NASA TN D-1711

NASA TN D-1711

CASE FILE COPY

TECHNICAL NOTE

D-1711

NUMERICAL SOLUTION OF AXIALLY SYMMETRIC POISSON EQUATION; THEORY AND APPLICATION TO ION-THRUSTOR ANALYSIS

By Vladimir Hamza

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

WASHINGTON May 1963

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

TECHNICAL NOTE D-1711

NUMERICAL SOLUTION OF AXIALLY SYMMETRIC POISSON EQUATION; THEORY AND APPLICATION TO ION-THRUSTOR ANALYSIS

By Vladimir Hamza

SUMMARY

A numerical solution of the axially symmetric Poisson equation for mixed boundary conditions is presented, and the theory and application to an ion-thrustor analysis is discussed.

A method of solution is developed in which the differential equation is replaced by finite difference equations, and the properties of the resulting matrix are studied. The theory of matrix "regular splittings" is applied. The Cyclic Chebyshev Semi-Iterative method is used to solve the matrix equation, and an estimate of an optimum relaxation factor is given. The Poisson equation is solved by a method of successive approximations.

The numerical method is demonstrated with an example of a hollow-cylindrical ion beam. The program for an IBM 7090 computer is also included.

INTRODUCTION

The numerical method of solution of the space-charge-flow problem or, mathematically, the solution of the two-dimensional Poisson equation is developed and presented in reference 1. Application of this two-dimensional method together with its extension involving curved ion-emitter surfaces is demonstrated in reference 2. Good agreement of this method with available experimental data is also reported in reference 2. A method of analysis of an axially symmetric ion beam in an ion thrustor is discussed herein. The material presented represents a logical extension of the work described in references 1 and 2.

After boundary conditions of finite extent are stipulated, the Poisson equation in a continuous form is replaced by finite-difference equations. The finite-difference equations give rise to a real symmetric matrix by virtue of the axial symmetry of the problem. The properties and convenient partitioning of the resulting real symmetric matrix are discussed in detail. The solution of the matrix equation is accomplished by the use of the Cyclic Chebyshev Semi-Iterative method. The general procedure for obtaining a solution of the axially symmetric Poisson equation is essentially the same as that described in

reference 1. The space-charge-density-distribution function is not known a priori; and therefore, the Laplacian equation is solved to determine a first approximation of the potential distribution in the bounded region. The equations of motion are used together with this initial potential distribution, to obtain the ion trajectories and the initial values of the space-charge-density function. A series of successive approximations then gives the solution of the Poisson equation.

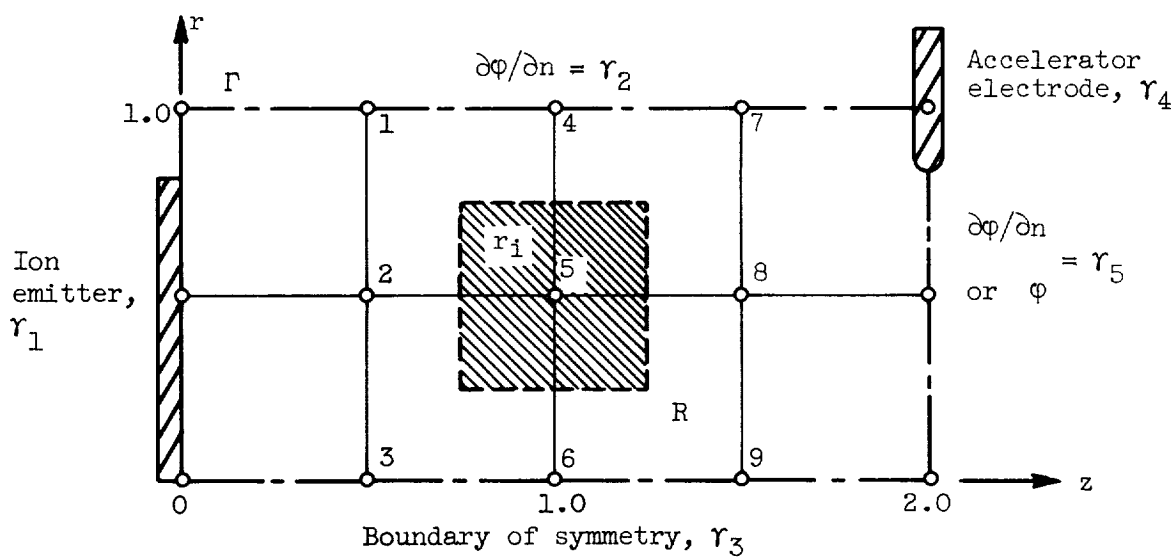
The theory of the numerical method is demonstrated in an example problem. The problem, which was analyzed, is the space-charge-limited flow in a hollow-cylindrical-beam ion thrustor that has a cylindrical ion emitter. The solution was obtained with the aid of an IBM 7090 computer. The computer program is described in appendix C by Carl D. Bogart. No effort was made to optimize the computer program other than the method of obtaining rapid convergence to the solution of the matrix equation. The electrode coordinates together with the method of determination of electrode shapes are given in reference 3.

STATEMENT OF PROBLEM AND NUMERICAL ANALYSIS

The theory of a numerical method of solving the axially symmetric Poisson equation is presented for the steady-state space-charge-limited flow of an ion beam in an ion thrustor. In this section, a mathematical model is determined from a hypothetical axially symmetric ion thrustor, and the finite-difference equations that give rise to the matrix equation are established. The solution of the matrix equation is then developed.

Mathematical Model

A region considered for presentation of the theory of the numerical solution of the axially symmetric Poisson equation is shown in sketch (a). The external



(a)

boundary represents a hypothetical ion thrustor with a disk emitter and a cylindrical ion beam extending in the z-direction. The location of the downstream boundary γ_5 is arbitrarily shown in the plane of the accelerator. For analysis of a real thrustor configuration, the location of this boundary will normally be further downstream, as is described in the example problem.

The Poisson equation in cylindrical coordinates is in the form

$$-\nabla^2\varphi(r,\theta,z) = \frac{1}{\epsilon_o}\rho(\varphi,r,\theta,z) \tag{1}$$

The external boundary Γ of the region R satisfies the equation

$$\alpha\varphi(r,\theta,z) + \beta\frac{\partial\varphi(r,\theta,z)}{\partial n} = \gamma_i \qquad \text{for } i=1,2,3,4,5,\ \alpha \neq \beta \tag{2}$$

Two sets of values of α and β apply: 1,0 and 0,1. Values of $\alpha = 1$ and $\beta = 0$ correspond to Dirichlet boundary conditions, while $\alpha = 0$ and $\beta = 1$ correspond to Neumann boundary conditions. (All symbols are defined in appendix A.)

The potential-distribution function is $\varphi(r,\theta,z)$ and the space-charge-density-distribution function is $\rho(\varphi,r,\theta,z)$ both of which are continuous inside the region R. The space-charge-density distribution function ρ is nonnegative for positive ion flow and is not known a priori. It depends on the potential-distribution function φ, which must satisfy equation (1) and the conditions of equation (2) on the exterior boundary Γ of the region R.

The numerical solution of equation (1) is accomplished by overlaying a discrete number of mesh points on the region R, as shown in sketch (a). The uniformity of mesh spacing overlayed on region R is not essential but was chosen for simplification of the matrix coefficients. The size of mesh spacing is arbitrary and depends on the physical dimensions of the problem and the anticipated potential gradients in certain portions of the region R. For that reason it is possible to have several nets of uniform mesh spacing, as described in reference 1. Next, the differential equation (eq. (1)) is replaced by the finite-difference equations that satisfy the conditions of a subregion, that is, the influence area of each mesh point.

Finite-Difference Equations - Five-Point-Formula Approximation

The Poisson equation (cylindrical coordinates) for the discrete case is

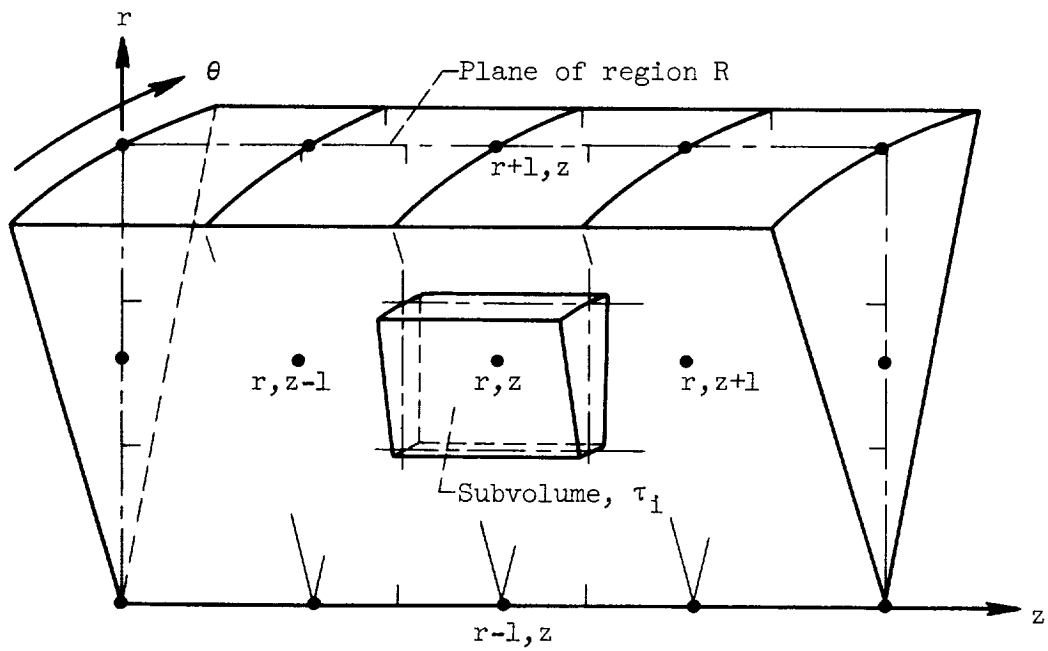
$$-\nabla^2 w(r,\theta,z) = f(w,r,\theta,z) \tag{3}$$

For each of the subvolumes τ_i surrounding the internal point r,z (shown in sketch (b)), the numerical approximation to equation (3) is given by the five-

point-formula approximation as follows:

$$r\left(4w_{r,z} - w_{r,z+1} - w_{r,z-1} - w_{r+1,z} - w_{r-1,z}\right) + \frac{h}{2}\left(w_{r-1,z} - w_{r+1,z}\right) = rh^2 f_{r,z} \tag{4}$$

The derivation of the five-point-formula approximation from Green's three-dimensional theorem can be found in reference 4. For completeness, it is included in appendix B.



(b)

Because of the axial symmetry, equation (4) represents the finite-difference approximation of equation (3) in the r,z plane (identified as R in sketches (a) and (b)). For example, writing equation (4) for point 5 of sketch (a) gives

$$r\left(4w_5 - w_8 - w_2 - w_4 - w_6\right) + \frac{h}{2}\left(w_6 - w_4\right) = rh^2 f_5 \tag{5}$$

The five-point-formula approximation of equation (3) can be expressed for each mesh point surrounded by subregion r_i for $i = 1, 2, \ldots, N$. The truncation error of this approximation is of the order h^2, that is, $O(h^2)$. For subregions along the external boundaries, the five-point-formula approximation of equation (3) must be modified to account for the particular boundary being considered. Various forms of the modified approximation formula are given in appendix B.

The finite-difference approximation of equation (3) for each subregion gives rise to a set of linear algebraic equations. For N mesh points in region R there are N linear algebraic equations with N unknowns. This set of equations can be expressed in matrix form and is discussed next.

Matrix Equation

If the ordering of mesh points is as shown in sketch (a), the N linear algebraic equations with N unknowns can be written in matrix form as

$$A\underline{w} = \underline{k} \tag{6}$$

where $\underline{w}$ is a column vector representing the discrete potential distribution $w_{1,2,\ldots,N}$, $\underline{k}$ is a column vector consisting of the discrete space-charge-density distribution $f_{1,2,\ldots,N}$ and, when applicable, the boundary values $\gamma_{1,2,3,4,5}$. The resulting matrix A is an N by N real symmetric matrix. For the example given in sketch (a), matrix A takes the form

$$A = \left[\begin{array}{ccc|ccc|ccc}
\frac{13}{8} & -\frac{3}{4} & 0 & -\frac{7}{16} & 0 & 0 & 0 & 0 & 0 \\
-\frac{3}{4} & 2 & -\frac{1}{4} & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 \\
0 & -\frac{1}{4} & \frac{3}{8} & 0 & 0 & -\frac{1}{16} & 0 & 0 & 0 \\
\hline
-\frac{7}{16} & 0 & 0 & \frac{13}{8} & -\frac{3}{4} & 0 & -\frac{7}{16} & 0 & 0 \\
0 & -\frac{1}{2} & 0 & -\frac{3}{4} & 2 & -\frac{1}{4} & 0 & -\frac{1}{2} & 0 \\
0 & 0 & -\frac{1}{16} & 0 & -\frac{1}{4} & \frac{3}{8} & 0 & 0 & -\frac{1}{16} \\
\hline
0 & 0 & 0 & -\frac{7}{16} & 0 & 0 & \frac{13}{8} & -\frac{3}{4} & 0 \\
0 & 0 & 0 & 0 & -\frac{1}{2} & 0 & -\frac{3}{4} & 2 & -\frac{1}{4} \\
0 & 0 & 0 & 0 & 0 & -\frac{1}{16} & 0 & -\frac{1}{4} & \frac{3}{8}
\end{array}\right]$$

The solution of the Poisson equation (eq. (3)) now has been reduced to the numerical solution of equation (6). The diagonal entries of matrix A are positive, whereas the off-diagonal entries are nonpositive. It can be proved that the real symmetric matrix A is a Stieltjes matrix, as defined in reference 4,

and thus matrix A has an inverse $A^{-1} > 0$ (therefore A is a nonsingular matrix), which ensures that the solution of equation (6) is unique. It should be noted that it may be necessary to multiply some of the finite-difference equations by appropriate scaling factors in order to ensure that matrix A is symmetric.

Because of the different values of the main diagonal entries, it is no longer convenient, as it was in the two-dimensional case of references 1 and 2, to premultiply the matrix A by a positive diagonal matrix D such that DA is a matrix with unity on its main diagonal. This step would destroy the symmetry of matrix A to which the following discussion is directed.

The solution of the matrix equation (eq. (6)) is based on the theory of "regular splittings" as discussed in reference 4. Using the method of this theory allows expressing matrix A as

$$A = B - C \tag{7}$$

where B and C are again N by N matrices that would be written for the example as

$$B = \begin{bmatrix} b_{1,1} & 0 & 0 \\ 0 & b_{2,2} & 0 \\ 0 & 0 & b_{3,3} \end{bmatrix}$$

where $b_{i,i}$ represents the diagonal block matrices of the example matrix A and

$$C = \begin{bmatrix} 0 & c_{1,2} & 0 \\ c_{2,1} & 0 & c_{2,3} \\ 0 & c_{3,2} & 0 \end{bmatrix}$$

where $c_{i,j}$ represents the off-diagonal blocks of A. Both matrices B^{-1} and C have nonnegative entries that satisfy the definition of regular splittings of matrix A according to reference 4. The theory of regular splittings is a powerful means by which a solution of matrix equations involving matrices more general than tridiagonal matrices can be accomplished very efficiently. This method could be applied as well to the solution of the matrix equation reported in references 1 and 2.

SOLUTION OF MATRIX EQUATION

As mentioned in the previous section, the solution of the axially symmetric

Poisson equation lies within the numerical solution of the matrix equation (eq. (6)). If equation (7) is substituted into the matrix equation (eq. (6)), it follows that

$$A\underline{w} = (B - C)\underline{w} = \underline{k}$$

or equivalently

$$\underline{w} = B^{-1}C\underline{w} + B^{-1}\underline{k} \tag{8}$$

Identifying $D = B^{-1}C$ and $\underline{g} = B^{-1}k$ and substituting in equation (8) yields

$$\underline{w} = D\underline{w} + \underline{g} \tag{9}$$

Equation (9) is now used to solve the matrix equation (eq. (6)).

Cyclic Chebyshev Semi-Iterative Method

The solution of equation (9) is based on the fact that matrix A of equation (6) is a Stieltjes matrix, furthermore, that matrices B and C are defined as a regular splitting of A, and that B is symmetric and positive definite. The iteration matrix $D = B^{-1}C$ of equation (9) is then irreducible, nonnegative, and has real eigenvalues, since D is similar to a symmetrical matrix and is convergent, as shown in reference 4.

