SLAC-51
UC-28, Particle Accelerators
and High-Voltage Machines
UC-34, Physics
TID-4500 (44th Ed.)

POISSON EQUATION SOLVING PROGRAM

by

Wm. B. Herrmannsfeldt

September 1965

Technical Report
Prepared Under
Contract AT(04-3)-400
for the USAEC
San Francisco Operations Office

Printed in USA. Price $2.00. Available from the Clearinghouse for Federal Scientific and Technical Information (CFSTI), National Bureau of Standards, U. S. Department of Commerce, Springfield, Virginia.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

The general characteristics of a pair of computer programs designed to solve applications of Poisson's equation are given. The programs are specifically written to solve electron trajectories in electrostatic focusing systems and to solve for magnetic fields. The programs could be adapted to other applications such as heat transfer problems.

I. INTRODUCTION

The modern high-speed digital computer has made possible calculations which formerly were too time-consuming to be practical. One application has been in the field of electron ballistics in which a very large amount of work has been devoted to tricks and recipes to aid in designing electron focusing devices such as electron guns. Elegant as they often are, most of these projects, together with their associated hardware such as electrolytic tanks, resistance analogues, and automatic trajectory-plotting servo systems, have been superseded by the computer.

II. BASIC POISSON EQUATION SOLVER

The basic program contains a subroutine which reads in data cards describing the boundary conditions and calculates the coefficients of the finite difference equations for each mesh point within the problem. Other subroutines are made to proceed to generate the solution to Poisson's equation which match those boundary conditions. The solution is found in terms of a set of points which form a mesh of identical squares. The potential is calculated for each intersection of the mesh. Figure 1 shows a small section of the mesh.

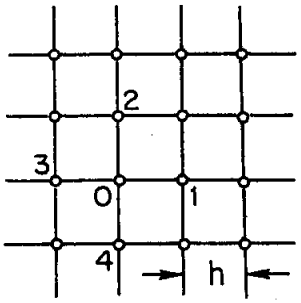


Fig. 1 - Section of mesh for solution of Poisson's equation.

In rectangular coordinates, the finite difference form of Poisson's equation is

$$V_1 + V_2 + V_3 + V_4 - 4V_0 = h^2 \text{ R.H.} \qquad (1)$$

where the V's refer to the numbered points in Fig. 1 and R.H. is the value of the righthand side of Poisson's equation at point 0 when written in the form

$$\nabla^2 V = \text{R.H.} \tag{2}$$

The usual application of the program is to problems with cylindrical symmetry. Then the finite difference equation becomes

$$RV_1 + RV_3 + (R + 1/2)V_2 + (R - 1/2)V_4 - 4RV_0 = R(\text{R.H.}) \tag{3}$$

where R is the distance in mesh units from the axis of symmetry to the point at 0.

A number of references* give the derivation of these equations and the special equations at boundaries. Three types of boundaries are of interest. A Dirichlet boundary is that boundary on which the potential is known. In an electrostatic problem, this would be an electrode fixed at a given potential. An ordinary Neumann boundary is one which lies coincident with the mesh and on which the normal derivative of the potential is known. In practice, the only value of the normal derivative that is ever known is zero. Thus, for example, the axis of symmetry of a cylindrically symmetric device has the normal derivative equal to zero and is an ordinary Neumann boundary. If, for example, we imagine a Neumann boundary being drawn horizontally through point "0" in Fig. 1 such that the body of the problem lies above the boundary, then Eq. (1) becomes

$$V_1 + 2V_2 + V_3 - 4V_0 = h^2(\text{R.H.}) \ . \tag{4}$$

---

*The following references contain material pertinent to the derivation of finite difference equations although none is complete:

1. Forsythe and Wasow, Finite Difference Methods for Partial Differential Equations, Wiley and Sons.
2. D. N. de G. Allen, M. A., Relaxation Methods, McGraw Hill.
3. F. S. Shaw, Relaxation Methods, Dover Publications.
4. Richard S. Varga, Matrix Iterative Analysis, Prentice Hall.
5. Vladimir Hamza: NASA TN 1323, TN 1665, TN 1711.

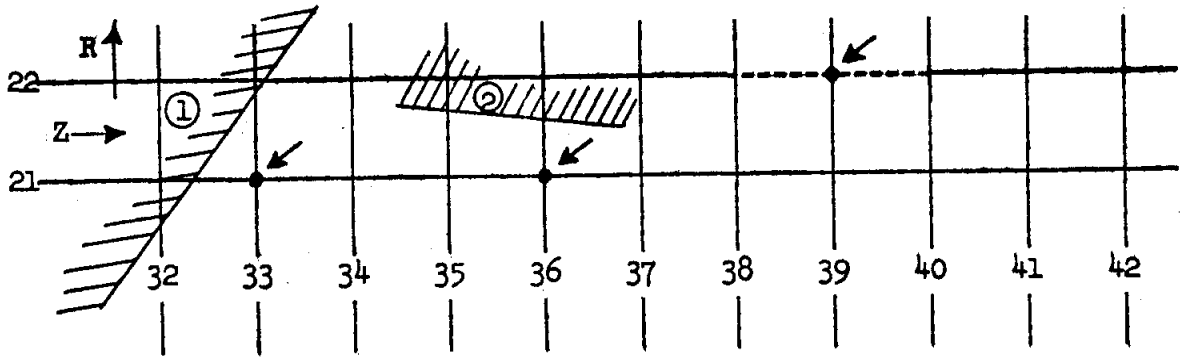
The third type of boundary, of special interest in solutions of the magnetic vector potential, is the general Neumann boundary, i.e., one which does not lie along a mesh line. It is always assumed that the normal derivative is zero. Considerable uncertainty appears in the literature over just how this type of boundary should be handled. Accordingly, we present a derivation for a special case of general Neumann boundary in Appendix II based on an approach described by Forsythe.[1]

The input of boundary information is designed to be as simple and labor saving as possible. The usual procedure is to draw the outline of the problem on coordinate paper, choosing an appropriate scale consistent with the limitations of the arrays in the program. Currently the limitations in the electron ballistics program are: Not more than 100 mesh points on the horizontal (Z) axis, not more than 60 mesh points on the vertical (R) axis, and not more than 3600 mesh points in the total system. Since the axes, which are numbered from zero, count toward the total, the limit on size is

$$(\text{RLIM} + 1)\ (\text{ZLIM} + 1) \leq 3600 \qquad (5)$$

where RLIM and ZLIM are the program notation for limits of the problem in the R and Z directions, respectively. The limitations on the magnet design program are somewhat higher since the length of the program is less, permitting more of the memory core to be used for storage. Surfaces of known potential are each assigned a number in series. Normally the cathode is labelled 1, the anode is 2 and the grid, if any, is labelled 3. Other surfaces up to a limit of 40 can be numbered in any sequence. The potential values are all read in on one card, thus permitting changes in focusing by changing only that one card, leaving the boundary cards alone.

The boundary points are specified in sequence around the problem to enable the plotter to draw the boundary in one continuous line. A separate data card is used for each point. Each card has five numbers; surface number (Integer), R-coordinate (Integer), Z-coordinate (Integer), Delta-R (Real), Delta-Z (Real). Some examples of boundary points are shown in Fig. 2.



Case A Case B Case C

Case	Surface	R	Z	Delta-R	Delta-Z
A	1	21	33	0.8	-0.6
B	2	21	36	0.6	2.0
C	0	22	39	0.0	2.0

Fig. 2 - Sample boundary points showing the input format and conventions.

Note that each boundary point is actually within the problem area. Case A shows a boundary point within one mesh unit of the surface in both directions. Delta-R and Delta-Z are the distance from the point to the surface. Thus, distances to the left or down result in negative delta-z or delta-r, respectively. By definition, delta-r and delta-z must be between 0.0 and $\pm$ 1.0. If, as in Case B, the surface does not cross within one mesh unit (in the z-direction in this case), then typically the number 2.0 is used as a code number. Since a boundary point cannot lie on a surface, it is possible to use 0.0 as a code for a Neumann boundary as in Case C. Both delta-r and delta-z can be zero at an outside corner of a problem.

Normally the intersection of different boundaries is always between a Neumann boundary and a Dirichlet boundary. This is because otherwise a short circuit would occur in practice. By making the code for a Neumann boundary be 0.0 for delta-r or delta-z, it is possible to specify a Dirichlet boundary in one direction and a Neumann boundary in the other. If there is no Dirichlet boundary within one mesh unit of a Neumann point,

then for the lack of any other value, the surface number is set to 0. However, in the case of a general Neumann boundary (see Appendix II) the 0 surface number does call up a special set of instructions for calculating the coefficients for the difference equations.

