1822
98.   100.                                       1823
50.   48.   46.   44.   42.   40.   38.   36.   36.   1824
34.   32.   30.   28.   26.   24.   22.   20.   20.   1825
18.   16.   14.   12.   10.   8.    6.    4.    4.   1826
2.    0.                                       1827
0.    -2.   -4.   -6.   -8.   -10.  -12.  -1.   -1.   1828
-16.  -18.  -20.  -22.  -24.  -26.  -28.  -3.   -3.   1829
-32.  -34.  -36.  -38.  -40.  -42.  -44.  -4.   -4.   1830
-48.  -50.                                       1831
-10.  -10.  -10.  -10.  -10.  -10.  -10.  -1.   -1.   1832
-10.  -10.  -10.  -10.  -10.  -10.  -10.  -1.   -1.   1833
-10.  -10.  -10.  -10.  -10.  -10.  -10.  -1.   -1.   1834
  0   1   0   10  20   5  20  25                   1835
.1    0.    .1    .1    75000.   15.             1836

```

```

PROGRAM SUMRTE(TAPE1, INPUT, OUTPUT, TAPE2, TAPE6=CUTPUT, TAPE5=INPLT)
DIMENSION
1 KHABSI(12), KHACTION(12), MESSG(5), OSET(4),
2 KHABSE(12), KHACTE(12), LINE(100), PVEC(4), IPSYM(4), PSCL(4)
NAMelist /CONTL/ DENCC, APN, AMX, BMN, BMX, CMN, CMX, DMN, DMX
C
C DEFAULT CONTL ASSIGNMENTS
C
DENCC=10000.
AMN=0.
AMX=30.
BMN=2500.
BMX=5000.
CMN=2500.
CMX=5000.
DMN=-1.
DMX=4.
C
IPSYM(1)=1HA
IPSYM(2)=1HB
IPSYM(3)=1HC
IPSYM(4)=1HD
C
C READ CONTL PARAMETERS
C
CALL CONNEC(5)
CALL CONNEC(6)
WRITE(6, 2002)
2002 FORMAT(1X, *ENTER DENCC AND RANGES OF PRINTER PLOT VARIABLES*,
* IN NAMELIST FORMAT*/)
READ(5, CONTL)
WRITE(6, 2004)
2004 FORMAT(1X, *ENTER THIRTY CHARACTER IDENTIFIER*/)
READ(5, 1001) MESSG
1001 FCRMAT(5A10)
PSCL(1)=AMX-AMN
PSCL(2)=BMX-BMN
PSCL(3)=CMX-CMN
PSCL(4)=DMX-DMN
OSET(1)=AMN
OSET(2)=BMN
OSET(3)=CMN
OSET(4)=DMN
CALL DISCON(6)
REWIND 6
REWIND 1
REWIND 2
C
1000 CONTINUE
LINE=0
C

```

C	INITIALIZE MATRICES FOR ANALYSIS OF VARIANCES	61
C		62
	DO 620 I=1,12	63
	KHABSI(I)=0	64
	KHACTI(I)=0	65
	KHABSE(I)=0	66
	KHACTE(I)=0	67
620	CONTINUE	68
650	CONTINUE	69
	READ(1,3001) IT,TIME,NIABS,NIESC,NIACT,NEABS,NEESC,NEACT,	70
	1 RMIABS,REMAES,RMIACT,RMEACT,AVIABS,AVEABS,AVIACT,AVEACT,XMAX	
	JLINE=JLINE+1	72
	IF(MOD(JLINE,50).EQ.1) WRITE(6,2001) DENCC	73
2001	FORMAT(1H1,9X,*TABULATED STATISTICAL SUMMARY OF DATA AS A *	
	+*FUNCTION OF TIME. DENCC = *,1PE12.4)	75
	IF(MOD(JLINE,50).EQ.1) WRITE(6,2005) MESS	76
2005	FORMAT(10X,5A10)	77
	IF(MOD(JLINE,50).EQ.1) WRITE(6,3002)	78
	JVAR=MOD(JLINE,10)+1	79
	IF(.EOF(1)) 99,660	80
660	CONTINUE	81
C		82
C	ADD TO REMAINING VARIANCE SUNS	83
C		84
	KHABSI(11)=KHABSI(11)+NIABS-KHABSI(JVAR)	85
	KHACTI(11)=KHACTI(11)+NIACT-KHACTI(JVAR)	86
	KHABSI(12)=KHABSI(12)+NIABS*NIABS-KHABSI(JVAR)*KHABSI(JVAR)	
	KHACTI(12)=KHACTI(12)+NIACT*NIACT-KHACTI(JVAR)*KHACTI(JVAR)	
	KHABSI(JVAR)=NIABS	89
	KHACTI(JVAR)=NIACT	90
	KHABSE(11)=KHABSE(11)+NEABS-KHABSE(JVAR)	91
	KHACTE(11)=KHACTE(11)+NEACT-KHACTE(JVAR)	92
	KHABSE(12)=KHABSE(12)+NEABS*NEABS-KHABSE(JVAR)*KHABSE(JVAR)	
	KHACTE(12)=KHACTE(12)+NEACT*NEACT-KHACTE(JVAR)*KHACTE(JVAR)	
	KHABSE(JVAR)=NEABS	95
	KHACTE(JVAR)=NEACT	96
730	CONTINUE	97
C		98
C	BEGIN CALCULATION OF VARIANCES	99
C		100
	DELT=MIND(JLINE,10)	101
	AVIABS=FLOAT(KHABSI(11))/DELT	102
	AVIACT=FLOAT(KHACTI(11))/DELT	103
	RMIABS=SQRT(FLOAT(KHABSI(12))/DELT-AVIABS*AVIABS)	104
	RMIACT=SQRT(FLOAT(KHACTI(12))/DELT-AVIACT*AVIACT)	105
	AVEABS=FLOAT(KHABSE(11))/DELT	106
	AVEACT=FLOAT(KHACTE(11))/DELT	107
	RMEABS=SQRT(FLOAT(KHABSE(12))/DELT-AVEABS*AVEABS)	108
	RMEACT=SQRT(FLOAT(KHACTE(12))/DELT-AVEACT*AVEACT)	109
3002	FORMAT(1//25X,21HSUMMARY FOR THIS STEP,16X,19HTEN-STEP RMS ERRORS,	
	1 17X,17HTEN-STEP AVERAGES/22X,4HIONS,12X,9HELECTRONS,1X,2(3X,2(7X,	