Now to explore the cyclic properties of the iteration matrix D. Similarly in the case of matrix M in reference 1 for the two-dimensional solution of the Poisson equation, equation (9) can be written as

$$\begin{bmatrix} \underline{w}_1 \\ \underline{w}_2 \end{bmatrix} = \begin{bmatrix} 0 & D_2 \\ D_1 & 0 \end{bmatrix} \begin{bmatrix} \underline{w}_1 \\ \underline{w}_2 \end{bmatrix} + \begin{bmatrix} \underline{g}_1 \\ \underline{g}_2 \end{bmatrix} \tag{10}$$

where now the subscript 1 is associated with odd-number lines and 2 with even-number lines. By the Cyclic Chebyshev Semi-Iterative method, the vector components of equation (10) can be written as they were in reference 1 for the two-dimensional case, namely,

$$\left.\begin{aligned} \underline{w}_1^{2m+1} &= \omega_{2m+1}\left(D_2\underline{w}_2^{2m} + \underline{g}_1 - \underline{w}_1^{2m-1}\right) + \underline{w}_1^{2m-1} \qquad \text{for} \quad m \geq 1 \\ \underline{w}_2^{2m+2} &= \omega_{2m+2}\left(D_1\underline{w}_1^{2m+1} + \underline{g}_2 - \underline{w}_2^{2m}\right) + \underline{w}_2^{2m} \qquad \text{for} \quad m \geq 0 \end{aligned}\right\} \tag{11}$$

where, for $m = 0$

$$\underline{w}_1^1 = D_2\underline{w}_2^0 + \underline{g}_1$$

Again, this iterative method requires only the initial guess of the single vector component $\underline{w}_2^o$, and the method of solution does not require any more computer storage space than any other iterative procedure.

The relaxation factor ω in equation (11) is given in the form of the Chebyshev polynomials, but for actual computation it is more convenient to express ω as

$$\left.\begin{aligned} \omega_{m+1} &= \frac{1}{1 - \frac{1}{4}\left[\overline{\mu}^2(D)\omega_m\right]} \qquad \text{for} \quad m \geq 2 \\ \omega_1 &= 1 \\ \omega_2 &= \frac{2}{2 - \overline{\mu}^2(D)} \end{aligned}\right\} \tag{12}$$

Spectral Radius of Matrix D

As was pointed out in reference 1, it is very important to choose the relaxation factor ω of equation (11) with great care in order to obtain an optimum rate of convergence of that equation. It is evident from equation (12) that ω is a function of the spectral radius $\overline{\mu}(D)$.

If $\underline{u}$ is any vector with positive components u_i, determination of nontrivial upper and lower bound estimates for the spectral radius of D may be obtained by applying the minimax theorem (ref. 4), which for the ith iteration is given by

$$\min_i\left(\frac{\sum_j d_{i,j}u_j}{u_i}\right) \leq \overline{\mu}(D) \leq \max_i\left(\frac{\sum_j d_{i,j}u_j}{u_i}\right)$$

Since D is irreducible, $\overline{\mu}(D)$ can be expressed finally as

$$\max_{\underline{u}\in R}\left[\min_i\left(\frac{\sum_j d_{i,j}u_j}{u_i}\right)\right] = \overline{\mu}(D) = \min_{\underline{u}\in R}\left[\max_i\left(\frac{\sum_j d_{i,j}u_j}{u_i}\right)\right] \tag{13}$$

The Iterative Procedure

The numerical solution of the axially symmetric Poisson equation (eq. (3)) now follows a procedure similar to that outlined in reference 1. The principal difference is in the iteration-matrix D of equation (9), which was derived in the previous section. Equation (11) is solved first with no space charge to obtain the Laplacian potential distribution inside region R. From this potential

distribution and the equations of motion, ion trajectories are calculated and the first-order space-charge-density-distribution function $f(w,r,\theta,z)$ of equation (3) is determined. There is an additional feature in this computer program for pointwise calculation of ion trajectories that is not included in reference 1. In the event a trajectory goes through an almost 90° bend, a switch in the program changes the direction of sweep (e.g., from r to z) of the increments from station n to station $n + 1$. The accuracy of computation is improved by this method.

A series of successive approximations to the space-charge-density function $f(w,r,\theta,z)$ together with the necessary precautionary checks, as discussed in reference 1, then gives the solution of the axially symmetric Poisson equation.

NUMERICAL EXAMPLE

The example chosen to demonstrate the numerical method of solution of the axially symmetric Poisson equation is a hollow-cylindrical-beam ion thrustor. The mathematical model that has been analyzed is shown in figure 1. The coordi-

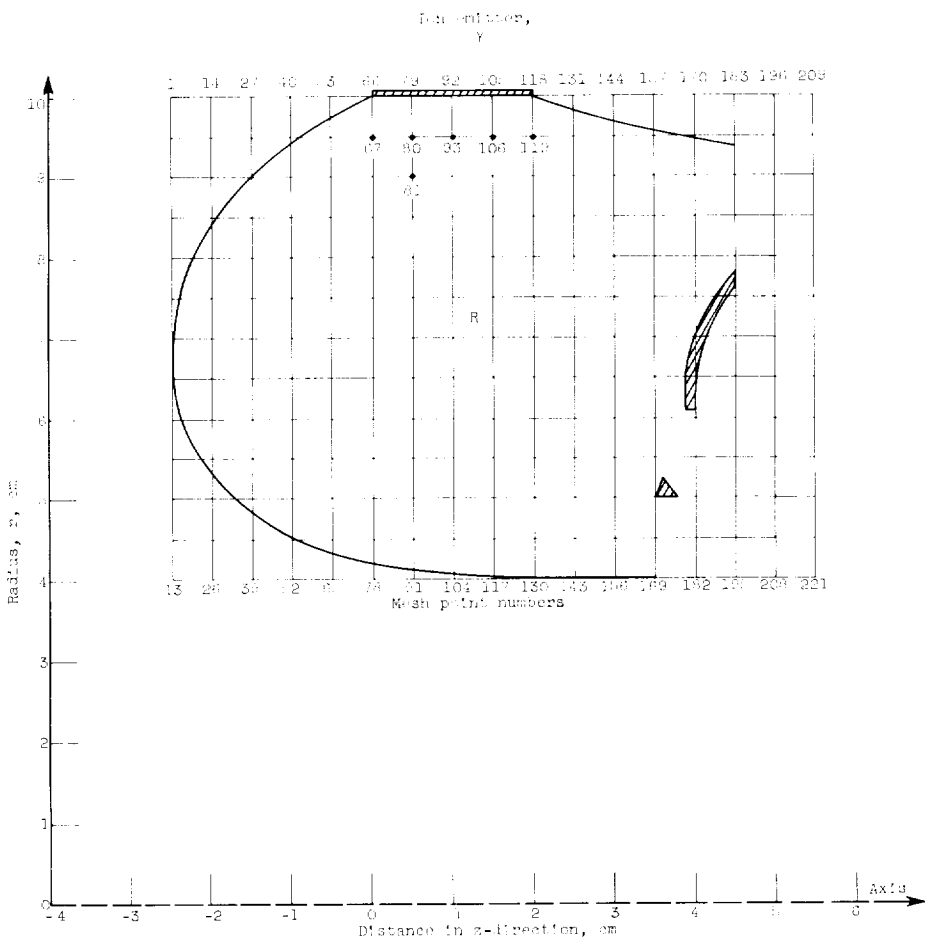
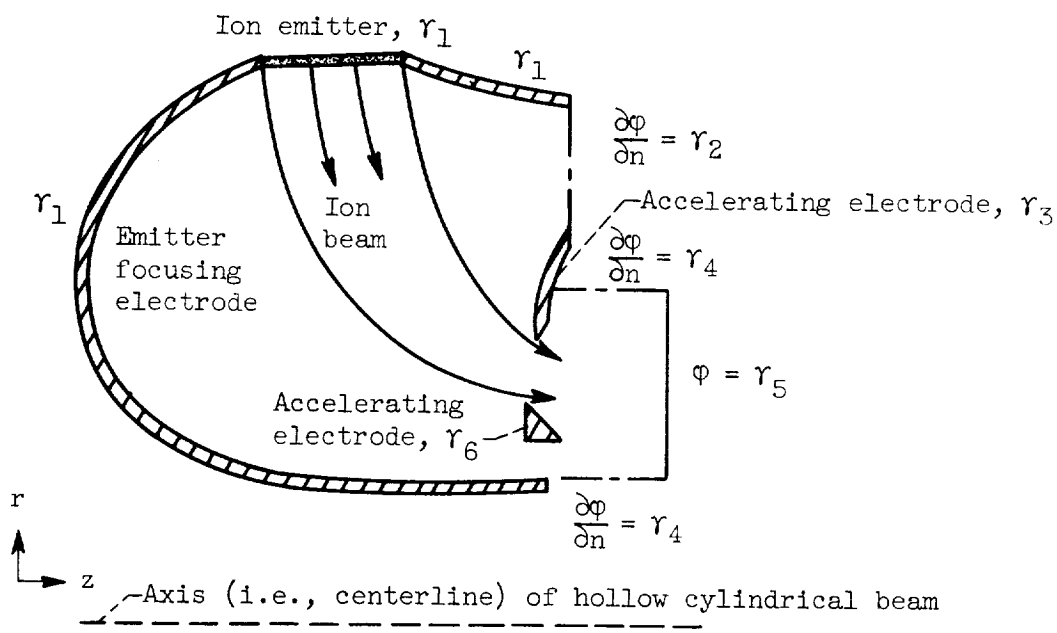


Figure 1. - Mathematical model for example problem from reference 3 for hollow-cylindrical ion beam.

nates of the electrode shapes were obtained from reference 3. The configuration was slightly modified to conform to physical reality, namely, the inner electrodes were assigned a definite thickness rather than lines as given in reference 3. The downstream external boundary conditions were chosen to permit the program to analyze in detail the aperture effect and the impingement current on the electrodes. These effects are of vital interest to good ion-thrustor design. An IBM 7090 computer was used to solve the numerical example. (The computer program is given in appendix C.)

Operation of the hollow-cylindrical beam ion thrustor is as follows: Ions are formed on the ion emitter, for example, the porous tungsten type, which is heated and at a positive potential relative to ground. The accelerator or inner electrodes are at a lower potential than the ion emitter, usually at a negative potential to prevent electrons from entering the thrustor. With the assumption that an adequate flow of propellant is supplied (cesium vapor in this case), the potential field created between the ion emitter and the accelerator electrodes gives rise to space-charge-limited flow of the ions.

The boundaries were chosen as follows (see sketch (c)): (1) the ion emitter and the emitter focusing electrode γ_1 at a uniform potential of 2828.6 volts, (2) the portion of the external boundary between the emitter focusing electrode and the accelerator electrode γ_2 with the normal derivative equal to zero, (3) the accelerating electrode γ_3 at -1592.4 volts, (4) the portion of external



(c)

boundaries behind the accelerating electrodes γ_4 arbitrarily chosen with the normal derivative equal to zero, (5) the problem arbitrarily terminated with

ground potential γ_5, (6) the inner accelerating electrode γ_6 at ground potential. All potentials are referenced to ground. The potentials γ_1, γ_3, and γ_6 resulted from inverting the potentials for electron flow given in reference 3. In this inversion, γ_6 was arbitrarily chosen at ground potential.

The treatment and the location of the downstream boundaries γ_4 and γ_5 are arbitrary from the mathematical viewpoint. For a real ion thrustor, however, these boundary conditions may be of considerable importance. As discussed in reference 1, various possibilities may arise that are primarily related to the problem of beam neutralization. It is not the purpose of this report to show the dependence of the selection of the downstream boundaries on the overall solution of the problem, but the reader is cautioned to use his own judgment or available experimental data to determine the approximate downstream boundary conditions for a given configuration. It was felt that the selection of the downstream boundary as shown in sketch (c) may be a good approximation to the actual operational conditions of this thrustor.

Equipotentials of the solution of the Poisson equation for region R (shown in fig. 1) together with ion trajectories are shown in figure 2. The average current density of the ion beam at the emitter was calculated to be 0.147 ampere per square meter for the potential distribution shown in figure 2. This value is very low because of the "accel length" of 4 to 5 centimeters instead of 4 to 5 millimeters as may occur in real ion-thrustor practice. From examination of the ion trajectories, the impingement current on the accelerator electrodes was calculated to be 5.0 percent of the ion emitter current.

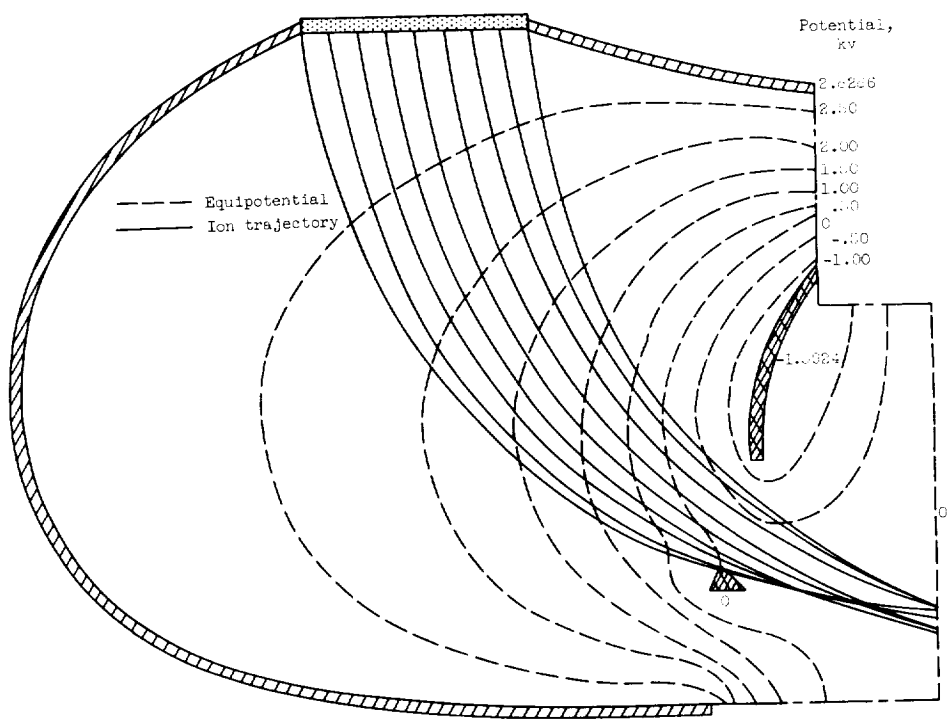


Figure 2. - Equipotentials and ion trajectories for space-charge-limited flow in region R of figure 1.

It is of interest to analyze the right-hand side (RHS) of equation (3), as was done for the two-dimensional case in reference 1, for the first row (or column) from the ion emitter (i.e., mesh points 67, 80, 93, 106, 119 in the first row from the emitter in fig. 1). If it is assumed that the trajectories very close to the emitter may follow straight lines, the RHS for the first row from the ion emitter can be written as

$$f \approx \frac{1}{\epsilon_o} \frac{j_E}{\overline{v}} \tag{14}$$

where j_E is obtained from the Child-Langmuir formula:

$$j_E = \frac{4}{9} \epsilon_o \sqrt{\frac{2q}{m}} \frac{\Delta w^{3/2}}{h^2}$$

From the conservation of energy equation,

$$\overline{v} = \overline{v}_r = \sqrt{\frac{2q}{m} \Delta_r w}$$

If the potential along the first row is assumed almost constant (trajectories are straight lines),

$$\Delta w = \Delta_r w$$

and equation (14) becomes

$$f = \frac{4}{9} \frac{\Delta_r w}{h^2} \tag{15}$$

Substitution of equation (15) into equation (4) written, for example, for mesh point 80 (see fig. 1) gives

$$r\left(4w_{80} - w_{93} - w_{67} - w_{81} - \gamma_1\right) + \frac{h}{2}\left(w_{81} - \gamma_1\right) = \frac{4r}{9} \Delta_r w$$

Thus, when the potential distribution from the solution of the Laplacian equation is known, a quick check of the RHS as a first approximation of the Poisson equation at the first row can be made by multiplying the potential difference between the emitter and the mesh point of the first row (or column) by $(4/9)r$.

It should be noted that the electrode shapes obtained from reference 3 were calculated by a numerical method for a hollow beam described by an analytical solution in which the voltage and electric fields vary exponentially with respect to distance along the axis. The analytic solution of a hollow beam with cylindrical emitter did not take into account the aperture effect. Therefore, it would be expected that the electrode shapes calculated from that analytical solution may be in error for an actual ion-thrustor design.