To save work, the program has been written to fill in gaps between data cards. Consecutive boundary points must be within $\pm$ 1 mesh unit in both R and Z. If the next boundary point is skipped in the input deck, the program immediately reads the third card in the sequence. It then attempts to fit the three points to a second degree equation of the form

$$Z = aR^2 + bR + c \tag{6}$$

or

$$R = a'Z^2 + b'Z + c', \tag{7}$$

The choice between Eqs. (6) and (7) is according to which of the slopes, b or b', are less than unity. The program can switch from Eq. (6) to Eq. (7), or vice-versa, if a surface curves through 45° to the axis. Care must be taken in using the fill-in features of the program so as to only specify segments of curves which can be reasonably fit by equations of the form of Eqs. (6) and (7). Usually, the agreement is quite good. The cathode of most gun problems can be specified on just three cards. An ordinary Pierce diode can be specified with 25 or so data cards even though the boundary may have about 200 points.

A variety of errors can occur in making up the input cards. If for any reason the boundaries are not completely specified by the input cards, the calculation is abandoned and a plot of the boundary is drawn. Error messages are generated under some conditions to aid in finding which boundary point is in error. Failure of the program to fill in boundary points successfully also generates a plot to help in debugging.

After successfully compiling the boundary and generating the difference equations, the program finds a solution to Poisson's equation. A single card gives a parameter called the "Spectral Radius" to aid in calculating Chebyshev polynomials which are used by the program in a matrix inversion technique known as a "semi-iterative Chebyshev" method

of solving the particular tri-diagonal matrix which is generated for each column of the problem. The preferred reference for this technique is Varga's book.[4] In one iteration, alternate columns are relaxed, first the odd numbered columns, then the even numbered, etc. The Spectral Radius, can be calculated by an iterative method similar to that used for the relaxation. In practice it has been found easier to guess at the Spectral Radius and if necessary, to improve on the first guess as a problem is rerun with minor changes. A large problem will typically have a Spectral Radius of 0.993 to 0.995. Too high a value slows up the convergence and can cause oscillation at Neumann boundaries. Too small a value can cause a sharp increase in convergence time. A typical problem with 2000 or more mesh points is solved in 50 to 100 iterations to an accuracy of one part in 10^{+7}. Problems with large areas of Neumann boundaries may take several hundred iterations. For a typical problem, one iteration requires about one second.

III. MAGNET DESIGN PROGRAM

Both magnetic and electrostatic fields can be solved by Poisson's equation. In the case of magnetic fields, however, the method and form of the solution may have several variations. In its present form, the magnet design program does not include saturation of the iron. However, it can be used quite effectively to predict where and how saturation will occur and to aid in designing a magnet so that saturation occurs as nearly uniformly as possible.

A. Vector Potential

The traditional way to apply Poisson's equation to magnetic fields is by means of the magnetic vector potential. In rectangular coordinates, the Z-component of the magnetic vector potential is a solution of the equation

$$\frac{\partial^2 A_z}{\partial x^2} + \frac{\partial^2 A_z}{\partial y^2} = - \mu j_z \qquad (8)$$

where

$$B_x = \frac{\partial A_z}{\partial y} \quad \text{and} \quad B_y = -\frac{\partial A_z}{\partial x} \quad . \tag{9}$$

The vector potential is constant along a flux line. Neglecting saturation, the normal derivative of the vector potential is zero at an iron boundary and zero on a plane of symmetry. Whenever a single line of flux is known, as through the center of a symmetric quadrupole, then it is convenient to apply a constant, such as zero, to a boundary along the known line of flux. Figure 3 shows a computer plot of a segment of a symmetric quadrupole. The diagonal boundary is set to A = 0. The horizontal axis and the iron pole piece both are Neumann boundaries with the normal derivative of A set to zero. The coils, indicated by the cross-hatching, are on the right. A few equipotential lines are plotted showing how typical lines of flux pass through the central region.

If there is no known line on which the vector potential is a constant it is very difficult to get the program to converge. Even with such a line, convergence is almost ten times slower than an equivalent scalar problem because of the large areas of Neumann boundaries. Special boundary versions of the finite difference equations are required (see Appendix II) on the iron surfaces. Within the limitations posed by these objections, quite useful results can be obtained with the vector potential method.

B. Scalar Potential

Perhaps because the concept of a fixed potential on a surface is so familiar, it has long been a useful approach to consider a magnetic scalar potential when designing magnets. This approach can be immediately extended to quantitative analysis of the fields by assigning a constant to both the iron surface and to the symmetry plane if there is one. Figure 4 shows the equipotential lines on the SLAC beam focusing quadrupole elements. This particular plot was generated by a program which was run to analyze the perturbations resulting from small errors on the poles. The errors were such that symmetry could not be assumed. The results showed that the magnets were still acceptable and production

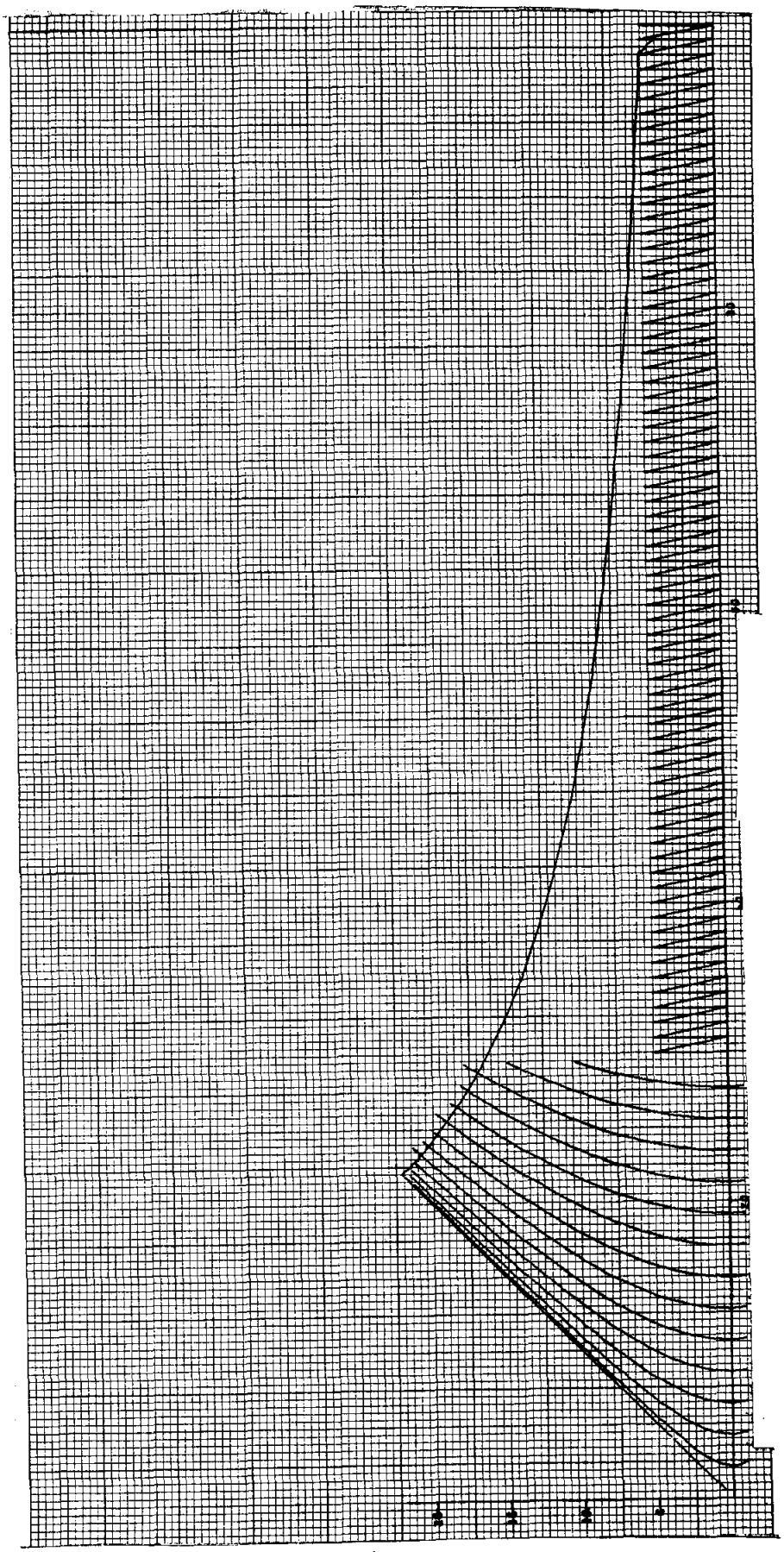


FIG. 3--Computer plot of one-eighth of a symmetric quadrupole.