```

2 1HI,7X,1HE))/1X,5HCYCLE,3X,4HTIME,2(4X,14HABS ESC ACT),
3 2(8X,3HABS,5X,3HABS,5X,3HACT,5X,3HACT),2X,4HPMAX) 113
WRITE(6,3001) IT,TIME,NIABS,NIESC,NIACT,NEABS,NEESC,NEACT,
1 RHIABS,RHEABS,RHIACT,RHEACT,AVIABS,AVEABS,AVIACT,AVEACT,XMAX
WRITE(2,3001) IT,TIME,NIABS,NIESC,NIACT,NEABS,NEESC,NEACT,
1 RHIABS,RHEABS,RHIACT,RHEACT,AVIABS,AVEABS,AVIACT,AVEACT,XMAX
3001 FORMAT(1X,I4,1PE10.3,1X,I3,2I6,3X,I3,2I6,3X,0P4F8.1,3X,4F8.1,F8.3)
GO TO 650 119
99 CONTINUE 120
REWINO 2 121
WRITE(6,2003) DENCC 122
2003 FORMAT(1H1,5X,*PRINTER PLOT SUMMARY OF DATA AS A FUNCTION*
+* OF TIME. DENCC = *,1PE12.4) 124
WRITE(6,2005) MESSG 125
WRITE(6,2006) AMN,AMX,8MN,8MX,CMN,CMX,DMN,DMX 126
2006 FORMAT(/5X,*SYMBOL CORRESPONDENCE*/10X,*A = ION ABSORPTIONS*,
+13X,*SCALED FROM*,5X,F8.1,* TO*,F8.1/ 128
+10X,*B = ION POPULATION*,14X,*SCALED FROM*,5X,F8.1,* TO*,F8.1/
+10X,*C = ELECTRON POPULATION*,9X,*SCALED FROM*,5X,F8.1,* TO*,
+F8.1/10X,*D = MAXIMUM POTENTIAL*,11X,*SCALED FROM*,5X,F8.1,
+* TO*,F8.1//) 132
30 CONTINUE 133
READ(2,3001) IT,TIME,NIABS,NIESC,NIACT,NEABS,NEESC,NEACT,RHIABS,
1 RHEABS,RHIACT,RHEACT,PVEC(1),AVEABS,PVEC(2),PVEC(3),PVEC(4)
IF(EOF(2)) 40,50 136
50 CONTINUE 137
IFLG=0 138
IF(MOD(IT,10).EQ.0) IFLG=1 139
DO 10 II=1,100 140
LINE(II)=1H 141
IF(IFLG.EQ.1.AND.MOD(II,10).EQ.0) LINE(II)=1H+ 142
10 CONTINUE 143
DO 20 II=1,4 144
IPV=(PVEC(II)-OS(T(II)))/PSOL(II)*100.+1. 145
IPV=MINO(MAXO(IPV,1),100) 146
LINE(IPV)=IPSYM(II) 147
20 CONTINUE 148
IF(IFLG.EQ.1) WRITE(6,3004) IT,LINE 149
IF(IFLG.NE.1) WRITE(6,3003) LINE 150
3003 FORMAT(10X,1H*,100A1,1H*) 151
3004 FORMAT(4X,I3,3X,1H*,100A1,1H*) 152
GO TO 30 153
40 CONTINUE 154
END 155

```