The calculated impingement current of 5 percent herein might be attributed to accounting for the aperture effect as well as a change of the electric field around the electrodes resulting from introduction of a finite thickness to the electrodes. It is felt, therefore, that the agreement of this numerical method with the method of reference 3 for determination of electrode shapes for axially symmetric ion thrustors is good. Improvement in the ion optics may possibly be obtained by blocking segments of the ion emitter that had contributed the portion of the space-charge flow that was intercepted on the accelerator electrode. This type of improvement of ion optics is discussed in reference 2.

CONCLUDING REMARKS

The objective of this paper was to develop a numerical method of solution of the axially symmetric Poisson equation for mixed boundary conditions. The region for which the solution was sought was overlayed with mesh points, and for each mesh point the differential equation was replaced by finite-difference approximations. These approximations gave rise to a set of linear algebraic equations. For N mesh points in the region R, there were N linear algebraic equations with N unknowns, which resulted in an N by N real symmetric matrix - a Stieltjes matrix.

The theory of regular splittings was used because of the nonuniform main diagonal entries in the matrix. The matrix equation was solved by the Cyclic Chebyshev Semi-Iterative method on an IBM 7090 computer. To obtain fast convergence in the iterative process, an estimate of an optimum relaxation factor was used.

The method of solution presented herein is general for any type of external boundary that satisfies equation (2). The application of this method of solution to the analysis of the space-charge flow in a particular axially symmetric ion-thrustor configuration was demonstrated. Agreement between ion trajectories obtained by this method and those obtained from an analytical solution was very good. The present method is superior to previous analytical methods in that the present method includes full account of the exhaust aperture that exists in real ion thrustors.

The method of solution presented herein may find use as a tool for diagnostic purposes by those working in this area. This method together with the two-dimensional method of analysis previously reported in references 1 and 2 extends the numerical program to include most of the ion-accelerator geometries currently being investigated.

Lewis Research Center
National Aeronautics and Space Administration
Cleveland, Ohio, February 4, 1963

APPENDIX A

SYMBOLS

A	matrix of matrix equation (eq. (6))
A^{-1}	matrix, inverse of matrix A
B	block diagonal matrix (eq. (7))
B^{-1}	matrix, inverse of matrix B
$b_{i,i}$	entries of matrix B
C	off-diagonal block matrix (eq. (7))
$c_{i,j}$	entries of matrix C
D	iteration matrix
$D_{1,2}$	matrix associated with odd- and even-number lines of D, respectively
$d_{i,j}$	entries of matrix D
f	space-charge-density-distribution function for discrete case, v/sq m
$\underline{g}$	column vector, $B^{-1}\underline{k}$
$\underline{g}_{1,2}$	column vectors for odd- and even-number lines, respectively
h	mesh spacing, m
j	current density, amp/sq m
$\underline{k}$	column vector of matrix equation
m	particle mass, kg
N	number of mesh points in region R
n	outward normal
q	unit charge, coulombs
R	region in sketch (a)
r	subregion of R; cylindrical coordinate
u	component of $\underline{u}$

$\underline{u}$ arbitrary column vector

$\overline{v}$ average velocity, m/sec

w potential-distribution function for discrete case, v

$\underline{w}$ column vector of matrix equation

$\underline{w}_{1,2}$ column vectors for odd and even lines, respectively

z cylindrical coordinate

α,β integers, (1 or 0)

Γ external boundary of R

γ discrete portion of external boundary

Δ increment or difference

δ increment of length in sketch (g) and eq. (B10)

∇^2 Laplacian operator

ϵ mathematical symbol representing "belongs to the set"

ϵ_o permittivity of free space, coulombs/(v)(m)

θ cylindrical coordinate

$\overline{\mu}(D)$ spectral radius of matrix D

ρ space-charge-density-distribution function for continuous case, coulombs/cu m

σ surface area bounded by subvolume τ in sketches (b) and (d)

τ subvolume of bounded space shown in sketches (b) and (d)

φ potential-distribution function for continuous case, v

ω relaxation factor

Subscripts:

E emitter

i,j number, 1, 2, . . ., N

m number of iteration

N	number of mesh points
r,z	direction

Superscripts:

m	number of iteration
o	initial guess
1	first iteration

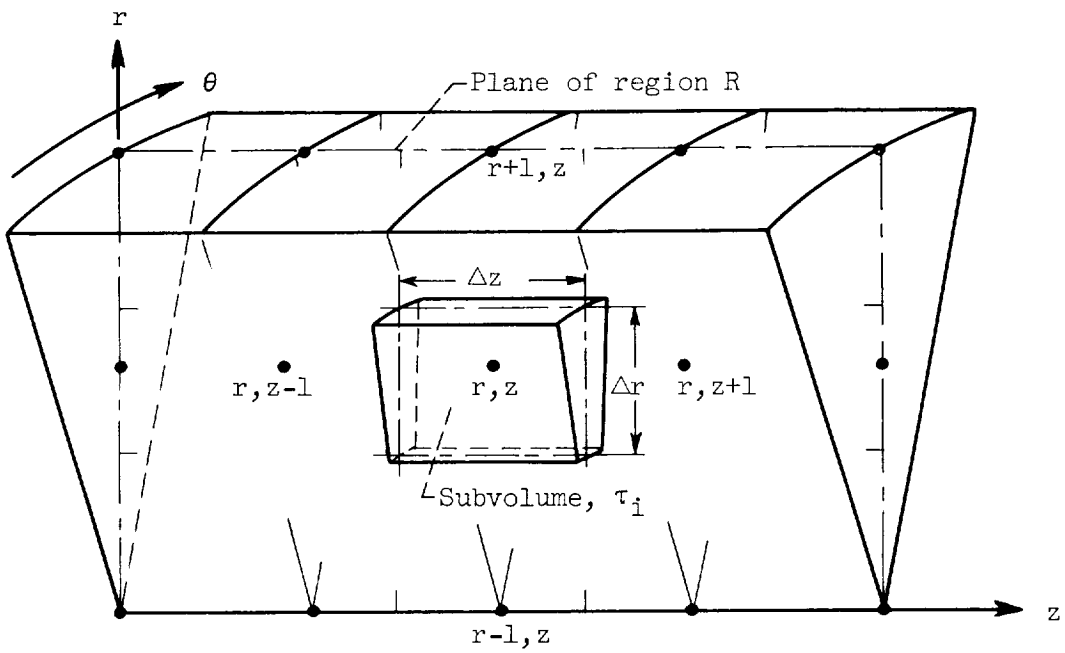
APPENDIX B

DERIVATION OF FINITE-DIFFERENCE EQUATIONS

ACCORDING TO REFERENCE 4

The Poisson equation in cylindrical coordinates is

$$-\nabla^2 w(r,\theta,z) = f(w,r,\theta,z) \tag{B1}$$



(d)

From the integration of equation (B1) over the subvolume τ_i shown in sketch (d) it follows that

$$-\iiint_{\tau_i} \nabla^2 w(r,\theta,z) r \, dr \, d\theta \, dz = \iiint_{\tau_i} f(w,r,\theta,z) r \, dr \, d\theta \, dz \tag{B2}$$

By Green's theorem, the term on the left-hand side of equation (B2) can be reduced to a surface integral over the area σ_i bounding the subvolume τ_i, and equation (B2) can be written as

$$-\iint_{\sigma_i} \frac{\partial w(r,\theta,z)}{\partial n} d\sigma = \iiint_{\tau_i} f(w,r,\theta,z) r \, dr \, d\theta \, dz \tag{B3}$$

where $\partial w(r,\theta,z)/\partial n$ is the derivative in the direction of the outward normal to σ_i.

To obtain a five-point-formula approximation, the following numerical approximations to the integrals of equation (B3) are made. The function $f(w,r,\theta,z)$ is assumed to be constant for the subvolume τ_i, and therefore the RHS of equation (B3) becomes

$$\iiint_{\tau_i} f(w,r,\theta,z) r \, dr \, d\theta \, dz \approx f_i \iiint_{\tau_i} r \, dr \, d\theta \, dz \tag{B4}$$

For the axially symmetric case, w is independent of θ and the normal derivatives of the left-hand side of equation (B3) are approximated by the central difference formula as

$$\frac{\partial w}{\partial z}\left(r, z + \frac{1}{2}\right) \approx \frac{w(r,z+1) - w(r,z)}{\Delta z} \tag{B5}$$

for the z-direction shown in sketch (d). Substitution of the preceding approximations into equation (B3) and integration gives the five-point formula, which can be written as

$$\begin{aligned} -\Bigg\{ & \frac{w_{r,z+1} - w_{r,z}}{\Delta z}\left[\pi\left(r + \frac{\Delta r}{2}\right)^2 - \pi\left(r - \frac{\Delta r}{2}\right)^2\right]\frac{\Delta\theta}{2\pi} + \frac{w_{r,z-1} - w_{r,z}}{\Delta z} \\ & \times \left[\pi\left(r + \frac{\Delta r}{2}\right)^2 - \pi\left(r - \frac{\Delta r}{2}\right)^2\right]\frac{\Delta\theta}{2\pi} + \frac{w_{r+1,z} - w_{r,z}}{\Delta r}\left[\left(r + \frac{\Delta r}{2}\right)\Delta\theta\,\Delta z\right] \\ & + \frac{w_{r-1,z} - w_{r,z}}{\Delta r}\left(r - \frac{\Delta r}{2}\right)\Delta\theta\,\Delta z\Bigg\} = f_{r,z} r\,\Delta r\,\Delta\theta\,\Delta z \end{aligned} \tag{B6}$$

To simplify the matrix coefficients, it is necessary to choose $\Delta r = \Delta z = h$. Equation (B6) then reduces to

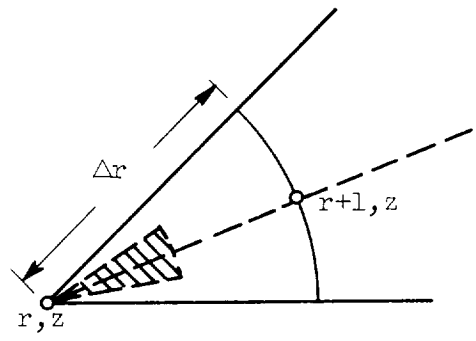
$$r\left(4w_{r,z} - w_{r,z+1} - w_{r,z-1} - w_{r+1,z} - w_{r-1,z}\right) + \frac{h}{2}\left(w_{r-1,z} - w_{r+1,z}\right) = rh^2 f_{r,z} \tag{B7}$$

For $r = 0$ (see sketch (e)), the five-point-formula approximation of equation (B3) is

$$\begin{aligned} -\Bigg\{ & \frac{w_{r,z+1} - w_{r,z}}{\Delta z}\left[\pi\left(\frac{\Delta r}{2}\right)^2 \frac{\Delta\theta}{2\pi}\right] + \frac{w_{r,z-1} - w_{r,z}}{\Delta z}\left[\pi\left(\frac{\Delta r}{2}\right)^2 \frac{\Delta\theta}{2\pi}\right] \\ & + \frac{w_{r+1,z} - w_{r,z}}{\Delta r}\left(\frac{\Delta r}{2}\Delta\theta\,\Delta z\right)\Bigg\} = f_{r,z}\pi\left(\frac{\Delta r}{2}\right)^2 \frac{\Delta\theta}{2\pi}\Delta z \end{aligned}$$

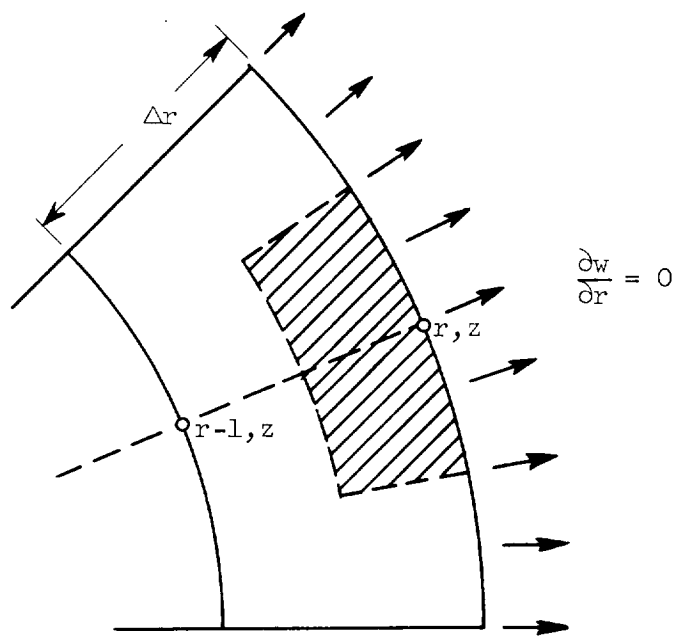
For $\Delta r = \Delta z = h$, it follows that

$$\frac{3w_{r,z}}{4} - \frac{w_{r,z+1}}{8} - \frac{w_{r,z-1}}{8} - \frac{w_{r+1,z}}{2} = \frac{h^2}{8} f_{r,z} \tag{B8}$$



(e)

For the case where the normal derivative is specified at the external boundary (see sketch (f)), the five-point-formula approximation of equation (B3) is



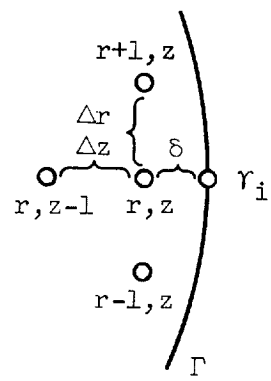
(f)

$$-\left\{\frac{w_{r,z+1} - w_{r,z}}{\Delta z}\left[\pi r^2 - \pi\left(r - \frac{\Delta r}{2}\right)^2\right]\frac{\Delta\theta}{2\pi} + \frac{w_{r,z-1} - w_{r,z}}{\Delta z}\left[\pi r^2 - \pi\left(r - \frac{\Delta r}{2}\right)^2\right]\frac{\Delta\theta}{2\pi}\right.$$
$$\left. + \frac{w_{r-1,z} - w_{r,z}}{\Delta r}\left(r - \frac{\Delta r}{2}\right)\Delta\theta\,\Delta z\right\} = f_{r,z}\left[\pi r^2 - \pi\left(r - \frac{\Delta r}{2}\right)^2\right]\frac{\Delta\theta}{2\pi}\Delta z$$

For $\Delta r = \Delta z = h$, it follows that

$$\left(2r - \frac{3h}{4}\right)w_{r,z} - \left(\frac{r}{2} - \frac{h}{8}\right)w_{r,z+1} - \left(\frac{r}{2} - \frac{h}{8}\right)w_{r,z-1} - \left(r - \frac{h}{2}\right)w_{r-1,z} = f_{r,z}\,\frac{h^2}{2}\left(r - \frac{h}{4}\right) \tag{B9}$$

For regions with curved external boundaries (see sketch (g)) the numerical



(g)

approximation around the curved portion of the boundary Γ is as follows:

$$-\left\{\frac{w_{r+1,z} - w_{r,z}}{\Delta r}\left(r + \frac{\Delta r}{2}\right)\left(\frac{\Delta z + \delta}{2}\right)\Delta\theta + \frac{w_{r-1,z} - w_{r,z}}{\Delta r}\left(r - \frac{\Delta r}{2}\right)\left(\frac{\Delta z + \delta}{2}\right)\Delta\theta\right.$$
$$+ \frac{\gamma_i - w_{r,z}}{\delta}\left[\pi\left(r + \frac{\Delta r}{2}\right)^2 - \pi\left(r - \frac{\Delta r}{2}\right)^2\right]\frac{\Delta\theta}{2\pi} + \frac{w_{r,z-1} - w_{r,z}}{\Delta z}$$
$$\left.\times\left[\pi\left(r + \frac{\Delta r}{2}\right)^2 - \pi\left(r - \frac{\Delta r}{2}\right)^2\right]\frac{\Delta\theta}{2\pi}\right\} = f_{r,z}\left[\pi\left(r + \frac{\Delta r}{2}\right)^2 - \pi\left(r - \frac{\Delta r}{2}\right)^2\right]\frac{\Delta\theta}{2\pi}\left(\frac{\Delta z + \delta}{2}\right)$$

For $\Delta r = \Delta z = h$, it reduces to

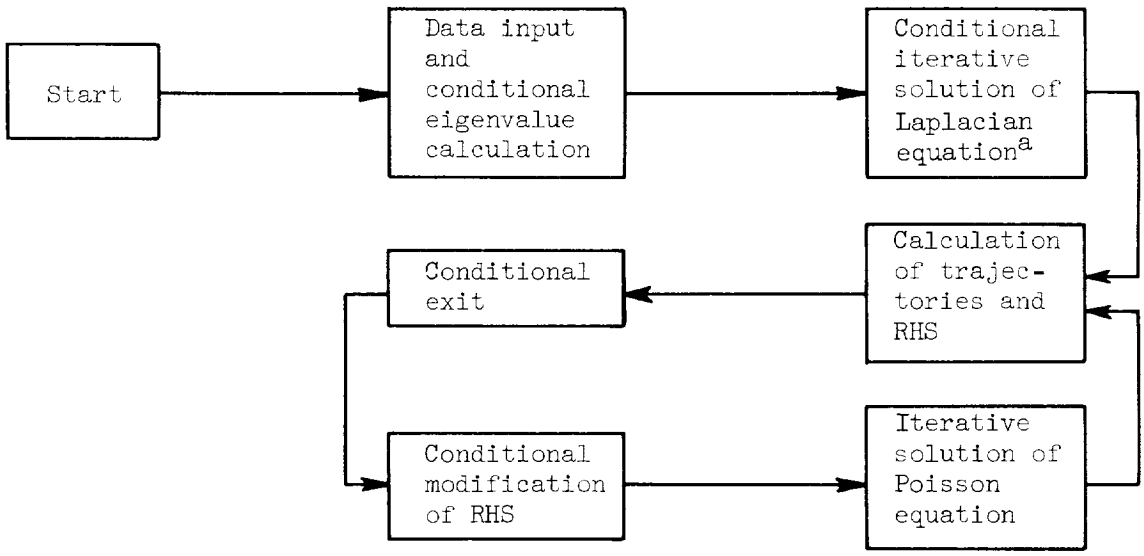
$$\frac{r}{\delta h}(h + \delta)^2 w_{r,z} - \left(r + \frac{h}{2}\right)\left(\frac{h + \delta}{2h}\right)w_{r+1,z} - \left(r - \frac{h}{2}\right)\left(\frac{h + \delta}{2h}\right)w_{r-1,z}$$
$$- \frac{rh}{\delta}\gamma_i - rw_{r,z-1} = f_{r,z}rh\left(\frac{h + \delta}{2}\right) \tag{B10}$$

APPENDIX C

IBM 7090 ION-THRUSTOR FORTRAN CODE AND BLOCK DIAGRAM

By Carl D. Bogart

A schematic representation of the Fortran program is shown, followed by the symbol list, a flow chart of the program (fig. 3), and a complete Fortran listing with the data for a sample case.