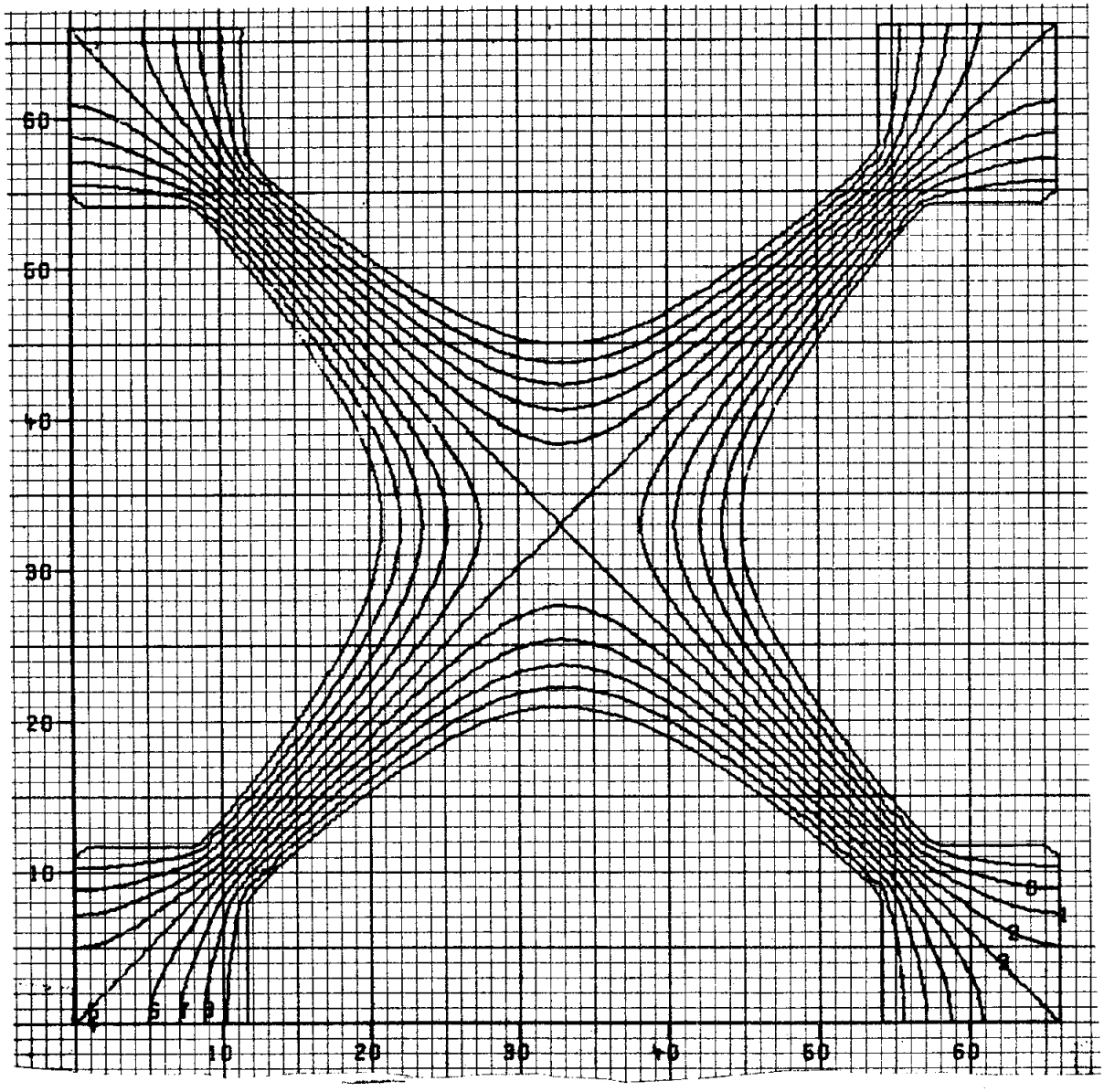


FIG. 4--Equipotential lines of SLAC triplet.

was allowed to proceed even though some doubt remained whether the pole pieces were being shaped to within the design tolerances.

The usefulness of the scalar potential has been extended to include current bearing elements by M. H. Blewett of Brookhaven National Laboratory.[6] This method has been incorporated into the program and since it is not a generally known approach, the derivation follows. If A is the usual vector potential, then let

$$-\frac{\partial \varphi}{\partial y} = B_y = -\frac{\partial A}{\partial x} . \tag{10}$$

Differentiating with respect to x yields

$$-\frac{\partial^2 \varphi}{\partial x \partial y} = -\frac{\partial^2 A}{\partial x^2}$$

which, when combined with Eq. (8) yields

$$-\frac{\partial^2 \varphi}{\partial x \partial y} = \mu j + \frac{\partial^2 A}{\partial y^2} . \tag{11}$$

Integrating with respect to y gives

$$-\frac{\partial \varphi}{\partial x} = \int_0^y \mu j dy + \frac{\partial A}{\partial y} + \text{constant.} \tag{12}$$

But, $\frac{\partial A}{\partial y} = B_x$ so that the constant of integration is zero if the x-axis is a line of symmetry. Thus

$$B_x = -\frac{\partial \varphi}{\partial x} - \int_0^y \mu j dy \tag{13}$$

so that φ is the ordinary scaler potential everwhere except in the current-bearing region where an extra term is added to the B_x expression.

---

[6] Particle Accelerators, Livingston and Blewett, McGraw-Hill, Page 253.

From $\vec{\nabla} \cdot \vec{B} = 0$ we have

$$\frac{\partial B_x}{\partial x} + \frac{\partial B_y}{\partial y} = - \frac{\partial^2 \varphi}{\partial x^2} - \frac{\partial}{\partial x} \int_0^y \mu j dy - \frac{\partial^2 \varphi}{\partial y^2} = 0$$

from which Poisson's equation becomes

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = - \frac{\partial}{\partial x} \left(\int_0^y \mu j dy \right) , \tag{14}$$

It should be noted that the Laplacian is zero within a uniform coil as well as in the current-free region of the magnet. The right-hand side of the equation is non-zero only at the vertical boundary of a coil.

An instructive example is the window magnet in Fig. 5, which is known to yield a uniform field in the y-direction.

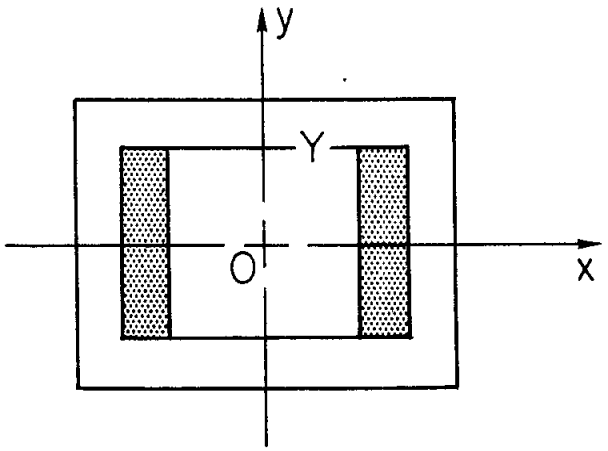


Fig. 5 - Window Magnet

On the top surface, y = Y, the field component $B_x = 0$, and Eq. (13) becomes

$$\frac{\partial \varphi}{\partial x} = - \int_0^Y \mu j dy = - \mu j Y . \tag{15}$$

Integrating yields

$$\varphi = \varphi_Y - \mu j Y \Delta x \tag{16}$$

where φ_y is the potential assigned to the upper poleface at x = 0 and Δx is the width of the coil. If $\varphi = 0$ at y = 0, then $\varphi_Y = + \mu I$ where I is the total number of ampere-turns in one quadrant of the coil. Equation (15) shows that

$$\frac{\partial \varphi}{\partial x} = 0 \quad \text{for } j = 0 . \tag{17}$$

The potential is thus graduated linearly along the upper edge of the coil from $\varphi_Y = \mu I$ to zero at the outer edge of the coil. This problem of knowing how to terminate the equipotentials is perhaps the greatest drawback to using this method. The current density μj, is given by

$$\mu j = \frac{+\varphi_Y}{Y \Delta x} \tag{18}$$

where $Y\Delta x$ is the area of the coil.

C. Magnetic Field Analysis

The usual way of studying the results of a magnet problem is to calculate the field along various lines of interest. This calculation uses PROCEDURE PARTIAL which was written for the electron ballistics program. PARTIAL calculates the first and second derivatives of the potential at the nearest mesh point to the spot at which the field is required. Then using the second derivatives, it adjusts the first derivatives according to the displacement from the mesh point. The first derivative then yields the magnetic field, either by the curl, as given by Eq. 9 for the vector potential, or by the gradient for the scalar potential. The program automatically chooses the correct derivatives to calculate the field according to which program option, scalar or vector, is being employed.

The field is usually calculated for the entire boundary of the problem. PROCEDURE PARTIAL has special forms which make it nearly correct near boundaries. The results can be used to pinpoint spots of high intensity along the pole tip and to check on the linearity or uniformity of the field along the plane of symmetry. Other diagnostics, such as one to calculate the strength of the sextuple element in a field, can be written at will.

IV. ELECTRON BALLISTICS AND GUN DESIGN

A. Program Organization

The primary purpose of this program is the design of electron guns such as the injection gun and klystron guns for the two-mile accelerator. The sequential operation of the program, including the most important subprograms, is as follows:

1. READ the title card and data giving RLIM, ZLIM, number of potentials, their values in sequence (volts), and MID which is a code number controlling which plots will be made.

2. READ the axial magnetic field in gauss. The axial field is assumed to be constant as a function of R. Variations of the axial field as a function of Z are used by the program to calculate the radial field from Maxwell's equations. Simplified input is available for sections of constant field and a SENTINEL card terminates the input of magnetic field, setting unspecified points to zero.

3. READ the boundary information as described earlier. The input of boundary cards and the calculation of the coefficients of the difference equations is made by SUBROUTINE CYLINDRICAL. A SENTINEL card terminates the input of boundary cards.

4. READ the "SPECTRAL RADIUS."

5. Solve Laplace's Equation using SUBROUTINE POISSON and PROCEDURE MATRIX to invert the matrix for each column. Up to 100 iterations are allowed for the initial solution and up to 50 for subsequent iterations. The error limits are specified within the program but may be changed at will.

6. READ the cathode data which includes (in mesh units) the spherical radius of the cathode, the radius of the emitting area of the cathode, the Z-coordinate of the center of the cathode, the scale factor of the gun in meters per mesh unit, the iteration step for trajectory calculations (which must be less than one and is usually about 0.4), the starting step ST from the cathode (usually $2 < ST < 4$), and the number of iterations (usually 5).

7. SUBROUTINE PERVEANCE uses the information on the cathode card, together with the boundary information, to construct a mathematical

model which nearly fits a theoretical Pierce diode. The perveance calculated will be used for the first guess of the program if the next card is a SENTINEL card. If instead, the next card contains a number, that number will be used for the first guess.

8. SUBROUTINE CHILD calculates the initial conditions for 27 rays assuming space-charge limited emission. On the first iteration the current assigned to each ray and the initial energy of the ray are normalized to the first guess obtained in Step 7. The starting points are spaced evenly along the surface of an imaginary sphere a distance ST from the cathode with one-half a space at the top and bottom.

9. SUBROUTINE TRAJECT calculates the trajectory of each of the 27 rays until the ray leaves the area of the problem either through the side or out the end or anode hole. As a ray crosses a vertical mesh line, the space charge contribution is calculated and distributed to the two mesh points above and below the point where the ray crossed. A point is recorded in an array for eventual plotting and magnetic fields are calculated also when a ray crosses a vertical mesh line. TRAJECT uses one of a variety of methods and sets of equations to integrate the trajectory. These will be described in more detail below. TRAJECT calls on PROCEDURE PARTIAL to calculate the field at any point at which it may be required. PARTIAL takes proper account of the presence of any nearby boundary with sufficient accuracy that trajectories may pass directly through a boundary. This latter feature permits the programmer to specify a thin grid as part of the boundary conditions. The trajectories pass through the grid as though it were a perfect grid which means zero interception and no leakage of electrostatic fields.

10. Solve Poisson's equation using the space charge calculated from Step 9.

11. Repeat the calculation of the initial conditions of the rays using SUBROUTINE CHILD. On the second and all subsequent iterations, the perveance is normalized to the average of the last value and the value calculated from the fields. This simple solution, i.e., averaging, yields faster convergence than any reported by any other worker in this field to the author's knowledge. Usually, the perveance has converged

to within a few percent after 4 or 5 iterations. Agreement with experimental measurement is almost always within 5% which is about the accuracy of such measurements considering uncertainty in the actual "hot" dimensions of the gun and uncertainty about the condition of the cathode.

12. At the end of the specified number of iterations a plot is generated automatically on the Cal-Comp Plotter. Normally two plots are made; the first shows the outline of the problem and the equipotential lines, the second plot shows the outline of the problem and the complete trajectory paths of the 27 rays. Examples of these plots are shown in Fig. 6 and Fig. 7, respectively. Options, controlled by the control number MID on the first data card, permit the user to get both the first or Laplace iteration and the last iteration or just the last iteration. The user can also elect to suppress the equipotential contour plots or to suppress all plotting.

13. A set of 27 cards are punched at the end of the program giving the final conditions of the 27 rays. These cards have the format required to use the starting option of SUBROUTINE INIT. With this option, instead of using a cathode, the trajectories are calculated from the starting conditions specified on the cards. The cards may be those punched by a previous run or they may contain data specified by the user. To use the option, the "cathode" card is replaced by a SENTINEL card, the next card contains (in mesh units), the iteration step, the number of iterations, the position of the origin of the previous run in the new coordinate system (0.0 if the problem is run from user specified data), the ratio of the old system mesh length to that of the new system (1.0 if the problem is run from user specified data), and the scale factor in meters per mesh unit. Next, the cards with initial conditions are read in, up to 27 in number, finishing with a SENTINEL card. The initial conditions specified on the cards are; the trajectory number, the starting radius, the starting value of Z, the energy in the R-Z plane in electron volts, the angle in radians between the trajectory and the Z-axis, the energy due to the azimuthal component of the velocity, and the current in microamperes assigned to a one-radian segment of the ray.