[a]Same program as Poisson solution with RHS equal to zero.

FORTRAN CODE SYMBOL LIST

Control Words:

ATX x-emitter coordinates

ATY y-emitter coordinates

IXO positive, RHS prints out; negative or zero, RHS will not print out

JOT number of lines of trajectories to be printed out

KAB number of lines to traverse to obtain equipotential line

KAN test point for RHS

KBA test point for equipotential line if current is calculated from equal delta potential

KBN	vector to test for space-charge-limited flow
KCY	positive, all print-outs occur; negative or zero, no print-outs
KISW	positive, calculation of current is with equal delta-x; negative or zero, calculation of current is on equal delta potential
KRL	cycle counter
LINC	vector determining order of calculation of lines
MO	positive, test RHS upper bound; negative, test RHS lower bound; zero, no test
NCOOR	positive, emitter coordinates are beginning trajectory coordinates; negative or zero, beginning trajectory coordinates are calculated
NEM	number of emitter coordinates
NH	number of heading cards
NJOT	number of lines of trajectory output for Poisson solution
NLB	number of LB's to be read in
NLC	number of LC's to be read in
NLD	number of LD's to be read in
NLIN	number of lines in matrix equation
NPIT	negative, eigenvalue to be calculated; zero, potential input is from resistance paper; positive, potential input is from dump for restart
NPOL	first point in equipotential or first point for calculation of current from equal delta-x
NPONT	number of points per line
NRL	number of cycles
NSPAN	width of y-sweep if equipotential is calculated for horizontal emitter
NSWP	positive or negative, sweep is in y-direction; zero, sweep is in x-direction initially
NTJ	number of trajectories
NTOP	total number of points
NTP	number of KT's or XT's plus eight

NUL number of iterations on matrix equation

NURL initially to change NUL for Poisson; later as a switch to indicate end of problem

Problem Specifications:

A atomic number of ions

EPS convergence test for matrix equation

H mesh size

HGH lower coordinate of upper electrode

HSL maximum slope test to change sweep

JT vector of type numbers

KT vector of relative subscripts

LB vector defining first and last point to be calculated for each line

LC vector determining trajectory and RHS calculation

LD vector determining equipotentials calculation

NJT number of JT's

NKAN number of test points for over space-charge-limited flow plus one

RX suppression factor for RHS

SIZE step size in equipotential calculation

VA emitter potential

VAT upper potential in equipotentials calculation

VB upper electrode potential

VBT lower potential in equipotentials calculation

VC lower electrode potential

XLOW upper coordinate of lower electrode

XLSL minimum slope test to change sweep

XM error function term

XN error function term

XQM charge-to-mass ratio

XR spectral radius of matrix

XT vector of relative weights

YEP permittivity of free space

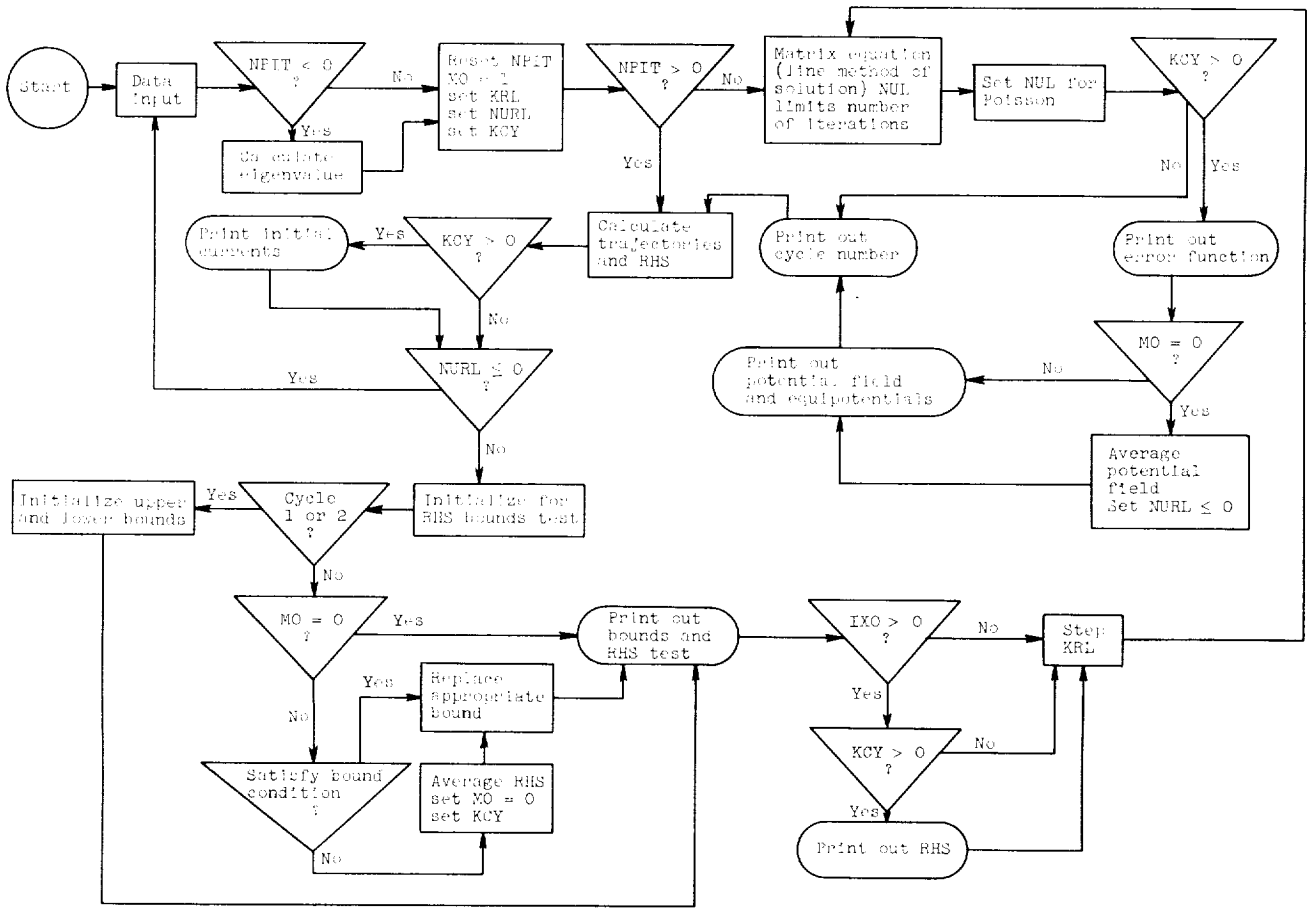


Figure 3. - Flow chart.

```
                                                                        MAIN1
C MAIN ONE -DATA INPUT
      COMMON URH,UB,JT,RH,U,XT,KT,LINC,LB,XR,NTOP,NUL,NLIN,NREG,NTP,
     1 NPIT,XEP,NXEP,CU,NPCNT,AX,AY,VX,VY,DX,DELY,YEP,XCM,F,KISW,ETX,
     2 ETY,PTY,PTX,NAJ,NTJ,KBA,VA,VB,VC,NCCCR,KCH,HSL,XLSL,EPS,NPOL,
     3 NPUL,JOT,KAB,NSPAN,RX,NRL,KRL,KAN,LC,LD,VAT,VBT,SIZE,KBN,RHUP,
     4 RHDOWN,XCU,NOT,MC,KCY,NSWP,ATX,ATY,KAT,KATT,KABB,NURL,NJCT,
     5 IXO,HGH,XLOW,XMPR,NEM
      DIMENSION KBN(20),LD(4),LC(10),URH(4000),UB(4000),JT(4000),
     1 RH(4000),U(4000),XT(630),KT(630),LINC(50),LB(200),CU(40),VY(40)
     2 ,VX(40),KCH(40),PTX(40),PTY(40),AY(40),AX(40),ATX(40),ATY(40),
     3 ETX(40),ETY(40),XCU(40),KCY(10)
 200  MO=1
      READ INPUT TAPE 7,100,NH
      DO 13 J=1,NH
      READ INPUT TAPE 7,107
 13   WRITE OUTPUT TAPE 6,107
      READ INPUT TAPE 7,100,NTOP,NPCNT,KAN,NSWP,NURL,NJCT,IXO
      READ INPUT TAPE 7,100,NUL,NLIN,NLB,NTP,NJT,NPIT
      READ INPUT TAPE 7,100,(LINC(J),J=1,NLIN)
      READ INPUT TAPE 7,100,(LB(J),J=1,NLB)
      DO 1 J=9,NTP,6
      K=J+5
      READ INPUT TAPE 7,101,(KT(M),M=J,K)
 1    READ INPUT TAPE 7,102,(XT(M),M=J,K)
      READ INPUT TAPE 7,103,(JT(M),M=1,NJT)
      READ INPUT TAPE 7,102,XR
      IF(NPIT) 2,3,23
C MATIN CALCULATES THE EIGENVALUE USED IN THE ITERATION SCHEME
 2    CALL MATIN
 3    JB=1
      DO 5 JL=1,NLIN
      READ INPUT TAPE 7,100,NC,NL,NR,NS
      READ INPUT TAPE 7,104,(LB(J),J=1,NC)
      NL=NL+LB(JB)
      NR=NR+LB(JB)
      DO 7 K=1,NC
      DO 4 JB=NL,NR
 4    U(JB)=UB(K)
      NL=NL+NS
 7    NR=NR+NS
 5    JB=JB+4
      GO TO 24
 23   NTOP=NTOP
      CALL BCREAD(U(NTOP),U(1))
 24   DO 6 J=1,NTOP
      UB(J)=U(J)
 6    RH(J)=0.
      READ INPUT TAPE 7,100,NCCCR,KBA,NTJ,NEM,NPOL,NPUL,KAB,NSPAN,JCT
     1 ,NLC,KISW,NRL,NKAN,NLD
      READ INPUT TAPE 7,100,(KBN(J),J=1,NKAN)
      READ INPUT TAPE 7,100,(LC(J),J=1,NLC)
      READ INPUT TAPE 7,100,(LD(J),J=1,NLD)
      READ INPUT TAPE 7,102,VA,VB,VC,H,EPS,RX
      READ INPUT TAPE 7,102,HSL,XLSL,HGH,XLOW,XN,XM
```

```
      XMPR=2.*(XN*XM)**2/(XN**2+XM**2)
      READ INPUT TAPE 7,1C5,YEP,XQM,A
      READ INPUT TAPE 7,1C6,VAT,VBT,SIZE
      XQM=XQM/A
      READ INPUT TAPE 7,1C4,(ATX(J),J=1,NEM)
      READ INPUT TAPE 7,1C4,(ATY(J),J=1,NEM)
      IF(NCOOR) 1C,10,11
 11   DO 12 J=1,NEM
      ETX(J)=ATX(J)
 12   ETY(J)=ATY(J)
 10   CONTINUE
      KRL=NRL
      READ INPUT TAPE 7,1CC,NPIT
      READ INPUT TAPE 7,1CC,(KCY(J),J=1,14)
      IF(NPIT) 25,25,26
C UCAL SOLVES THE MATRIX EQUATION
 25   CALL UCAL
C MNTRI CALCULATES THE RHS AND TRAJECTORIES
 26   CALL MNTRI
      GO TO 2C0
 100  FORMAT(14I5)
 101  FORMAT(6I5)
 102  FORMAT(6F10.5)
 103  FORMAT(13I5)
 104  FORMAT(7F10.5)
 105  FORMAT(7E10.5)
 106  FORMAT(3F10.5)
 107  FORMAT(72H
     1
      END
```

```
      SUBROUTINE MATIN
C MATIN CALCULATES THE EIGENVALUE
      COMMON URH,UB,JT,RH,L,XT,KT,LINC,LB,XR,NTOP,NUL,NLIN,NREC,NTP,
     1 NPIT,XEP,NXEP,CL,NPCNT,AX,AY,VX,VY,DX,DELY,YEP,XCM,F,KISW,ETX,
     2 ETY,PTY,PTX,NAJ,NIJ,KBA,VA,VB,VC,NCCCR,KCH,HSL,XLSL,EPS,NPCL,
     3 NPUL,JOT,KAB,NSPAN,RX,NRL,KRL,KAN,LC,LD,VAT,VBT,SIZE,KBN,RHUP,
     4 RHDOWN,XCL,NOT,MC,KCY,NSWP,ATX,ATY,KAT,KATT,KABB,NURL,NJCT,
     5 IXO,HGH,XLOW,XMPR,NEM
      DIMENSION KBN(20),LD(4),LC(10),URH(4000),UB(4000),JT(4000),
     1 RH(4000),L(4000),XT(630),KT(630),LINC(50),LB(200),CU(40),VY(40)
     2 ,VX(40),KCH(40),PTX(40),PTY(40),AY(40),AX(40),ATX(40),ATY(40),
     3 ETX(40),ETY(40),XCU(40),KCY(10)
      DIMENSION A(63),X(21)
      EQUIVALENCE (URH(1),A(1)),(URH(100),X(1))
C MIT=NUMBER OF ITERATIONS ON EIGENVALUE CALCULATION
C KWR=- OR 0,NO INTERMEDIATE OUTPUT
C KWR=+,PRINT INTERMEDIATE ITERATIONS ON EIGENVALUE CALCULATIO
      READ INPUT TAPE 7,102,MIT,KWR
C COLUMN VECTOR INITIALIZED
 50   DO 46 J=1,NTOP
      IF(JT(J))43,43,44
 43   U(J)=0.
      GO TO 46
 44   U(J)=1.
 46   CONTINUE
      JS=-1
C ITERATIVE LOOP
      DO 41 M=1,MIT
      JB=1
C LOOP ON NUMBER OF LINES
      DO 31 ND=1,NLIN
      IF(LINC(ND)) 23,31,23
 23   KA=LB(JB)
      KB=LB(JB+1)+KA
      KC=LB(JB+2)+KA
      JV=1
      JU=1
      DO 13 K=KB,KC
      SUM=0.
      JZ=JT(K)
      IF(JZ) 13,13,14
 14   DO 15 JY=3,5
      JX=JZ+JY
      A(JU)=XT(JX)
 15   JU=JU+1
      DO 16 JY=1,2
      JX=JZ+JY
      JW=K+KT(JX)
 16   SUM=SUM+XT(JX)*L(JW)
      X(JV)=SUM
      JV=JV+1
 13   CONTINUE
      N=KC-KB+1
      CALL MATRIX(N,A(2),X)
C MATRIX IS USED FOR THE FORWARD AND BACK SUBSTITUTION
      JV=1
```

```
      DO 17 J=KB,KC
      RH(J)=X(JV)
17    JV=JV+1
31    JB=JB+4
C SWITCH ALLOWING MATRIX TO BE APPLIED TWICE
      IF(JS) 32,34,34
32    DO 33 J=1,NTOP
      UB(J)=U(J)
33    U(J)=RH(J)
      GO TO 41
C DETERMINATION OF SMALLEST AND LARGEST RATIOS FOR EIGENVALUE
34    XL=0.
      XS=1.
      DO 39 JD=1,NTOP
      IF(U(JD))39,39,35
35    X=RH(JD)/UB(JD)
      IF(XL-X)36,37,37
36    XL=X
      NNL=JD
37    IF(XS-X)39,39,38
38    XS=X
      NS=JD
39    CONTINUE
      IF(KWR) 51,51,52
52    WRITE OUTPUT TAPE 6,101,M,XS,XL,NS,NNL
51    YL=RH(NNL)
      DO 40 JD=1,NTOP
40    U(JD)=RH(JD)/YL
      IF(XL-XS-1.0E-07)42,42,41
41    JS=-JS
42    XR=SQRTF(.5*(XL+XS))
      WRITE OUTPUT TAPE 6,103,XR
47    RETURN
101   FORMAT(20H     LOW    HIGH        I5,2F13.8,5H       2I6)
102   FORMAT(14I5)
103   FORMAT(4H0XR= F10.8)
      END
```

```
      SUBROUTINE MATRIX(N,A,X)
C MATRIX IS USED FOR THE FORWARD AND BACK SUBSTITUTION
         COMMON URH,UB,JT,RH,U,XT,KT,LINC,LB,XR,NTOP,NUL,NLIN,NREG,NTP,
     1   NPIT,XEP,NXEP,CL,NPCNT,AX,AY,VX,VY,DX,DELY,YEP,XGM,F,KISW,ETX,
     2   ETY,PTY,PTX,NAJ,NTJ,KBA,VA,VB,VC,NCCCR,KCH,HSL,XLSL,EPS,NPOL,
     3   NPUL,JOT,KAB,NSPAN,RX,NRL,KRL,KAN,LC,LD,VAT,VBT,SIZE,KBN,RHUP,
     4   RHDOWN,XCU,NOT,MC,KCY,NSWP,ATX,ATY,KAT,KATT,KABB,NURL,NJCT,
     5   IXU,FGH,XLOW,XMPR,NEM
         DIMENSION KBN(20),LD(4),LC(10),URH(4000),UB(4000),JT(4000),
     1   RH(4000),U(4000),XT(630),KT(630),LINC(50),LB(200),CU(40),VY(40)
     2   ,VX(40),KCH(40),PTX(40),PTY(40),AY(40),AX(40),ATX(40),ATY(40),
     3   ETX(40),ETY(40),XCL(40),KCY(10)
         DIMENSION A(63),X(21)
      M=3*N-2
         A(M+1)=0.
      A(2)=A(2)/A
      X=X/A
         IF(N-1) 12,12,9
 9    K=2
      DO 10 J=4,M,3
      A(J)=A(J)-A(J-1)*A(J-2)
      A(J+1)=A(J+1)/A(J)
      X(K)=(X(K)-A(J-1)*X(K-1))/A(J)
 10   K=K+1
         K=K-1
      DO 11 J=1,M,3
      NB=M-J-1
      K=K-1
 11   X(K)=X(K)-A(NB)*X(K+1)
 12      RETURN
         END
```

```
      SUBROUTINE UCAL
C UCAL  SOLVES THE MATRIX EQUATION
      COMMON URH,UB,JT,RH,U,XT,KT,LINC,LB,XR,NTCP,NUL,NLIN,NREC,NTP,
     1 NPIT,XEP,NXEP,CL,NPCNT,AX,AY,VX,VY,DX,DELY,YEP,XCM,F,KISW,ETX,
     2 ETY,PTY,PTX,NAJ,NTJ,KBA,VA,VB,VC,NCCCR,KCH,HSL,XLSL,EPS,NPCL,
     3 NPUL,JOT,KAB,NSPAN,RX,NRL,KRL,KAN,LC,LD,VAT,VBT,SIZE,KBN,RHUP,
     4 RHDOWN,XCU,NOT,MC,KCY,NSWP,ATX,ATY,KAT,KATT,KABB,NURL,NJCT,
     5 IXO,FGH,XLOW,XMPR,NEM
      DIMENSION KBN(20),LD(4),LC(10),URH(4000),UB(4000),JT(4000),
     1 RH(4000),U(4000),XT(630),KT(630),LINC(50),LB(200),CU(40),VY(40)
     2 ,VX(40),KCH(40),PTX(40),PTY(40),AY(40),AX(40),ATX(40),ATY(40),
     3 ETX(40),ETY(40),XCU(40),KCY(10)
      DIMENSION A(63),X(21)
      EQUIVALENCE(URH(1),A(1)),(URH(100),X(1))
      SX=1.
46    XW=1.
      DO 30 NLU=1,NUL
      NEZ=NLU
      XEP=0.
      XM=.25*XR**2
      NOD=1
38    JB=1
      DO 29 NL=1,NLIN
      IF(LINC(NL)) 35,35,33
33    KA=LB(JB)
      KB=LB(JB+1) +KA
      KC=LB(JB+2) +KA
      JV=1
      JU=1
      DO 28 K=KB,KC
      SUM=0
      JZ=JT(K)
      IF(JZ) 28,28,27
27    DO 22 JY=3,5
      JX=JZ+JY
      A(JU)=XT(JX)
22    JU=JU+1
      DO 26 JY=1,5
      JX=JZ+JY
      IF(KT(JX)) 34,26,34
34    IF(XT(JX)) 26,26,25
25    JW=K+KT(JX)
      SUM=SUM+XT(JX)*U(JW)
26    CONTINUE
      X(JV)=SUM+XT(JZ)*RH(K)
      JV=JV+1
28    CONTINUE
      N=KC-KB+1
C MATRIX IS USED FOR THE FORWARD AND BACK SUBSTITUTION
      CALL MATRIX (N,A(2),X)
      JV=1
      DO 24 K=KB,KC
      IF(JT(K)) 24,24,23
23    DIF=X(JV)-U(K)
      JV=JV+1
C MATRIX EQUATION
```

```
      U(K)=XW*DIF+U(K)
      DIF=ABSF(DIF)
      IF(DIF-XEP) 24,24,21
21    XEP=DIF
      NXEP=K
24    CONTINUE
35    JB=JB+4
29    LINC(NL)=-LINC(NL)
      IF(XW-1.)  40,31,40
40    XW=1./(1.-XM*XW)
      GO TO 41
31    XW=1./(1.-2.*XM)
41    IF(NOD) 37,37,36
36    NOD=0
      GO TO 38
37    XCON=XEP*XMPR
      IF(XCON -EPS) 42,42,30
30    CONTINUE
42    KNUT=KBN(1)
      J=2
      DO 43 K=1,KNUT
      JN=KBN(J)
      IF(U(JN)-VA) 43,44,44
43    J=J+1
      GO TO 47
44    SX=SX*RX
      WRITE OUTPUT TAPE 6,100,JN,U(JN),SX
      DO 45 J=1,NTOP
      U(J)=UB(J)
45    RH(J)=RH(J)*RX
      KAN=KAN
      RHUP=RH(KAN)
      GO TO 46
47    IF(NRL-KRL) 2,1,2
1     NUL=NURL
      NTOP=NTOP
      CALL BCDUMP (U(NTOP),U(1))
2     IWRL=NRL-KRL+XABSF(MC)
      IF(KCY(IWRL)) 4,4,3
C TWOUT PRINTS OUT THE POTENTIAL FIELD
3     CALL TWOUT(NEZ)
C EQLINE CALCULATES THE EQUIPOTENTIALS
      CALL EQLINE
4     WRITE OUTPUT TAPE 6,101,IWRL
      RETURN
100   FORMAT(31HOPOINT/VALUE/TOTAL SUPRESSION   I5,2F15.5)
101   FORMAT(8HCR LOOP   I2)
      END
```

```
      SUBROUTINE EQLINE
C EQLINE SOLVES FOR THE EQUIPOTENTIALS
      COMMON URH,UB,JT,RH,U,XT,KT,LINC,LB,XR,NTCP,NUL,NLIN,NREC,NTP,
     1 NPIT,XEP,NXEP,CL,NPCNT,AX,AY,VX,VY,DX,DELY,YEP,XCM,F,KISW,ETX,
     2 ETY,PTY,PTX,NAJ,NTJ,KBA,VA,VB,VC,NCCCR,KCH,HSL,XLSL,EPS,NPOL,
     3 NPUL,JOT,KAB,NSPAN,RX,NRL,KRL,KAN,LC,LD,VAT,VBT,SIZE,KBN,RHUP,
     4 RHDOWN,XCL,NOT,MC,KCY,NSWP,ATX,ATY,KAT,KATT,KABB,NURL,NJCT,
     5 IXU,HGH,XLOW,XMPR,NEM
      DIMENSION KBN(20),LD(4),LC(10),URH(4000),UB(4000),JT(4000),
     1 RH(4000),U(4000),XT(630),KT(630),LINC(50),LB(200),CU(40),VY(40)
     2 ,VX(40),KCH(40),PTX(40),PTY(40),AY(40),AX(40),ATX(40),ATY(40),
     3 ETX(40),ETY(40),XCL(40),KCY(10)
      JB=1
      IF(SIZE)  1,27,1
1     POTEN=VAT
2     JE=LD(JB)
      JD=LD(JB+1)-1
      JC=LD(JB+2)
      DX=LD(JB+3)
      DX=DX*H
      BX=0.
      JED=JE+JD
      L=1
      IF(JL-1) 3,3,4
3     AX=0.
4     DO 22 JJ=1,JC
      KS=1
5     AAY=0.
      DO 19 K=JE,JED
6     IF(KS) 8,8,7
7     M=1
      J=K-LD(JB+1)
      GO TO 9
8     J=K-1
9     IF(JT(K)+JT(J)) 18,18,28
28    IF((U(K)-POTEN)*(U(J)-POTEN)) 10,10,18
10    DIF=ABSF(U(J)-U(K))
      IF(DIF) 13,13,11
11    IF(M) 12,14,12
12    VX(L)=ABSF(U(J)-POTEN)/DIF*DX+AX
      VY(L)=AAY
      GO TO 15
13    VX(L)=AX+DX
      VY(L)=AAY
      GO TO 15
14    VX(L)=AX+DX
      VY(L)= ABSF(U(J)-POTEN)/DIF*DX+AAY
15    IF(L-7) 17,16,16
16    WRITE OUTPUT TAPE 6,100,POTEN,(VX(I),VY(I),I=1,7)
      L=0
17    L=L+1
18    AAY=AAY+DX
19    CONTINUE
      IF(KS) 21,21,20
20    KS=0
      M=0
```

```
      JE=JE+1
      GO TO 5
21    JE=JE+JD