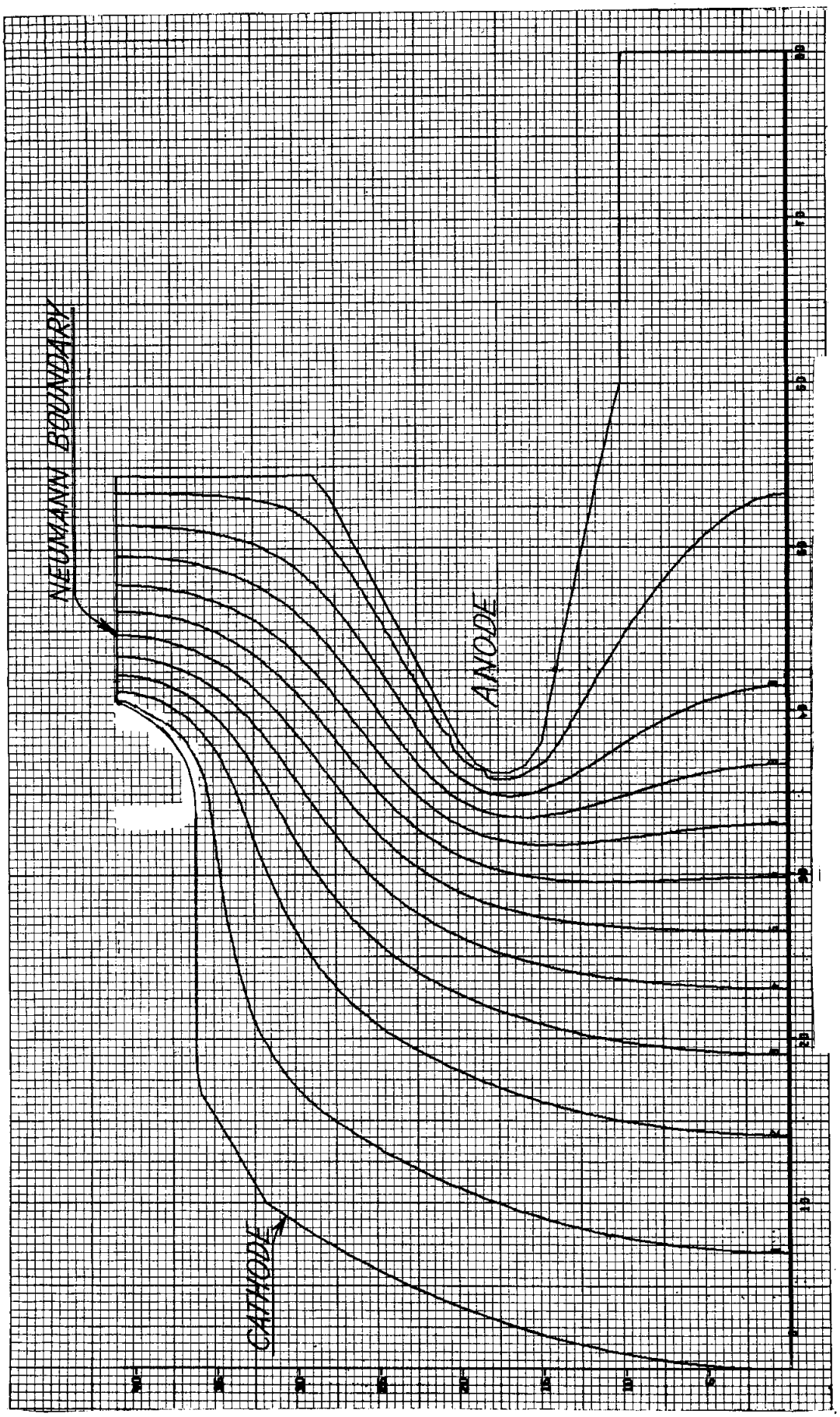


FIG. 6--Equipotential plot of SLAC klystron gun.

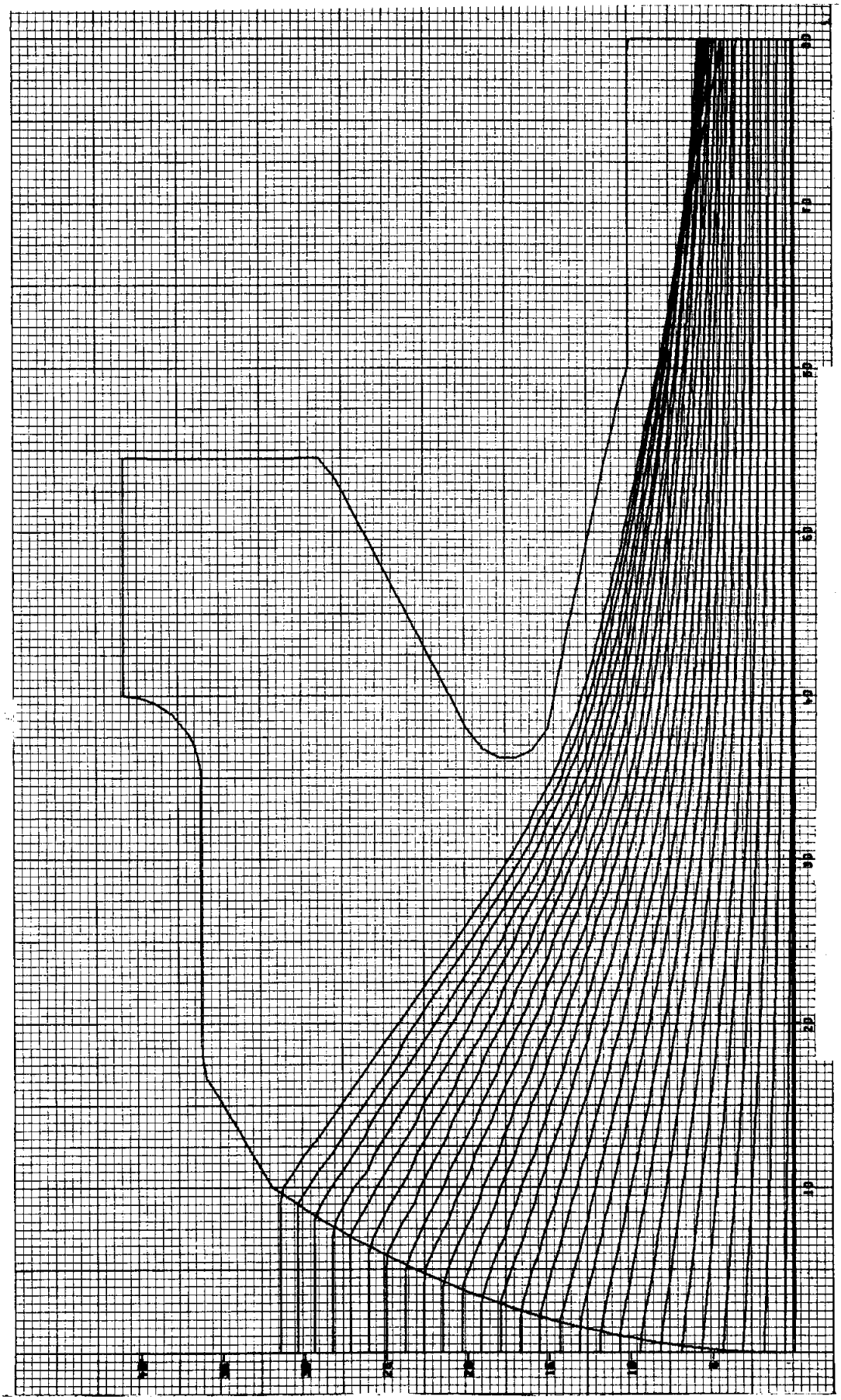


FIG. 7--Electron trajectories in the SLAC klystron gun.

B. Trajectory Calculations

Various sets of equations and techniques have been used to calculate the trajectories. The present version of the program uses a fourth-order Runge-Kutta method of solving the relativistic differential equations given below. Suitable substitutions are used to reduce the three second-order equations to six first-order differential equations.

The independent variable is time but the time interval is calculated from the allowed iteration step and the velocity. It is necessary to use fairly short steps because of the auxilliary calculations that must be made at each mesh unit. Thus, it is generally not helpful to use any self-checking "corrector" solving routine. If some unusual application requires shorter iteration steps, the results usually show this by their internal inconsistency.

The relativistic differential equations are derived in Appendix I and are

$$\dot{\dot{Z}} = \alpha\,(1-\beta^2)^{\frac{1}{2}}\left[-E_z(1-\dot{Z}^2) + \dot{Z}\dot{R}E_r + \dot{Z}\dot{A}E_\varphi - c\dot{R}B_\varphi + c\dot{A}B_r\right], \tag{19}$$

$$\dot{\dot{R}} = \alpha\,(1-\beta^2)^{\frac{1}{2}}\left[-E_r(1-\dot{R}^2) + \dot{Z}\dot{R}E_z + \dot{R}\dot{A}E_\varphi + c\dot{Z}B_\varphi - c\dot{A}B_z\right] + \frac{\dot{A}^2}{R} \tag{20}$$

and

$$\dot{\dot{A}} = \alpha\,(1-\beta^2)^{\frac{1}{2}}\left[-E_\varphi(1-\dot{A}^2) + \dot{Z}\dot{A}E_z + \dot{R}\dot{A}E_r - c\dot{Z}B_r - c\dot{R}B_r\right] - \frac{\dot{R}\dot{A}}{R}\ . \tag{21}$$

Recapping the definitions from the appendix, we have

$$\beta^2 = \dot{Z}^2 + \dot{R}^2 + \dot{A}^2 \quad \text{where } \beta = v/c\ . \tag{22}$$

The constant $\alpha = e\lambda/m_oc^2$ where e is the magnitude of the electron charge (the "-" sign is in the equations), m_oc^2 is the rest energy of the electron and λ is the constant of proportionality between the real coordinates and the dimensionless coordinates. Thus

$$z = \lambda Z,\ r = \lambda R,\ a = \lambda A \text{ and } ct = \lambda T\ . \tag{23}$$

We choose to let $\lambda = 5.11 \times 10^5$ mesh units so that $\alpha = 1.0$ mesh unit per volt. Inspection of the differential equations shows that they are dimensionally correct if the electric fields are specified in volts per mesh unit.

Dimensionally $E = vB$, so that in mksa units E is in volts per meter, v is in meters per second and B is in webers per meter^2. Thus one weber is one volt-second. Then cB has units of volts per meter. To convert to program fields of volts per mesh unit, we multiply by the number of meters per mesh unit.

The program stores the axial magnetic field along the Z-axis. It is assumed that the axial component of the field is constant in the radial direction. From Maxwell's equation $\nabla \cdot \vec{B} = 0$, where B_φ is zero (except for the self-magnetic field of the beam) we have

$$\frac{1}{r} \frac{\partial}{\partial r} (rB_r) = - \frac{\partial B_z}{\partial Z}$$

from which $rB_r = - \frac{\partial B_z}{\partial Z} \int_0^r rdr$ and

$$B_r = - \frac{r}{2} \frac{\partial B_z}{\partial Z} \tag{24}$$

Thus, by assuming radial uniformity of the axial field, and given the axial field and the scale factor, the program calculates both the radial and axial field in program units. The field is specified in gauss which is the common engineering unit.

The azimuthal magnetic field B_φ comes from the current in the electron beam and is called the self-magnetic field of the beam. The magnetic field of a current in a wire is given by

$$B_\varphi = \frac{\mu_o}{2\pi} \frac{I}{r} \text{ webers/meter}^2. \tag{25}$$

We assume that the field is due to an infinite conductor which is a pretty good approximation in the area in which the field is significant. When we multiply B_φ by the scale factor and express r in meters which

requires multiplying r by the scale factor also, the scale factor cancels as we might expect. Thus the scale factor only enters if we have external magnetic fields. The current I in Eq. (25) is the summation of the current in the trajectories at lower radii than the trajectory being calculated.

Two field components are neglected. The azimuthal electric field is neglected because of the axial symmetry assumed. The axial magnetic field can have a contribution from the beam due to azimuthal velocity of the beam. The magnitude has been shown to be less than one gauss and so is neglected.

The space charge is calculated to supply the right side of Poisson's equation which is

$$\nabla^2 V = -\frac{\rho}{\epsilon_o} = -\frac{J}{v\epsilon_o} \qquad (26)$$

The element of area for J is $r\delta r$ where r is the radius and δr is the gap between rays. The current in that area is constant and is calculated by Child's law in SUBROUTINE CHILD. The velocity is only the Z-component since the space charge is being spread between adjacent points on the same column. Thus the right hand side of Eq. (26) becomes, in program notation,

$$RO = \frac{36\pi \times 10^9 \; IO(K) \times 10^{-6}}{ZDOT \times 3 \times 10^8} = (3.77**-4)\frac{IO(K)}{ZD\varphi T}$$

where RO is to be spread between two points in inverse ratio to the distance the ray is between them, IO(K) is the current in the one radian segment of the ray (in micro amperes) and ZDOT is the velocity in units of c.

APPENDIX I

DERIVATION OF EQUATIONS OF MOTION*

The equations of motion are derived* from the Lorentz force equation

$$\frac{d(m\vec{v})}{dt} = -e(\vec{E} + \vec{v} \times \vec{B}), \tag{1}$$

where e is the magnitude of the charge of an electron. The electron velocity vector v, expressed in cylindrical coordinates is

$$\vec{v} = u_z \dot{z} + u_r \dot{r} + u_\varphi \dot{a} . \tag{2}$$

Here u_z, u_r and u_φ are unit vectors and $\dot{a} = r\dot{\varphi}$ is the azimuthal or peripheral velocity. The left side of Eq. (1) can be found from

$$\frac{d(m\vec{v})}{dt} = \frac{d}{dt}\left(m_o \left(1 - \frac{v^2}{c^2}\right)^{-\frac{1}{2}} \vec{v}\right) \tag{3}$$

where m_o is the electron rest mass. Differentiating Eq. (3) yields

$$\frac{d(m\vec{v})}{dt} = m_o \left(1 - \frac{v^2}{c^2}\right)^{-3/2} \left[\frac{v}{c^2}\frac{dv}{dt}\vec{v} + \left(1 - \frac{v^2}{c^2}\right)\frac{d\vec{v}}{dt}\right] \tag{4}$$

where

$$\frac{d\vec{v}}{dt} = u_z \ddot{z} + u_r (\ddot{r} - r\dot{\varphi}^2) + u_\varphi (2\dot{r}\dot{\varphi} + r\ddot{\varphi}) \tag{5}$$

which becomes

$$\frac{d\vec{v}}{dt} = u_z \ddot{z} + u_r(\ddot{r} - \dot{a}^2/r) + u_\varphi (\ddot{r}a/r + \ddot{a}) . \tag{6}$$

From $v = (\dot{z}^2 + \dot{r}^2 + \dot{a}^2)^{\frac{1}{2}}$ where v is the scalar velocity, (7)

we have

$$\frac{dv}{dt} = \frac{1}{v}(\dot{z}\ddot{z} + \dot{r}\ddot{r} + \dot{a}\ddot{a}) . \tag{8}$$

---

*This derivation was suggested by Dr. Gene Lang in a private communication.

Substituting Eqs. (6), (7) and (8) in Eq. (4) yields

$$\frac{d(m\vec{v})}{dt} = m_o \left(1 - \frac{v^2}{c^2}\right)^{-3/2} \left[\frac{1}{c^2}\left(\dot{z}\,\ddot{z} + \dot{r}\,\ddot{r} + \dot{a}\,\ddot{a}\right)\left(u_z\dot{z} + u_r\dot{r} + u_\varphi\dot{a}\right)\right.$$

$$\left. + \left(1 - \frac{v^2}{c^2}\right)\left\{u_z\ddot{z} + u_r(\ddot{r} - \dot{a}^2/r) + u_\varphi(\dot{r}\,\dot{a}/r + \ddot{a})\right\}\right] . \qquad (9)$$

Equation (9) can be expanded and grouped by vector components yielding

$$\frac{d(m\vec{v})}{dt} = m_o \left(1 - \frac{v^2}{c^2}\right)^{-3/2} \left[u_z\left\{\frac{1}{c^2}\dot{z}(\dot{r}\,\ddot{r} + \dot{a}\,\ddot{a}) + \ddot{z}\left(1 - \frac{v^2}{c^2} + \frac{\dot{z}^2}{c^2}\right)\right\}\right.$$

$$+ u_r\left\{\frac{1}{c^2}\dot{r}\,(\dot{z}\,\ddot{z} + \dot{a}\,\ddot{a}) - \frac{\dot{a}^2}{r}\left(1 - \frac{v^2}{c^2}\right) + \ddot{r}\left(1 - \frac{v^2}{c^2} + \frac{\dot{r}^2}{c^2}\right)\right\} \qquad (10)$$

$$\left. + u_\varphi\left\{\frac{1}{c^2}\dot{a}(\dot{z}\,\ddot{z} + \dot{r}\,\ddot{r}) + \frac{\dot{r}\,\dot{a}}{r}\left(1 - \frac{v^2}{c^2}\right) + \ddot{a}\left(1 - \frac{v^2}{c^2} + \frac{\dot{a}^2}{c^2}\right)\right\}\right] .$$

A similar vector component expansion can be made for the right side of Eq. (1) yielding

$$\frac{d(m\vec{v})}{dt} = -e\left[u_z(E_z + \dot{r}B_\varphi - \dot{a}B_r) + u_r(E_r + \dot{a}B_z - \dot{z}B_\varphi) + u_\varphi(E_\varphi + \dot{z}B_r - \dot{r}B_z)\right] . \qquad (11)$$