      JED=JE+JD
      BX=BX+DX
      AX=AX+DX
22    CONTINUE
      IF(L-2) 25,23,23
23    DO 24 J=L,7
      VX(J)=0.
24    VY(J)=0.
      WRITE OUTPUT TAPE 6,100,POTEN,(VX(I),VY(I),I=1,7)
25    POTEN=POTEN-SIZE
      IF(POTEN-VBT) 27,26,26
26    AX=AX-BX
      GO TO 2
27    RETURN
100   FORMAT(17HOPOTENTIAL (X,Y) F8.1,2H  7(2H (F5.3,1H,F5.3,2H) ))
      END
```

```
      SUBROUTINE TWOUT(KK)
C TWOUT PRINTS OUT THE POTENTIAL FIELD
      COMMON URH,UB,JT,RH,U,XT,KT,LINC,LB,XR,NTOP,NUL,NLIN,NREG,NTP,
     1 NPIT,XEP,NXEP,CU,NPONT,AX,AY,VX,VY,DX,DELY,YEP,XGM,F,KISW,ETX,
     2 ETY,PTY,PTX,NAJ,NTJ,KBA,VA,VB,VC,NCCCR,KCH,HSL,XLSL,EPS,NPOL,
     3 NPUL,JOT,KAB,NSPAN,RX,NRL,KRL,KAN,LC,LD,VAT,VBT,SIZE,KBN,RHUP,
     4 RHDOWN,XCU,NOT,MC,KCY,NSWP,ATX,ATY,KAT,KATT,KABB,NURL,NJCT,
     5 IXO,HGH,XLOW,XMPR,NEM
      DIMENSION KBN(20),LD(4),LC(10),URH(4000),UB(4000),JT(4000),
     1 RH(4000),U(4000),XT(630),KT(630),LINC(50),LB(200),CU(40),VY(40)
     2 ,VX(40),KCH(40),PTX(40),PTY(40),AY(40),AX(40),ATX(40),ATY(40),
     3 ETX(40),ETY(40),XCU(40),KCY(10)
      NN=KK
      WRITE OUTPUT TAPE 6,102,NN
      XCON=XEP*XMPR
      WRITE OUTPUT TAPE 6,101,NXEP,XCON,EPS
      IF(MO) 10,3,10
10    K=1
      DO 1 J=1,NTOP
      KT(K)=J
      XT(K)=U(J)
      K=K+1
      IF(K-9) 1,2,2
2     WRITE OUTPUT TAPE 6,100,(KT(L),L=1,8),(XT(L),L=1,8)
      K=1
1     CONTINUE
      IF(K-2) 8,6,6
6     DO 7 J=K,8
      KT(J)=0
7     XT(J)=0.
      WRITE OUTPUT TAPE 6,100,(KT(L),L=1,8),(XT(L),L=1,8)
8     RETURN
3     WRITE OUTPUT TAPE 6,104
      DO 11 J=1,NTOP
11    U(J)=.5*(U(J)+UB(J))
      JOT=NJOT
      NURL=-7777
      GO TO 10
100   FORMAT(1H 8I5,8F10.3)
101   FORMAT(21HOPOINT/ERROR/EPSILON     I5,2F15.6)
102   FORMAT(13HOL-ITERATION    I3)
104   FORMAT(23HOL-FIELD IS AVERAGED.        )
      END
```

```
      SUBROUTINE MNTRI
C MNTRI COORDINATES THE RHS AND TRAJECTORY CALCULATION
      COMMON URH,UB,JT,RH,U,XT,KT,LINC,LB,XR,NTOP,NUL,NLIN,NREC,NTP,
     1 NPIT,XEP,NXEP,CU,NPONT,AX,AY,VX,VY,DX,DELY,YEP,XCM,F,KISW,ETX,
     2 ETY,PTY,PTX,NAJ,NTJ,KBA,VA,VB,VC,NCCCR,KCH,HSL,XLSL,EPS,NPOL,
     3 NPUL,JOT,KAB,NSPAN,RX,NRL,KRL,KAN,LC,LD,VAT,VBT,SIZE,KBN,RHUP,
     4 RHDOWN,XCU,NOT,MO,KCY,NSWP,ATX,ATY,KAT,KATT,KABB,NURL,NJCT,
     5 IXO,HGH,XLOW,XMPR,NEM
      DIMENSION KBN(20),LD(4),LC(10),URH(4000),UB(4000),JT(4000),
     1 RH(4000),U(4000),XT(630),KT(630),LINC(50),LB(200),CU(40),VY(40)
     2 ,VX(40),KCH(40),PTX(40),PTY(40),AY(40),AX(40),ATX(40),ATY(40),
     3 ETX(40),ETY(40),XCU(40),KCY(10)
      NPIT=0
 301  DO 1 J=1,40
      PTX(J)=0.
      PTY(J)=0.
      VY(J)=.000001
      VX(J)=0.
      CU(J)=0.
 1    KCH(J)=-1
      DO 35 J=1,NTOP
      UB(J)=U(J)
 35   URH(J)=RH(J)
      NOT=JOT
      NAJ=NTJ-1
      IF(NCOOR) 2,2,3
C ARC CALCULATES THE EMITTER COORDINATES IF NECESSARY
 2    CALL ARC
      NCOOR=1
 3    IF(KISW) 13,13,12
C PEQ CALCULATES THE EQUIPOTENTIAL LINE FOR CU CALCULATION
 13   CALL PEQ
 12   DO 11 J=1,NTJ
      AX(J)=ETX(J)
 11   AY(J)=ETY(J)
      DX=LC(1)
      DX=DX*H
      DELY=DX
      JC=LC(2)
      JCX=LC(3)
      JE=LC(4)
      JEX=LC(5)
      JD=LC(6)
      JDX=LC(7)
      RM=DX/YEP
      AAX=0.
      NS=0
      JED=JD
      DO 22 JN=1,JC
      DO 7 J=1,NTJ
 7    AY(J)=AY(J)+DX
      IF(NS) 4,4,5
C CURRNT CALCULATES THE CURRENT PER UNIT LENGTH
 4    CALL CURRNT
      NS=1
C TRAJY CALCULATES THE TRAJECTORY COORDINATES IN THE Y SWEEP
```

```
 5        CALL TRAJY(JE,JED)
          IF(NOT)6,6,17
 17       NOT=NOT-1
          WRITE OUTPUT TAPE 6,100,NOT,AY(1),(K,AX(K),VX(K),VY(K),K=1,NTJ)
 6        DO 203 K=JE,JED,JDX
 203      RH(K)=0.
C CALRY CALCULATES THE RHS IN THE Y SWEEP
          CALL CALRY(JE,JED)
          DO 21 K=JE,JED ,JDX
 21       RH(K)=RH(K)*RM
          BIG=0.
          SML=1.
          DO 34 J=1,NTJ
          S=ABSF(VY(J)/VX(J))
          IF(S-BIG) 33,33,36
 36       BIG=S
 33       IF(SML-S) 34,34,19
 19       SML=S
 34       CONTINUE
C SLOPE TEST FOR CHANGE OF SWEEP
          IF(BIG-HSL) 55,55,18
 18       IF(SML-XLSL) 55,55,23
 23       JED=JED+1
 22       JE=JE+1
          WRITE OUTPUT TAPE 6,104
          GO TO 307
 55       JX=(AY+.0001)/DX
          NOT=0
          DO 209 J=1,NAJ
 209      CU(J)=-CU(J)
          JE=JEX+JX
          DO 200 JN=1,JCX
          JED=JEX+JDX-1
          AAX=AAX+DX
C TRAJX PICKS UP THE TRAJECTORIES IN THE X SWEEP
          CALL TRAJX(JEX,JDX,AAX)
C CORRCT CONDITIONALLY TERMINATES OR SHIFTS THE TRAJECTORIES
          CALL CORRCT(AAX,JN)
          IF(NOT) 8,8,9
 9        NOT=NOT-1
          WRITE OUTPUT TAPE 6,101,NOT,AAX,(K,AY(K),VX(K),VY(K),K=1,NTJ)
 8        JR=JE+1
          DO 202 K=JR,JED
 202      RH(K)=0.
C CALRX CALCULATES THE RHS IN THE X SWEEP
          CALL CALRX(JEX,JED,JE)
          DO 201 K=JR,JED
 201      RH(K)=RH(K)*RM
          JE=JE+JDX
 200      JEX=JEX+JDX
 307      JOT=0
          IWRL=NRL-KRL+1
          IF(KOY(IWRL)) 304,304,305
 305      WRITE OUTPUT TAPE 6,102,(J,XCL(J),J=1,NAJ)
          SUM=0.
          DO 10 J=1,NAJ
```

```
 1C       SUM=SUM+XCU(J)
          WRITE OUTPUT TAPE 6,1C3,SUM
 304      IF(NURL) 303,3C3,3C2
C RTEST CHECKS THE UPPER AND LOWER BOUNDS ON RHS
 302      CALL RTEST
C UCAL SOLVES THE MATRIX EQUATION
          CALL UCAL
          GO TO 301
 303      RETURN
 100      FORMAT(I5,F1C.5,(1CH X VX VY   I5,3E15.6))
 101      FORMAT(I5,F1C.5,(1CH Y VX VY   I5,3E15.6))
 102      FORMAT(17HOINITIAL CURRENTS   //(7(1H ,I2,E14.6)))
 103      FORMAT(8HCTOTAL  E14.6)
 104      FORMAT(24HO TEST ON SLOPES UNMET.    )
          END
```

```
      SUBROUTINE ARC
C ARC CALCULATES THE BEGINNING TRAJECTORY COORDINATES
      COMMON URH,UB,JT,RH,U,XT,KT,LINC,LB,XR,NTCP,NUL,NLIN,NREG,NTP,
     1 NPIT,XEP,NXEP,CL,NPCNT,AX,AY,VX,VY,DX,DELY,YEP,XCM,F,KISW,ETX,
     2 ETY,PTY,PTX,NAJ,NTJ,KBA,VA,VB,VC,NCCCR,KCH,HSL,XLSL,EPS,NPOL,
     3 NPUL,JOT,KAB,NSPAN,RX,NRL,KRL,KAN,LC,LD,VAT,VBT,SIZE,KBN,RHUP,
     4 RHDOWN,XCU,NOT,MO,KCY,NSWP,ATX,ATY,KAT,KATT,KABB,NURL,NJCT,
     5 IXO,HGH,XLOW,XMPR,NEM
      DIMENSION KBN(20),LD(4),LC(10),URH(4000),UB(4000),JT(4000),
     1 RH(4000),U(4000),XT(630),KT(630),LINC(50),LB(200),CU(40),VY(40)
     2 ,VX(40),KCH(40),PTX(40),PTY(40),AY(40),AX(40),ATX(40),ATY(40),
     3 ETX(40),ETY(40),XCU(40),KCY(10)
      SUM=0.
      DO 1 J=2,NEM
1     SUM=SUM+SQRTF((ATX(J)-ATX(J-1))**2+(ATY(J)-ATY(J-1))**2)
      ARCL=NTJ-1
      ARCL=SUM/ARCL
      K=1
      REM=0.
      J=1
      HYP=ARCL
      ETX(K)=ATX(K)
      ETY(K)=ATY(K)
      K=K+1
2     XDIF=ATX(J)-ATX(J+1)
      YDIF=ATY(J+1)-ATY(J)
      SA=SQRTF(XDIF**2+YDIF**2)
      CSA=YDIF/SA
      SNA=XDIF/SA
3     ETX(K)=ATX(J)-HYP*SNA
      ETY(K)=ATY(J)+HYP*CSA
      HYP=ARCL+HYP
      K=K+1
      REM=REM+ARCL
      IF(K-NTJ) 4,6,6
4     IF(SA-REM-ARCL) 5,3,3
5     J=J+1
      HYP=ARCL-SA+REM
      REM=HYP-ARCL
      GO TO 2
6     CONTINUE
      NEM=NEM
      ETX(K)=ATX(NEM)
      ETY(K)=ATY(NEM)
      IF(JOT) 8,8,7
7     WRITE OUTPUT TAPE 6,103,SUM,ARCL
      WRITE OUTPUT TAPE 6,100,(J,ATX(J),ATY(J),J=1,NEM)
      WRITE OUTPUT TAPE 6,102,(J,ETX(J),ETY(J),J=1,NTJ)
8     CONTINUE
100   FORMAT (12H0X,Y-EMITTER//(7(1H ,I2,2H (F5.3,1H,F5.3,2H)  )))
102   FORMAT (16H0X,Y-BEGIN TRAJ.//(7(1H ,I2,2H (F5.3,1H,F5.3,2H) )))
103   FORMAT (30H ARC,DELTA ARC LENGTH   2F10.5)
      RETURN
      END
```

```
      SUBROUTINE PEQ
C PEQ CALCULATES THE EQUIPOTENTIAL LINE FOR THE CURRENT DENSITY
C CALCULATION
      COMMON URH,UB,JT,RH,U,XT,KT,LINC,LB,XR,NTCP,NUL,NLIN,NREC,NTP,
     1 NPIT,XEP,NXEP,CL,NPCNT,AX,AY,VX,VY,DX,DELY,YEP,XCM,H,KISW,ETX,
     2 ETY,PTY,PTX,NAJ,NTJ,KBA,VA,VB,VC,NCCCR,KCH,HSL,XLSL,EPS,NPOL,
     3 NPUL,JOT,KAB,NSPAN,RX,NRL,KRL,KAN,LC,LD,VAT,VBT,SIZE,KBN,RHUP,
     4 RHDOWN,XCU,NOT,MC,KCY,NSWP,ATX,ATY,KAT,KATT,KABB,NURL,NJCT,
     5 IXO,HGH,XLOW,XMPR,NEM
      DIMENSION KBN(20),LD(4),LC(10),URH(4000),UB(4000),JT(4000),
     1 RH(4000),U(4000),XT(630),KT(630),LINC(50),LB(200),CU(40),VY(40)
     2 ,VX(40),KCH(40),PTX(40),PTY(40),AY(40),AX(40),ATX(40),ATY(40),
     3 ETX(40),ETY(40),XCU(40),KCY(10)
      KBA=KBA
      POTEN=U(KBA)
      DX=H
      L=0
      KAB=KAB
      IF(NSWP) 32,24,32
32    AAY=0.
      JE=LC(4)
      JD=LC(6)
      DO 8 JJ=1,KAB
      AAX=0.
      JED=JE+NSPAN
      DO 7 K=JE,JED
      J=K-1
      IF((U(K)-POTEN)*(U(J)-POTEN)) 1,1,7
1     DIF=ABSF(U(J)-U(K))
      L=L+1
      PTX(L)=AAY
      IF(DIF) 2,6,2
2     PTY(L)=ABSF(U(J)-POTEN)/DIF*DX+AAX
      GO TO 7
6     PTY(L)=AAX+DX
7     AAX=AAX+DX
      AAY=AAY+DX
8     JE=JE+JD
      DO 11 J=1,L
      LL=L-J+1
      T=0.
      DO 10 I=1,LL
      IF(T-PTX(I)) 9,9,10
9     T=PTX(I)
      NN=I
10    CONTINUE
      PP=PTX(LL)
      PTX(LL)=T
      PTX(NN)=PP
      PP=PTY(LL)
      PTY(LL)=PTY(NN)
      PTY(NN)=PP
11    CONTINUE
      DO 14 J=2,L
      IF(PTX(J)-PTX(J-1)) 14,12,14
12    NN=L-1
```

```
      DO 13 JJ=J,NN
      PTX(JJ)=PTX(JJ+1)
13    PTY(JJ)=PTY(JJ+1)
14    CONTINUE
50    IF(JOT) 60,60,70
70    WRITE OUTPUT TAPE 6,100,(J,PTX(J),PTY(J),J=1,L)
60    RETURN
24    AAX=0.
      JE=LC(5)
      JD=LC(6)
      DO 28 JJ=1,KAB
      AAY=0.
      JED=JE+JD-1
      DO 27 K=JE,JED
      J=K-JD
      IF((U(K)-POTEN)*(U(J)-POTEN)) 21,21,27
21    DIF=ABSF(U(J)-U(K))
      L=L+1
      PTY(L)=AAY
      IF(DIF) 22,26,22
22    PTX(L)=ABSF(U(J)-POTEN)/DIF*DX+AAX
      GO TO 27
26    PTX(L)=AAX+DX
27    AAY=AAY+DX
      AAX=AAX+DX
28    JE=JE+JD
      DO 41 J=1,L
      LL=L-J+1
      T=0.
      JJ=LL
      DO 40 I=1,LL
      IF(T-PTY(I)) 39,39,40
39    T=PTY(I)
      NN=I
40    CONTINUE
      PP=PTY(JJ)
      PTY(JJ)=T
      PTY(NN)=PP
      PP=PTX(JJ)
      PTX(JJ)=PTX(NN)
      PTX(NN)=PP
41    CONTINUE
      DO 44 J=2,L
      IF(PTY(J)-PTY(J-1)) 44,42,44
42    NN=L-1
      DO 43 JJ=J,NN
      PTX(JJ)=PTX(JJ+1)
43    PTY(JJ)=PTY(JJ+1)
44    CONTINUE
      GO TO 50
100 FORMAT(18HCX,Y-EQUIPOTENTIAL//,(7(1H ,I2,2H (F5.3,1H,F5.3,2H) )))
      END
```

```
      SUBROUTINE CURRNT
C CURRNT CALCULATES THE CURRENT PER UNIT LENGTH
      COMMON URH,UB,JT,RH,L,XT,KT,LINC,LB,XR,NTCP,NUL,NLIN,NREC,NTP,
     1 NPIT,XEP,NXEP,CL,NPCNT,AX,AY,VX,VY,DX,DELY,YEP,XQM,H,KISW,ETX,
     2 ETY,PTY,PTX,NAJ,NTJ,KBA,VA,VB,VC,NCCCR,KCH,HSL,XLSL,EPS,NPOL,
     3 NPUL,JOT,KAB,NSPAN,RX,NRL,KRL,KAN,LC,LD,VAT,VBT,SIZE,KBN,RFUP,
     4 RFDOWN,XCU,NOT,MC,KCY,NSWP,ATX,ATY,KAT,KATT,KABB,NURL,NJCT,
     5 IXO,HGH,XLOW,XMPR,NEM
      DIMENSION KBN(20),LD(4),LC(10),URH(4000),UB(4000),JT(4000),
     1 RH(4000),U(4000),XT(630),KT(630),LINC(50),LB(200),CU(40),VY(40)
     2 ,VX(40),KCH(40),PTX(40),PTY(40),AY(40),AX(40),ATX(40),ATY(40),
     3 ETX(40),ETY(40),XCU(40),KCY(10)
      DIMENSION HX(20)
      XK=2./9.*YEP*SQRTF(2.*XQM)*H
      IF(KISW) 20,20,30
 20   DO 10 J=1,30
      KBA=KBA
      DELU=VA-U(KBA)
      XK=XK*DELU*SQRTF(DELU)
      N=J
      K=(PTX(J)+.0001)/ETX
      IF(K) 10,10,11
 10   CONTINUE
 11   J=1
      L=1
 12   XD=PTY(N)-ETY(J)
      YD=PTY(N+1)-ETY(J+2)
      XMID=.5*(XD+YD)
      UP= .5*(XMID+XD)
      DN= .5*(XMID+YD)
      HX(L)=UP
      HX(L+1)=DN
      L=L+2
      J=J+2
      N=N+1
      IF(L-NAJ) 12,13,13
 13   DO 14 J=1,NAJ
 14   CU(J)=      XK/HX(J)**2
 40   DO 50 J=1,NAJ
 50   XCU(J)=CU(J)
      RETURN
 30   K=NPOL
      JD=NPUNT
      XK=XK/H**2
      J=1
 39   JOB=K+JD
      UA=.5*(U(K)+U(JOB ))
      UU=.5*(UA+U(K))
      UA=.5*(UA+U(JOB ))
      DELU=VA-UU
      CU(J)=XK*DELU*SQRTF(DELU)
      DELU=VA-UA
      CU(J+1)=XK*DELU*SQRTF(DELU)
      J=J+2
      K=K+JD
      IF(J-NAJ) 39,40,40
      END
```

```
      SUBROUTINE TRAJY(KE,KED)
C TRAJY CALCULATES THE COORDINATES IN THE Y SWEEP
      COMMON URH,UB,JT,RH,U,XT,KT,LINC,LB,XR,NTCP,NUL,NLIN,NREG,NTP,
     1 NPIT,XEP,NXEP,CL,NPCNT,AX,AY,VX,VY,DX,DELY,YEF,XCM,F,KISW,ETX,
     2 ETY,PTY,PTX,NAJ,NTJ,KBA,VA,VB,VC,NCCCR,KCH,HSL,XLSL,EPS,NPOL,