Equating vector components we have finally

$$m_o \left(1 - \frac{v^2}{c^2}\right)^{-3/2} \left\{\left(1 - \frac{v^2}{c^2} + \frac{\dot{z}^2}{c^2}\right)\ddot{z} + \frac{1}{c^2}\dot{z}\dot{r}\ddot{r} + \frac{1}{c^2}\dot{z}\dot{a}\ddot{a}\right\} = -e(E_z + \dot{r}B_\varphi - \dot{a}B_r) , \qquad (12)$$

$$m_o\left(1 - \frac{v^2}{c^2}\right)^{-3/2} \left\{\frac{1}{c^2}\dot{r}\dot{z}\,\ddot{z} + \left(1 - \frac{v^2}{c^2} + \frac{\dot{r}^2}{c^2}\right)\ddot{r} + \frac{1}{c^2}\dot{r}\dot{a}\,\ddot{a} - \frac{\dot{a}^2}{r}\left(1 - \frac{v^2}{c^2}\right)\right\}$$

$$= -e(E_r - \dot{z}B_\varphi + \dot{a}B_z) \qquad (13)$$

and

$$m_o\left(1-\frac{v^2}{c^2}\right)^{-3/2}\left\{\frac{1}{c^2}\dot{a}\ddot{z}\,\ddot{z}+\frac{1}{c^2}\dot{a}\ddot{r}\,\ddot{r}+\left(1-\frac{v^2}{c^2}+\frac{\dot{a}^2}{c^2}\right)\ddot{a}+\frac{\dot{r}\dot{a}}{r}\left(1-\frac{v^2}{c^2}\right)\right\}$$

$$=-e(E_\varphi+\dot{z}B_r-\dot{r}B_z)\ . \tag{14}$$

For computer programming it is convenient to express the variables in a normalized form. Accordingly, we let

$$z=\lambda Z,\ r=\lambda R,\ a=\lambda A \text{ and } ct=\lambda T. \tag{15}$$

We differentiate with respect to $T=\frac{ct}{\lambda}$ to get

$$\dot{z}=c\dot{Z},\ \ddot{z}=\frac{c^2\ddot{Z}}{\lambda}\ ,$$

$$\dot{r}=c\dot{R},\ \ddot{r}=\frac{c^2\ddot{R}}{\lambda}\ , \tag{16}$$

and

$$\dot{a}=c\dot{A},\ \ddot{a}=\frac{c^2\ddot{A}}{\lambda}\ .$$

From the definitions in Eqs. (16) it follows that

$$\beta^2=\frac{v^2}{c^2}=\dot{Z}^2+\dot{R}^2+\dot{A}^2\ . \tag{17}$$

Making the normalizing substitutions in Eqs. (12), (13), and (17) yields

$$\frac{m_oc^2}{\lambda(1-\beta^2)^{3/2}}\left[\left(1-\beta^2+\dot{Z}^2\right)\ddot{Z}+\dot{Z}\dot{R}\,\ddot{R}+\dot{Z}\dot{A}\,\ddot{A}\right]=-e\left[E_z+c\dot{R}B_\varphi-c\dot{A}B_r\right], \tag{18}$$

$$\frac{m_oc^2}{\lambda(1-\beta^2)^{3/2}}\left[\dot{R}\dot{Z}\,\ddot{Z}+(1-\beta^2+\dot{R}^2)\ddot{R}+\dot{R}\dot{A}\,\ddot{A}-\frac{\dot{A}^2}{R}(1-\beta^2)\right]=-e\left[E_r-c\dot{Z}B_\varphi+c\dot{A}B_z\right] \tag{19}$$

and

$$\frac{m_o c^2}{\lambda(1-\beta^2)^{3/2}}\left[\dot{A}\dot{Z}\,\ddot{Z} + \dot{A}\dot{R}\,\ddot{R} + (1-\beta^2+\dot{A}^2)\ddot{A} + \frac{\dot{R}\dot{A}}{R}(1-\beta^2)\right] = -e\left[E_\varphi + c\dot{Z}B_r - c\dot{R}B_z\right]. \quad (20)$$

Our goal is to get separated equations solved for the second order derivative of each of the orthogonal variables. To solve the equations, we arrange them in the form

$$\begin{aligned} A_1\ddot{Z} + B_1\ddot{R} + C_1\ddot{A} &= D_1 \\ A_2\ddot{Z} + B_2\ddot{R} + C_2\ddot{A} &= D_2 \\ A_3\ddot{Z} + B_3\ddot{R} + C_3\ddot{A} &= D_3 \end{aligned} \quad (21)$$

and apply the standard determinant method of solving simultaneous equations. Rearranging Eqs. (18), (19) and (20) in the form of Eq. (21) yields

$$(1-\beta^2+\dot{Z}^2)\ddot{Z} + \dot{Z}\dot{R}\,\ddot{R} + \dot{Z}\dot{A}\,\ddot{A} = \frac{-e\lambda}{m_o c^2}(1-\beta^2)^{3/2}(E_z + c\dot{R}B_\varphi - c\dot{A}B_r), \quad (22)$$

$$\dot{R}\dot{Z}\,\ddot{Z} + (1-\beta^2+\dot{R}^2)\ddot{R} + \dot{R}\dot{A}\,\ddot{A} = (1-\beta^2)\frac{\dot{A}^2}{R} - \frac{e\lambda}{m_o c^2} \times (E_r - c\dot{Z}B_\varphi + c\dot{A}B_z)(1-\beta^2)^{3/2}, \quad (23)$$

and

$$\dot{A}\dot{Z}\,\ddot{Z} + \dot{A}\dot{R}\,\ddot{R} + (1-\beta^2+\dot{A}^2)\ddot{A} = -(1-\beta^2)\frac{\dot{R}\dot{A}}{R} - \frac{e\lambda}{m_o c^2} \times (E_\varphi + c\dot{Z}B_r - cRB_z)(1-\beta^2)^{3/2}. \quad (24)$$

The determinant of the coefficients is

$$\Delta = (1-\beta^2+\dot{Z}^2)\left[(1-\beta^2+\dot{R}^2)(1-\beta^2+\dot{A}^2)-\dot{A}^2\dot{R}^2\right] + \dot{Z}\dot{R}\left[\dot{R}\dot{Z}\dot{A}^2 - \dot{Z}\dot{R}(1-\beta^2+\dot{A}^2)\right]$$

$$+ \dot{Z}\dot{A}\left[\dot{Z}\dot{R}^2\,\dot{A} - \dot{A}\dot{Z}\,(1-\beta^2+ R^2)\right] = (1-\beta^2+\dot{Z}^2)(1-\beta^2)(1-\beta^2+\dot{A}^2+\dot{R}^2)$$

$$- \dot{Z}^2\dot{R}^2(1-\beta^2) - \dot{Z}^2\dot{A}^2(1-\beta^2) = (1-\beta^2)^2(1-\beta^2+ \dot{Z}^2+ \dot{R}^2+ \dot{A}^2)$$

which is simply

$$\Delta = (1-\beta^2)^2 \ . \qquad (25)$$

It is convenient to let $\alpha = e\lambda/m_o c^2$. The axial acceleration $\ddot{Z}$, is given by

$$\Delta\ddot{Z} = D_1(B_2C_3 - C_2B_3) + D_2(C_1B_3 - B_1C_3) + D_3(B_1C_2 - C_1B_2)$$

which becomes

$$(1-\beta^2)^2\,\ddot{Z} = \left[-\alpha(1-\beta^2)^{3/2}(E_z+c\dot{R}B_\varphi - c\dot{A}B_r)\right]\left[(1-\beta^2+\dot{R}^2)\times(1-\beta^2+\dot{A}^2)-\dot{R}^2\dot{A}^2\right]$$

$$+ \left[(1-\beta^2)\,\frac{\dot{A}^2}{R} - \alpha(1-\beta^2)^{3/2}\,(E_r - c\dot{Z}B_\varphi + c\dot{A}B_z)\right]\times\left[\dot{Z}\dot{R}\dot{A}^2-\dot{Z}\dot{R}(1-\beta^2+\dot{A}^2)\right]$$

$$+ \left[-(1-\beta^2)\,\frac{\dot{R}\dot{A}}{R} - \alpha(1-\beta^2)^{3/2}(E_\varphi + c\dot{Z}B_r - c\dot{R}B_z)\right]\times\left[\dot{Z}\dot{R}^2\dot{A}-(1-\beta^2+\dot{R}^2)\dot{Z}\dot{A}\right].$$

Simplified, the above equation yields

$$\ddot{Z} = \alpha(1-\beta^2)^{\frac{1}{2}}\left[-(E_z+c\dot{R}B_\varphi - c\dot{A}B_r)(1-\beta^2+\dot{R}^2+\dot{A}^2) + (E_r - c\dot{Z}B_\varphi + c\dot{A}B_z)\dot{Z}\dot{R}\right.$$

$$\left. + (E_\varphi + c\dot{Z}B_r - c\dot{R}B_z)\,\dot{Z}\dot{A}\right] .$$

Noting that $(1-\beta^2 + \dot{R}^2 + \dot{A}^2) = 1 - \dot{Z}^2$, we have finally

$$\ddot{Z} = \alpha(1-\beta^2)^{\frac{1}{2}}\left[-E_z(1-\dot{Z}^2)+ \dot{Z}\dot{R}E_r + \dot{Z}\dot{A}E_\varphi - c\dot{R}B_\varphi + c\dot{A}B_r\right] . \qquad (26)$$