     3 NPUL,JOT,KAB,NSPAN,RX,NRL,KRL,KAN,LC,LD,VAT,VBT,SIZE,KBN,RHUP,
     4 RHDOWN,XCU,NOT,MC,KCY,NSWP,ATX,ATY,KAT,KATT,KABB,NURL,NJCT,
     5 IXO,HGH,XLOW,XMPR,NEM
      DIMENSION KBN(20),LD(4),LC(10),URH(4000),UB(4000),JT(4000),
     1 RH(4000),U(4000),XT(630),KT(630),LINC(50),LB(200),CU(40),VY(40)
     2 ,VX(40),KCH(40),PTX(40),PTY(40),AY(40),AX(40),ATX(40),ATY(40),
     3 ETX(40),ETY(40),XCU(40),KCY(10)
      JE=KE
      JED=KED
      JD=NPONT
      DO 40 K=1,NTJ
41    AD=AX(K)/H
      JX=AD
      XA=JX
      XA=AD-XA
      JP=JE+JX*JD
      JS=4
      JQ=JP-1
      JOP=JQ+JD
      UL=(1.-XA)*U(JQ)+XA*U(JOP)
      JAP=JP+JD
      UK=(1.-XA)*U(JP)+XA*U(JAP)
      IF(XA-.5) 1,1,5
5     XA=XA-1.
      JX=JX+1
      JQ=JQ+13
1     JOP=JQ+JD
      JAP=JQ-JD
      YLA=(XA+.5)*U(JOP  )-2.*XA*U(JQ)+(XA-.5)*U(JAP)
      DY=VX(K)/VY(K)
      JOX=JX
7     XB=XA+DY
      IF(DY) 9,9,10
8     XB=1.+XB
      JX=JX-1
9     IF(XB+.5) 8,12,12
11    XB=XB-1.
      JX=JX+1
10    IF(XB-.5) 12,12,11
12    JP=JE+JX*JD
      JOP=JP+JD
      JAP=JP-JD
      YRA=(XB+.5)*U(JOP  )-2.*XB*U(JP)+(XB-.5)*U(JAP)
      IF(XB) 19,20,20
19    XB=1.-ABSF(XB)
      JP=JP-13
20    JQ=JP-1
      JOP=JQ+JD
      JAP=JP+JD
      USN=(1.-XB)*U(JQ)+XB*U(JOP)
      UQN=(1.-XB)*U(JP)+XB*U(JAP)
```

```
      DUX=.5*(UK-UL-USN+UQN)
      VYB=SQRTF(VY(K)**2-2.*XQM*DUX)
      DT=2.*DX/(VYB+VY(K))
      JS=JS-1
      YA=.5*XQM*(YLA+YRA)/DX
      DY=DT*(VX(K)-.5*YA*DT)/DX
      JX=JOX
      IF(JS) 21,21,7
21    VY(K)=VYB
      VX(K)=VX(K)-YA*DT
      AX(K)=AX(K)+DY*DX
40    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CALRY(KE,KED)
C CALRY CALCULATES THE RHS IN THE Y SWEEP
      COMMON URH,UB,JT,RH,U,XT,KT,LINC,LB,XR,NTCP,NUL,NLIN,NREG,NTP,
     1 NPIT,XEP,NXEP,CL,NPCNT,AX,AY,VX,VY,DX,DELY,YEP,XCM,F,KISW,ETX,
     2 ETY,PTY,PTX,NAJ,NTJ,KBA,VA,VB,VC,NCLCR,KCH,HSL,XLSL,EPS,NPOL,
     3 NPUL,JOT,KAB,NSPAN,RX,NRL,KRL,KAN,LC,LD,VAT,VBT,SIZE,KBN,RHUP,
     4 RHDOWN,XCU,NOT,NC,KCY,NSWP,ATX,ATY,KAT,KATT,KABB,NURL,NJCT,
     5 IXO,FGH,XLOW,XMPR,NEM
      DIMENSION KBN(20),LD(4),LC(10),URH(4000),UB(4000),JT(4000),
     1 RH(4000),U(4000),XT(630),KT(630),LINC(50),LB(200),CU(40),VY(40)
     2 ,VX(40),KCH(40),PTX(40),PTY(40),AY(40),AX(40),ATX(40),ATY(40),
     3 ETX(40),ETY(40),XCL(40),KCY(10)
      JE=KE
      JED=KED
      DO 33 JJ=1,NAJ
      J=JJ
1     HT=AX(J+1)-AX(J)
      IF(HT) 12,12,13
12    HT=-HT
      XH=AX(J+1)
      JX=XH/DELY
      NB=AX(J)/DELY+1.
      WA=SQRTF(VX(J+1)**2+VY(J+1)**2)
      WB=SQRTF(VY(J)**2+VY(J)**2)
      GO TO 14
13    XH=AX(J)
      JX=XH/DELY
      WA=SQRTF(VX(J)**2+VY(J)**2)
      WB=SQRTF(VX(J+1)**2+VY(J+1)**2)
      NB=AX(J+1)/DELY+1.
14    YL=XH+HT
      DO 32 KK=JX,NB
      K=JE+KK*NPONT
      XA=KK
      XUD=(XA-.5)*DELY
      XX=XUD+DELY
      YU=MAX1F(XUD,XH)
      YD=MIN1F(XX,YL)
      IF(YD-YL) 32,32,27
27    XA=.5*(YD+YL)
      W=WA+(XA-XH)*(WB-WA)/HT
      RH(K)=RH(K)+(YD-YL)*CL(JJ)/(HT*W)
32    CONTINUE
33    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TRAJX(KE,KED,BX)
C TRAJX PICKS UP THE TRAJECTORIES IN THE X SWEEP
      COMMON URH,UB,JT,RH,U,XT,KT,LINC,LB,XR,NTCP,NUL,NLIN,NREC,NTP,
     1 NPIT,XEP,NXEP,CL,NPONT,AX,AY,VX,VY,DX,DELY,YEP,XCM,F,KISW,ETX,
     2 ETY,PTY,PTX,NAJ,NTJ,KBA,VA,VB,VC,NCCCR,KCH,HSL,XLSL,EPS,NPOL,
     3 NPUL,JOT,KAB,NSPAN,RX,NRL,KRL,KAN,LC,LD,VAT,VBT,SIZE,KBN,RHUP,
     4 RHDOWN,XCL,NOT,MC,KCY,NSWP,ATX,ATY,KAT,KATT,KABB,NURL,NJCT,
     5 IXO,HGH,XLOW,XMPR,NEM
      DIMENSION KBN(20),LD(4),LC(10),URH(4000),UB(4000),JT(4000),
     1 RH(4000),U(4000),XT(630),KT(630),LINC(50),LB(200),CU(40),VY(40)
     2 ,VX(40),KCH(40),PTX(40),PTY(40),AY(40),AX(40),ATX(40),ATY(40),
     3 ETX(40),ETY(40),XCL(40),KCY(10)
      JE=KE
      JED=KED
      AAX=BX
      DO 10 J=1,NTJ
      IF(AX(J)+1.) 10,10,1
1     IF(KCH(J)) 2,11,11
2     IF(AAX-AX(J)) 10,10,3
3     IF(J-1) 4,4,5
5     CU(J-1) =-CU(J-1)
4     JX=(AY(J)+.001)/DX
      IF(NOT) 40,40,41
40    IF(JOT) 41,41,42
42    NOT=JOT
41    JP=JE+JX
      JQ=JP-NPONT
      JS=4
      XA=AAX-AX(J)
      UL=(1.-XA)*U(JP)+XA*U(JQ)
      UK=U(JP)
      UA=(1.-XA)*U(JP-1)+XA*U(JQ-1)
      UD=(1.-XA)*U(JP+1)+XA*U(JQ+1)
      YLA=.5*UD-.5*UA
      DY=VY(J)/VX(J)
      JOX=JX
7     XB=DY
      GO TO 30
31    XB=XB-1.
      JX=JX+1
30    IF(XB-.5) 12,12,31
12    JP=JE+JX
      YRA=(XB+.5)*U(JP+1)-2.*XB*U(JP)+(XB-.5)*U(JP-1)
      IF(XB) 19,19,20
19    JP=JP-1
      XB=1.-ABSF(XB)
20    JQ=JP-NPONT
      UQN=(1.-XB)*U(JP)+XB*U(JP+1)
      USN=(1.-XB)*U(JQ)+XB*U(JQ+1)
      USN=(1.-XA)*UQN+XA*USN
      DUX=.5*(UK-UL+UQN-USN)
      VXB=SQRTF(VX(J)**2-2.*XQM*DUX)
      DT=2.*XA/(VXB+VX(J))
      JS=JS-1
      YA=.5*XQM/DX*(YLA+YRA)
      DY=DT*(VY(J)-.5*YA*DT)/DX
```

```
      JX=JOX
      IF(JS) 21,21,7
 21   VX(J)=VXB
      VY(J)=VY(J)-YA*DT
      AY(J)=AY(J)+DY*DX
      AX(J)=AAX
      KCH(J)=C.
      GO TO 1C
 11   AX(J)=AX(J)+DX
C TRAJ CALCULATES THE TRAJECTORY COORDINATES IN THE X SWEEP
      CALL TRAJ(JE,JED,J)
 1C   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TRAJ(KE,KED,M)
C TRAJ CALCULATES THE TRAJECTORY COORDINATES IN THE X SWEEP
      COMMON URH,UB,JT,RH,U,XT,KT,LINC,LB,XR,NTCP,NUL,NLIN,NREG,NTP,
     1 NPIT,XEP,NXEP,CL,NPCNT,AX,AY,VX,VY,DX,DELY,YEP,XQM,F,KISW,ETX,
     2 ETY,PTY,PTX,NAJ,NTJ,KBA,VA,VB,VC,NCCCR,KCH,HSL,XLSL,EPS,NPOL,
     3 NPUL,JOT,KAB,NSPAN,RX,NRL,KRL,KAN,LC,LD,VAT,VBT,SIZE,KBN,RHUP,
     4 RHDOWN,XCU,NOT,MO,KCY,NSWP,ATX,ATY,KAT,KATT,KABB,NURL,NJOT,
     5 IXO,HGH,XLOW,XMPR,NEM
      DIMENSION KBN(20),LD(4),LC(10),URH(4000),UB(4000),JT(4000),
     1 RH(4000),U(4000),XT(630),KT(630),LINC(50),LB(200),CU(40),VY(40)
     2 ,VX(40),KCH(40),PTX(40),PTY(40),AY(40),AX(40),ATX(40),ATY(40),
     3 ETX(40),ETY(40),XCL(40),KCY(10)
      JE=KE
      JED=KED
      K=M
C     TRAJ CALCULATES COORDINATES FOR X-SWEEP
      AD=AY(K)/DX
      JX=AD
      XA=JX
      XA=AD-XA
      JP=JX+JE
      JS=4
      JQ=JP-NPONT
      UL=(1.-XA)*U(JQ)+XA*U(JQ+1)
      UK=(1.-XA)*U(JP)+XA*U(JP+1)
      IF(XA-.5) 1,1,5
 5    XA=XA-1.
      JQ=JQ+1
      JX=JX+1
 1    YLA=(XA+.5)*U(JQ+1)-2.*XA*U(JQ)+(XA-.5)*U(JQ-1)
      DY=VY(K)/VX(K)
      JOX=JX
 7    XB=XA+DY
      IF(DY) 9,9,10
 8    XB=XB+1.
      JX=JX-1
 9    IF(XB+.5) 8,12,12
 11   XB=XB-1.
      JX=JX+1
 10   IF(XB-.5) 12,12,11
 12   JP=JE+JX
      YRA=(XB+.5)*U(JP+1)-2.*XB*U(JP)+(XB-.5)*U(JP-1)
      IF(XB) 19,20,20
 19   JP=JP-1
      XB=1.-ABSF(XB)
 20   JQ=JP-NPONT
      USN=(1.-XB)*U(JQ)+XB*U(JQ+1)
      UQN=(1.-XB)*U(JP)+XB*U(JP+1)
      DUX=.5*(UK-UL-USN+UQN)
      VXB=SQRTF(VX(K)**2-2.*XQM*DUX)
      DT=2.*DX/(VXB+VX(K))
      JS=JS-1
      YA=.5*XQM*(YLA+YRA)/DX
      DY=DT*(VY(K)-.5*YA*DT)/DX
      JX=JOX
      IF(JS) 21,21,7
```

```
21      VX(K)=VXB
        VY(K)=VY(K)-YA*DT
        AY(K)=AY(K)+DY*DX
        RETURN
        END
```

```
      SUBROUTINE CALRX(KE,KED,KD)
C CALRX CALCULATES THE RHS IN THE X SWEEP
      COMMON LRH,LB,JT,RH,L,XT,KT,LINC,LB,XR,NTCP,NUL,NLIN,NREG,NTP,
     1 NPIT,XEP,NXEP,CL,NPCNT,AX,AY,VX,VY,DX,DELY,YEP,XCN,F,KISW,ETX,
     2 ETY,PTY,PTX,NAJ,NTJ,KBA,VA,VB,VC,NCCCR,KCH,HSL,XLSL,EPS,NPCL,
     3 NPUL,JOT,KAB,NSPAN,RX,NRL,KRL,KAN,LC,LD,VAT,VBT,SIZE,KBN,RHUP,
     4 RHDOWN,XCU,NOT,MC,KCY,NSWP,ATX,ATY,KAT,KATT,KABB,NURL,NJCT,
     5 IXO,FGH,XLOW,XMPR,NEM
      DIMENSION KBN(20),LD(4),LC(10),LRH(4000),UB(4000),JT(4000),
     1 RF(4000),L(4000),XT(630),KT(630),LINC(50),LB(200),CU(40),VY(40)
     2 ,VX(40),KCH(40),PTX(40),PTY(40),AY(40),AX(40),ATX(40),ATY(40),
     3 ETX(40),ETY(40),XCL(40),KCY(10)
      JAD=KD
      JE=KE
      JED=KED
      DO 33 JJ=1,NAJ
      J=JJ
      IF(CL(JJ)) 33,33,1
1     HT=AY(J+1)-AY(J)
      IF(HT) 12,12,13
12    HT=-HT
      XH=AY(J+1)
      JX=XH/DELY
      NB=AY(J)/DELY+1.
      WA=SQRTF(VX(J+1)**2+VY(J+1)**2)
      WB=SQRTF(VX(J)**2+VY(J)**2)
      GO TO 14
13    XH=AY(J)
      JX=XH/DELY
      WA=SQRTF(VX(J)**2+VY(J)**2)
      WB=SQRTF(VX(J+1)**2+VY(J+1)**2)
      NB=AY(J+1)/DELY+1.
14    NA=JX+JE
      NB=NB+JE
      YL=XH+HT
      DO 32 K=NA,NB
      IF(K-JAD) 32,32,10
10    XA=K-JE
      XUD=(XA-.5)*DELY
      XX=XUD+DELY
      YU=MAX1F(XUD,XH)
      YD=MIN1F(XX,YL)
      IF(YD-YU) 32,32,27
27    XA=.5*(YD+YU)
      W=WA+(XA-XH)*(WB-WA)/HT
      RF(K)=RF(K)+(YD-YU)*CL(JJ)/(HT*W)
32    CONTINUE
33    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CORRCT(BAX,M)
C CORRCT CONDITIONALLY ENDS OR SHIFTS THE TRAJECTORIES
      COMMON URH,UB,JT,RH,U,XT,KT,LINC,LB,XR,NTCP,NUL,NLIN,NREC,NTP,
     1 NPIT,XEP,NXEP,CL,NPCNT,AX,AY,VX,VY,DX,DELY,YEP,XCM,F,KISW,ETX,
     2 ETY,PTY,PTX,NAJ,NTJ,KBA,VA,VB,VC,NCECR,KCH,HSL,XLSL,EPS,NPOL,
     3 NPUL,JOT,KAB,NSPAN,RX,NRL,KRL,KAN,LC,LD,VAT,VBT,SIZE,KBN,RHUP,
     4 RHDOWN,XCU,NOT,MC,KCY,NSWP,ATX,ATY,KAT,KATT,KABB,NURL,NJCT,
     5 IXO,HGH,XLOW,XMPR,NEM
      DIMENSION KBN(20),LD(4),LC(10),URH(4000),UB(4000),JT(4000),
     1 RH(4000),U(4000),XT(630),KT(630),LINC(50),LB(200),CU(40),VY(40)
     2 ,VX(40),KCH(40),PTX(40),PTY(40),AY(40),AX(40),ATX(40),ATY(40),
     3 ETX(40),ETY(40),XCU(40),KCY(10)
      JN=M
      AAX=BAX
      IF(JN-12) 10,1,12
12    IF(JN-13) 10,13,10
1     DO 30 J=1,NTJ
      IF(AX(J)+1.) 30,30,3
3     IF( XLOW-AY(J)) 4,30,30
4     AYD=AY(J)-AY(J+1)
      AY(J)= XLOW
      IF(AY(J)-AY(J+1)) 5,5,6
5     AY(J)=-1.
      AX(J)=-1.
      CU(J)=0.
      GO TO 30
6     AYDS=AY(J)-AY(J+1)
      AYDL=AYD-AYDS
      CU(J)=CU(J)*AYDS/AYD
      VY(J)=(VY(J)*AYDS+VY(J+1)*AYDL)/AYD
      VX(J)=(VX(J)*AYDS+VX(J+1)*AYDL)/AYD
30    CONTINUE
10    RETURN
13    DO 40  J=1,NTJ
      IF(AX(J)+1.) 40,40,21
21    IF(AY(J)- HGH ) 22,40,40
22    AYD=AY(J-1)-AY(J)
      AY(J)= HGH
      IF(AY(J)-AY(J-1)) 24,23,23
23    AY(J)=-1.
      AX(J)=-1.
      CU(J-1)=0.
      GO TO 40
24    AYDS=AY(J-1)-AY(J)
      AYDL=AYD-AYDS
      CU(J-1)=CU(J-1)*AYDS/AYD
      VY(J)=(VY(J)*AYDS+VY(J-1)*AYDL)/AYD
      VX(J)=(VX(J)*AYDS+VX(J-1)*AYDL)/AYD
40    CONTINUE
      GO TO 10
      END
```

```
      SUBROUTINE RTEST
      COMMON URH,UB,JT,RH,U,XT,KT,LINC,LB,XR,NTOP,NUL,NLIN,NREC,NTP,
     1 NPIT,XEP,NXEP,CU,NPCNT,AX,AY,VX,VY,DX,DELY,YEP,XCM,F,KISW,ETX,
     2 ETY,PTY,PTX,NAJ,NTJ,KBA,VA,VB,VC,NCCCR,KCH,HSL,XLSL,EPS,NPOL,
     3 NPUL,JOT,KAB,NSPAN,RX,NRL,KRL,KAN,LC,LD,VAT,VBT,SIZE,KBN,RHUP,
     4 RHDOWN,XCU,NOT,MC,KCY,NSWP,ATX,ATY,KAT,KATT,KABB,NURL,NJCT,
     5 IXO,HGH,XLOW,XMPR,NEM
      DIMENSION KBN(20),LD(4),LC(10),URH(4000),UB(4000),JT(4000),
     1 RH(4000),U(4000),XT(630),KT(630),LINC(50),LB(200),CU(40),VY(40)
     2 ,VX(40),KCH(40),PTX(40),PTY(40),AY(40),AX(40),ATX(40),ATY(40),
     3 ETX(40),ETY(40),XCU(40),KCY(10)
C RTEST CHECKS ON THE UPPER AND LOWER BOUNDS
C KAN=TEST POINT
      KAN=KAN
      RT=RH(KAN)
      SX=1.
C FIRST TWO CYCLES SET UPPER AND LOWER BOUNDS
      IF(KRL) 22,23,22
 23   IF(MO) 11,22,11
 22   IF(KRL-NRL+1) 5,2,1
C RHUP=UPPER BOUND
 1    RHUP=RT
      GO  TO 19
 2    IF(RT-RHUP) 18,18,3
C RHDOWN=LOWER BOUND
 3    RHDOWN=RHUP
      MO=-MO
      RHUP=RT
 4    GO TO 19
C MO=+1 ,CHECK IS ON UPPER BOUND
C MO=-1 ,CHECK IS ON LOWER BOUND
C MO=0 ,NO CHECK
 5    IF(MO) 10,19, 8
 6    SX=SX*RX
      DO 7 J=1,NTOP
 7    RH(J)=RH(J)*RX
 8    IF(RHUP-RH(KAN)) 6,9,9
 9    IF(RH(KAN)-RHDOWN) 11,16,16
 10   IF((RT-RHDOWN)*(RT-RHUP)) 16,16,11
 11   POW=1./SX
      DO 12 J=1,NTOP
 12   RH(J)=.5*(RH(J)*POW+URH(J))
      KRL=1
      JJ=NRL+1
      KCY(JJ)=777
      KCY(JJ+1)=777
      WRITE OUTPUT TAPE 6,100
      IF(MO) 14,13,13
 13   RHUP=RT
      GO TO 15
 14   RHDOWN=RT
 15   MO=0
 16   IF(MO) 18,19,17
 17   RHUP=RH(KAN)
      GO TO 19
 18   RHDOWN=RT
```

```
 19       XT(1)=     RHDOWN
          XT(2)=     RH(KAN)
          XT(3)=     RHUP
          WRITE OUTPUT TAPE 6,101,(XT(J),J=1,3), U(KAN)
          IF(IXO) 21,21,24
 24       IW=NRL-KRL+1
          IF(KCY(IW)) 21,21,20
C TROUT PRINTS OUT THE RHS
 20       CALL TROUT
 21       MO=-MO
          KRL=KRL-1
          RETURN
 100      FORMAT(11HORH=AVERAGE    )
 101      FORMAT(7HCRHLOW=F6.3,4H RH=F6.3,6H RHUP=F6.3,7H UTEST=F11.4)
          END
```

```
      SUBROUTINE TROUT
C TROUT PRINTS OUT THE RHS
      COMMON LRH,UB,JT,RH,U,XT,KT,LINC,LB,XR,NTOP,NUL,NLIN,NREC,NTP,
     1 NPIT,XEP,NXEP,CL,NPCNT,AX,AY,VX,VY,DX,DELY,YEP,XCM,F,KISW,ETX,
     2 ETY,PTY,PTX,NAJ,NTJ,KBA,VA,VB,VC,NCCCR,KCH,HSL,XLSL,EPS,NPOL,
     3 NPUL,JOT,KAB,NSPAN,RX,NRL,KRL,KAN,LC,LD,VAT,VBT,SIZE,KBN,RHUP,
     4 RHDOWN,XCU,NOT,MC,KCY,NSWP,ATX,ATY,KAT,KATT,KABB,NURL,NJCT,
     5 IXO,HGH,XLOW,XMPR,NEM
      DIMENSION KBN(20),LD(4),LC(10),LRH(4000),UB(4000),JT(4000),
     1 RH(4000),U(4000),XT(630),KT(630),LINC(50),LB(200),CU(40),VY(40)
     2 ,VX(40),KCH(40),PTX(40),PTY(40),AY(40),AX(40),ATX(40),ATY(40),
     3 ETX(40),ETY(40),XCL(40),KCY(10)
      IF(MO) 10,11,10
10    K=1
      DO 1 J=1,NTOP
      KT(K)=J
      XT(K)=RH(J)
      K=K+1
      IF(K-9) 1,2,2
2     WRITE OUTPUT TAPE 6,100,(KT(L),L=1,8),(XT(L),L=1,8)
      K=1
1     CONTINUE
      IF(K-2) 8,6,6
6     DO 7 J=K,8
      KT(J)=0
7     XT(J)=0.
      WRITE OUTPUT TAPE 6,100,(KT(L),L=1,8),(XT(L),L=1,8)
8     RETURN
11    WRITE OUTPUT TAPE 6,101
      K=1
      DO 21 J=1,NTOP
      KT(K)=J
      XT(K)=.5*(RH(J)+LRH(J))
      K=K+1
      IF(K-9) 21,22,22
22    WRITE OUTPUT TAPE 6,100,(KT(L),L=1,8),(XT(L),L=1,8)
      K=1
21    CONTINUE
      IF(K-2) 8,26,26
26    DO 27 J=K,8
      KT(J)=0
27    XT(J)=0.
      WRITE OUTPUT TAPE 6,100,(KT(L),L=1,8),(XT(L),L=1,8)
      GO TO 8
100   FORMAT(1H 8I5,8F10.3)
101   FORMAT(21HORH PRINT IS AVERAGE.    )
      END
```

```
* DATA
   11
1   HAMZA/AXI-SYMETRIC ION ENGINE
    MAXIMUM SLOPE TEST IS ON 1.
    MESH SIZE=.005
    EMITTER POTENTIAL =5657.2 VOLTS
    UPPER ELECTRODE=-3184.8 VOLTS
    LOWER ELECTRODE=0. VOLTS
    SUPPRESSION FACTOR =.4
    CURRENT IS CALCULATED BY EQUAL DELTA X
    INITIAL SWEEP IS IN Y-DIRECTION
    EMITTER COORD. ARE TAKEN AS THE BEGINNING TRAJ. COORD.
1
  221   13  106    1   30  200    1
   50   20   80  308  221    0
    0    1   -1    1   -1    1   -1    1   -1    1   -1    1   -1   -1
    1    1   -1   -1    1    0
    0    1   13    0   13    5   10    0   26    4   11    0   39    3
   11    0   52    2   12    0   65    2   12    0   78    2   12    0
   91    2   12    0  104    2   12    0  117    2   12    0  130    2
   12    0  143    2   12    0  156    2   10    0  156   12   12    0
  169    3    6    0  169    9   13    0  182    3    5    0  182    6
   13    0  195    6   13    0  208    1   13    0
    0  -13   13   -1    0    1
 .95        .95        .95       -.975      3.8       -.925                    9
    0  -13   13   -1    0    1
 .90        .90        .90        -.925     3.6       -.875                   15
    0  -13   13   -1    0    1
 .85        .85        .85        -.875     3.4       -.825                   21
    0  -13   13   -1    0    1
 .80        .80        .80        -.825     3.2       -.775                   27
    0  -13   13   -1    0    1
 .75        .75        .75        -.775     3.0       -.725                   33
    0  -13   13   -1    0    1
 .70        .70        .70        -.725     2.8       -.675                   39
    0  -13   13   -1    0    1
 .65        .65        .65        -.675     2.6       -.625                   45
    0  -13   13   -1    0    1
 .60        .60        .60        -.625     2.4       -.575                   51
    0  -13   13   -1    0    1
 .55        .55        .55        -.575     2.2       -.525                   57
    0  -13   13   -1    0    1
 .50        .50        .50        -.525     2.0       -.475                   63
    0  -13   13   -1    0    1
 .45        .45        .45        -.475     1.8      -.425                    69
    0  -13   13   -1    0    1
 .40        .40        .40        -.425     1.6      -.375                    75
    0  -13   13   -1    0    1
 .49375     .49375     .49375     0.        1.9625    -.975                   81
    0  -13   13   -1    0    1
 .20625     .20625     .20625    -.425      .8375     0.                      87
    0  -13   13   -1    0    1
 .65        .65        .65         .675     2.6       -.625                    93
    0  -13   13   -1    0    1
 .75        .75        .75        -.775     3.0        .725                    99
    0  -13   13   -1    0    1
```

```
.75084622 1.9729456 .99999975 .93708046 4.6810257 -.775                105
   0  -13    13   -1     0     1
.75       .95096646 .81006260 -.775      3.2609924 -.725                111
   0  -13    13   -1     0     1
.60       .90187053 .68290213 -.625      2.7847726 -.575                117
   0  -13    13   -1     0     1
.38358187 2.2193357 .61003368 -.575      4.7723274 1.3679580            123
   0  -13    13   -1     0     1
.33791599 1.1218393 .48075906 -.525      3.5244100 1.3968117            129
   0  -13    13   -1     0     1
.85963256 .85856154 .85856154 1.0136909 3.6058140 -.875                 135
   0  -13    13   -1     0     1
.72163908 .89234030 .76989569 1.8545964 4.4418322 -.925                 141
   0  -13    13   -1     0     1
.30196656 .30748522 .30748522 -.475      2.3804767 1.2905064            147
   0  -13    13   -1     0     1
.36033430 .36432314 .36432314 -.475      1.9267936 .72314733            153
   0  -13    13   -1     0     1
.40067118 .40311125 .40311125 -.475      1.8325834 .55136090            159
   0  -13    13   -1     0     1
.42863746 .42976792 .42976792 -.475      1.8067430 .47220713            165
   0  -13    13   -1     0     1
.78097865 .77732208 .77732208 1.4997761 3.9794202 -.925                 171
   0  -13    13   -1     0     1
.59066352 .58478411 .58478411 3.9264021 6.0209701 -.925                 177
   0  -13    13   -1     0     1
.52532034 .67748792 1.2721693 8.9925431 11.867200 -.925                 183
   0  -13    13   -1     0     1
.86194494 .86093254 .86093254 1.0081411 3.6050062 -.875                 189
   0  -13    13   -1     0     1
.79513451 .79256099 .79256099 1.1984555 3.6585774 -.875                 195
   0  -13    13   -1     0     1
.36875    .36875    .36875    0.       1.4625    -.725                 201
   0  -13    13   -1     0     1
.70       .70861765 .72628258 -.725      2.8349002 -.675                 207
   0  -13    13   -1     0     1
.65       .73865090 .97195333 -.675      3.0106042 -.625                 213
   0  -13     1   -1     0     1
.70771769 .88514481 1.4700828 -.775      3.9505388 .82031120            219
   0  -13    13   -1     0     1
.45       .45       .45       .475       1.8       .425                  225
   0  -13   -13   -1     0     1
.85       .85       .85       -.875      3.4       -.825                 231
   0  -13    13   -1     0     1
.50       1.7845933 .71418604 -.525      3.4987793 -.475                 237
   0  -13   -11   -1     0   -11
.60710078 .85717620 2.0065597 -.825      5.2088771 1.5201413            243
   0  -12    13  -12     0     1
.44784299 4.4459398 .75533228 3.8860436 9.8123157 -.7250000             249
   0  -13    13   -1     0     1
.70000000 .97871180 .77937012 -.7250000 3.1580819 -.6750000             255
   0  -13    13   -1     0     1
.65       .46800967 .58034818 -.675      2.3483578 -.625                 261
   0  -13    13   -1     0     1
.85       .85       .85       .875       3.4       -.825                 267
   0  -13   -13   -1     0     1
```

```
.95          .95          .95          .975         3.8          -.925                    273
   0  -13    13   -1     C     1
.45          .45          .45          -.475        1.8          .425                     279
   0  -13    13   -1     C     1
.20625       .20625       .20625       -.425        .8375        0.                       285
   0  -13    13   -1     C     1
.55          .55          .55          -.575        2.2           .525                    291
   0  -13    13   -1     C     1
.37680000    .37095959    .37095959    2.3687500    3.6856691    -.5750000                297
   0  -13    13   -1     C     1
.50          .50          .50          -.525        2.0           .475                    303
   0     C     C     0     C     C     C     0     0     0     0     0     0
   0     C     C     0   105   111    39    45   117   123     0     0     0
   0     C     C   267    27    33    39    45    51    57   129     0     0
   0     C   135    21    27    33    39    45    51    57   303     0     0
   0   141    15    21    27    33    39    45    51    57    63   147     0
   0   273    15    21    27    33    39    45    51    57    63   153     0
   0   273    15    21    27    33    39    45    51    57    63   159     0
   0   273    15    21    27    33    39    45    51    57    63   165     0
   0   273    15    21    27    33    39    45    51    57    63   279     0
   0   273    15    21    27    33    39    45    51    57    63   279     0
   0   171    15    21    27    33    39    45    51    57    63   279     0
   0   177    15    21    27    33    39    45    51    57    63   279     0
   0   183    15    21    27    33   207   213    51   291     0   225     0
   0     C   189    21    27   219     C     0   297    57   237    69   285
   0     C   195   231   243   249   255    45    51    57    63    69   285
   0     C     C     0     C   201    39    45    51    57    63    69   285
   0     C     C     0     C     C     C     0     0     0     0     0     0
.93417927
   1     1    13     1
5657.2
   1     1    13     1
5657.2
   1     1    13     1
5657.2
   3     1     4     5
5657.2     C.          5657.2
   3     1     4     5
5657.2     C.          5657.2
   3     1     4     5
5657.2     C.          5657.2
   3     1     4     5
5657.2     C.          5657.2
   3     1     4     5
5657.2     C.          5657.2
   3     1     4     5
5657.2     C.          5657.2
   3     1     4     5
5657.2     C.          5657.2
   3     1     4     5
5657.2     C.          5657.2
   3     1     4     5
5657.2     C.          5657.2
   2     1     6     6
5657.2     C.
   1    12    13     1
```

```
5657.2
   2     1     4     4
5657.2     -3184.8
   1     9    13     1
0.
   1     1     5     1
5657.2
   1     6    13     1
-3184.8
   5     1     4     4
5657.2     0.          C.          0.          0.
   1     1    13     1
0.
   1   119     9     9    67    13    11     2   200     7     1     5     7     4
   6    8C    93   1C6   119   132    67
   1    11    15     2    14   21C    13
  14    13    16     1
5657.2     -3184.8    C.          .C05        .0001       .40
1.         C.         .C3872      .C4777      8.          6.
8.854E-12  9.649E+C7  132.91E+C
6CCC.      -3CCC.     5CC.
.025C      .C275      .C3CC       .C325       .C350       .C375       .C400
.C425      .C45C
C.         C.         C.          C.          C.          C.          C.
C.         C.
   0
   1
```

REFERENCES

1. Hamza, Vladimir, and Richley, Edward A.: Numerical Solution of Two-Dimensional Poisson Equation: Theory and Application to Electrostatic-Ion-Engine Analysis. NASA TN D-1323, 1962.

2. Hamza, Vladimir, and Richley, Edward A.: Numerical Evaluation of Ion Thrustor Optics. NASA TN D-1665, 1963.

3. Harker, Kenneth J.: Determination of Electrode Shapes for Axially Symmetric Electron Guns. Jour. Appl. Phys., vol. 31, no. 12, Dec. 1960, pp. 2165-2170.

4. Varga, Richard S.: Matrix Iterative Analysis. Automatic Computation Ser., Prentice Hall, Inc., 1962.

NASA-Langley, 1963 E-2001