The radial acceleration $\ddot{R}$, is given by

$$\Delta\ddot{R} = D_1(A_3C_2 - A_2C_3) + D_2(A_1C_3 - A_3C_1) + D_3(A_2C_1 - A_1C_2)$$

which becomes

$$(1-\beta^2)^2\,\ddot{R} = \left[-\alpha(1-\beta^2)^{3/2}(E_z + c\dot{R}B_\varphi - c\dot{A}B_r)\right]\times\left[(\dot{R}\dot{Z}\dot{A}^2 - \dot{R}\dot{Z}(1-\beta^2+\dot{A}^2)\right]$$

$$+\left[(1-\beta^2)\frac{\dot{A}^2}{R} - \alpha(1-\beta^2)^{3/2}(E_r - c\dot{Z}B_\varphi + c\dot{A}B_z)\right]\times\left[(1-\beta^2+\dot{Z}^2)(1-\beta^2+\dot{A}^2)\right.$$

$$\left.-\dot{Z}^2\dot{A}^2\right]+\left[-(1-\beta^2)\frac{\dot{R}\dot{A}}{R} - \alpha(1-\beta^2)^{3/2}(E_\varphi + c\dot{Z}B_r - c\dot{R}B_z)\right]\times\left[\dot{Z}\dot{R}\dot{A}\right.$$

$$\left.-\dot{R}\dot{A}(1-\beta^2+\dot{Z}^2)\right].$$

Simplified, the above equation yields

$$\ddot{R} = \alpha(1-\beta^2)^{\frac{1}{2}}\left[E_z\dot{Z}\dot{R} + c\dot{Z}\dot{R}^2B_\varphi - c\dot{Z}\dot{R}\dot{A}B_r - (E_r - c\dot{Z}B_\varphi + c\dot{A}B_z)(1-\beta^2+\dot{Z}^2+\dot{A}^2)\right.$$

$$\left.+E_\varphi\dot{R}\dot{A} + c\dot{Z}B_r\dot{R}\dot{A} - c\dot{R}^2\dot{A}B_z\right] + \frac{\dot{A}^2}{R}(1-\beta^2+\dot{Z}^2+\dot{A}^2) + \frac{\dot{R}^2\dot{A}^2}{R}.$$

Noting that $(1-\beta^2+\dot{Z}^2+\dot{A}^2) = (1-\dot{R}^2)$, we have finally

$$\ddot{R} = \alpha(1-\beta^2)^{\frac{1}{2}}\left[-E_r(1-\dot{R}^2) + E_z\dot{Z}\dot{R} + E_\varphi\dot{R}\dot{A} + c\dot{Z}B_\varphi - c\dot{A}B_z\right] + \frac{\dot{A}^2}{R}. \qquad (27)$$

The azimuthal acceleration $\ddot{A}$, is given by

$$\Delta\ddot{A} = D_1(A_2B_3 - A_3B_2) + D_2(A_3B_1 - A_1B_3) + D_3(A_1B_2 - A_2B_1)$$

which becomes

$$(1-\beta^2)^2\,\ddot{A} = \left[-\alpha(1-\beta^2)^{3/2}(E_z+c\dot{R}B_\varphi-c\dot{A}B_r)\right]\left[\dot{A}\dot{R}^2\dot{Z} - \dot{Z}\dot{A}\times(1-\beta^2+\dot{R}^2)\right]$$

$$+\left[(1-\beta^2)\,\frac{\dot{A}^2}{R} - \alpha(1-\beta^2)^{3/2}(E_r-c\dot{Z}B_\varphi+c\dot{A}B_z)\right]\times\left[\dot{A}\dot{R}^2\dot{R}-\dot{A}\dot{R}(1-\beta^2+\dot{Z}^2)\right]$$

$$+\left[-(1-\beta^2)\,\frac{\dot{R}\dot{A}}{R} - \alpha(1-\beta^2)^{3/2}(E_\varphi+c\dot{Z}B_r-c\dot{R}B_z)\right]\times\left[(1-\beta^2+\dot{Z}^2)(1-\beta^2+\dot{R}^2)\right.$$

$$\left.-\dot{Z}^2\dot{R}^2\right].$$

Simplified, the above equation yields

$$\ddot{A} = \alpha(1-\beta^2)^{\frac{1}{2}}\left[E_z\dot{Z}\dot{A} + c\dot{R}B_\varphi\dot{Z}\dot{A} - c\dot{A}^2B_r\dot{Z} + \dot{A}\dot{R}E_r - c\dot{Z}\dot{A}\dot{R}B_\varphi + C\dot{A}^2B_z\dot{R}\right.$$

$$\left.-(E_\varphi+c\dot{Z}B_r-c\dot{R}B_z)(1-\beta^2+\dot{Z}^2+\dot{R}^2) - \frac{\dot{A}^3\dot{R}}{R} - \frac{\dot{R}\dot{A}}{R}(1-\beta^2+\dot{Z}^2+\dot{R}^2)\right].$$

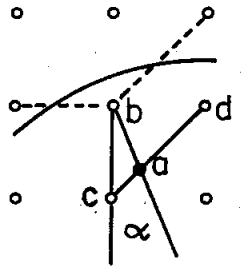
Noting that $(1-\beta^2+\dot{Z}^2+\dot{R}^2) = (1-A^2)$ we have finally

$$\ddot{A} = \alpha(1-\beta^2)^{\frac{1}{2}}\left[-E_\varphi(1-\dot{A}^2) + E_z\dot{Z}\dot{A} + E_r\dot{A}\dot{R} - c\dot{Z}B_r + c\dot{R}B_z\right] - \frac{\dot{R}\dot{A}}{R}. \qquad (28)$$

APPENDIX II

GENERAL NEUMANN BOUNDARIES

If a boundary with normal derivative equal to zero is as shown, then a problem boundary is drawn as shown by the dashed line. A point at "a" is chosen such that $V_a = V_b$. Point "a" is seen to lie on the normal to the boundary through the point "b" at the intersection between points "c" and "d". The slope of the boundary is given by tan α.



Starting from

$$V_a = V_b \tag{1}$$

we have

$$\frac{V_a - V_c}{\overline{ac}} = \frac{V_d - V_a}{\overline{ad}} \tag{2}$$

where, for example, $\overline{ac}$ is the distance from point "a" to point "c". The mesh interval is taken to be unity. Cross-multiplying, we have

$$\overline{ad}\, V_a - \overline{ad}\, V_c = \overline{ac}\, V_d - \overline{ac}\, V_a$$

or

$$(\overline{ad} + \overline{ac})\, V_a = \overline{ac}\, V_d + \overline{ad}\, V_c \,. \tag{3}$$

But, $\overline{ad} + \overline{ac} = \sqrt{2}$ and $V_a = V_b$, hence

$$\sqrt{2}\ V_b = \overline{ac}\, V_d + \overline{ad}\, V_c \,. \tag{4}$$

From the law of sines,

$$\frac{\overline{ac}}{\sin\alpha} = \frac{1}{\sin(\pi - \frac{\pi}{4} - \alpha)} = \frac{1}{\cos(\frac{\pi}{4} - \alpha)} = \frac{1}{\cos\frac{\pi}{4}\cos\alpha + \sin\frac{\pi}{4}\sin\alpha}$$

which becomes

$$\overline{ac} = \frac{\sqrt{2}\ \sin\alpha}{\sin\alpha + \cos\alpha} = \frac{\sqrt{2}\tan\alpha}{1+\tan\alpha} \ . \qquad (5)$$

Then the other segment is

$$\overline{ad} = \sqrt{2} - \overline{ac} = \sqrt{2}\left(1 - \frac{\tan\alpha}{1+\tan\alpha}\right) = \frac{\sqrt{2}}{1+\tan\alpha} \ . \qquad (6)$$

The complete difference equation from Eq. (4) is

$$\sqrt{2}\,V_b = \frac{\sqrt{2}\tan\alpha}{1+\tan\alpha}\,V_d + \frac{\sqrt{2}}{1+\tan}\,V_c$$

which in the notation used in the main text is

$$\frac{\tan\alpha}{1+\tan\alpha}\,V_1 + \frac{1}{1+\tan\alpha}\,V_4 - V_o = 0 \ . \qquad (7)$$