

## ELECTRON RAY TRACING PROGRAMS FOR GUN DESIGN AND BEAM TRANSPORT\*

W. B. HERRMANNSFELDT

*Stanford Linear Accelerator Center,  
Stanford, University, Stanford, CA 94309*

### ABSTRACT

Computer simulation of electron and ion sources is made by using a class of computer codes known as gun design programs. In this paper, we shall first list most of the necessary and some optional capabilities of such programs. Then we will briefly note specific codes and/or authors of codes with attention to specialized applications if any. There may be many more such programs in use than are treated here; we are only trying to cover a range of examples, not perform a comprehensive survey.

### CAPABILITIES AND LIMITATIONS OF GUN PROGRAMS

At the most basic level, an electron gun program should be able to accurately define the boundaries and the electric fields that result from imposing voltages or these boundaries. Ease of entering the boundary data is important but should not be so limited as to compromise the necessary versatility. Such details as isolated grid wires, dielectrics and shadow grids require significant amounts of information to be defined. Generally, there are two ways in which the needed detail can be expressed:

- 1) Through the use of a regular square or rectangular mesh with interpolated fractional meshes to define boundaries, and
- 2) Through the use of a deformable, triangular mesh.

Most of the paper will concentrate on specific methods used in the program EGUN.<sup>1</sup> We will discuss the mathematical algorithms and discuss their implementation with respect to an example of a Pierce diode.

A gun program, as distinguished from other beam transport programs, must be able to treat longitudinal space-charge effects and, in particular, must be able to calculate the space-charge limited current of an emitter.

Any charged particle transport program must be able to accurately integrate individual particle motion through the defined fields. A gun program must do this from very low velocities, typified by the emission of ions, through

---

\*Work supported by the Department of Energy, contract DE-AC03-76SF00515.

fully relativistic velocities. Not only must the equations of motion be completely defined, with no approximations, but also the mathematical methods must not break down, especially at very low velocities.

Beam transport calculations should include the effects of self-fields, both space-charge and self-magnetic fields. There are essentially two ways in which this is typically done:

- 1) by calculating the forces on each individual particle, or ray, from every other ray, or
- 2) by solving for Poisson's equation with space-charge and separately calculating the self-magnetic field term.

The first of these has the advantage that, for relativistic beams, the self-fields nearly cancel each other, to  $(1 - v_z^2/c^2)$ , and thus the method is intrinsically more accurate. This method, however, lacks the flexibility required to perform many of the functions of a general purpose program. This is because it is necessary to keep particles in step with each other, which is not always possible as, for example, in the case of a depressed collector. A third approach, which effectively combines the above two, is to reduce the space-charge by  $(1 - v_z^2/c^2)$  when there are no significant longitudinal fields, e.g., in the transport of an intense beam through a drift pipe. This makes it possible to use the second method near the cathode, and since particles are not usually highly relativistic there, this approach is generally quite accurate.

Externally imposed magnetic fields play a critical role in many electron guns and transport systems. In some ways this subject is larger than the subject of electron guns themselves. A recipe for the design of a magnetically focused device is as follows:

- 1) Make a preliminary test without a magnetic field.
- 2) Perform the actual design using an ideal magnetic field that can perform the needed transport function. This field can be expressed in any form that is compatible with Maxwell's equations. For example, if the field on the axis is chosen, it can be expressed in terms of ideal point coils, or ideal solenoids, thus avoiding problems inherent with off-axis expansions.
- 3) Convert the ideal field to a form that can be built using realistic magnetic elements, aided by a magnet design program.
- 4) Take output from the magnet design program and use it to check the effects of the realistic fields on the beam transport problem.

From the above, it follows that the ray tracing program should have a variety of options for specifying magnetic fields including a simple uniform field, ideal coils, and by accepting results from a magnet design program.

Optional capabilities are usually designed to fulfill objectives of a specific application. Some of these include:

- 1) treating plasma discharge ionization sources including finding the emission surface of a plasma,
- 2) including neutralization, second species or secondary emission,
- 3) operating in other coordinate systems besides the usual two-dimensional cylindrical or rectangular coordinates. This includes full three-dimensional as well as mixed round beams in rectangular coordinates or round beams offset in cylindrical coordinates.
- 4) Optional capabilities also include other applications of Laplace's Equation such as finding peak surface fields, calculating capacitance and accounting for dielectrics.

We should also note some of the limitations of electron gun programs of this type (we are not here dealing with fully electromagnetic particle-in-cell programs):

- 1) Two-dimensional calculations of electric fields: either cylindrical or rectangular symmetry. In cylindrical coordinates, a cylindrically symmetric beam is propagated along the axis. In rectangular coordinates, both the electrodes and the beam extend infinitely far in the directions normal to the "plane of the paper" on which the problem is shown. In both symmetries, the nominal direction of propagation of the beam lies in the "plane of the paper" but transverse motion is allowed. Thus, for example, the spiral motion of a beam in an axial magnetic field can be simulated.
- 2) Time independence: these are dc calculations after the beam has reached "steady state." A common characteristic of such programs is that if steady state cannot be achieved, for example, if an attempt is made to propagate a beam beyond the space-charge limit, then the programs will not converge to a satisfactory steady solution. Under such conditions, one is not justified in claiming any physical reality for the results.
- 3) Idealized computer models: the nature of modeling programs is to ignore various real complications. Such things as tolerances out of cylindrical symmetry, stray electrons or ions, partially poisoned cathodes, etc., may play large parts in any real device but are usually ignored in models. Other aspects of models; finite elements, numbers of trajectories, discrete iteration steps, etc., may also affect the accuracy of the results. One should not expect a computer code to yield exactly correct predictions of operating parameters. One should expect that the effects of varying input parameters, particularly for small perturbations, should be reliably reflected in changes in the real device. Of course, some predictions are

better than others; for example, the EGUN<sup>1</sup> program typically predicts gun perveance correctly to within a few percent but has a somewhat harder time in predicting beam diameter for a high intensity beam.

It is useful to examine the limitation on time independence. If an electron gun is suddenly turned on (for example, with a laser photocathode), the initial burst of current can substantially exceed the space-charge limited current, and the kinetic energy of the beam can exceed the product  $eV$ , where  $V$  is the diode voltage. The electron optics of the front of the pulse closely resembles the electrostatic solution without space-charge. Later on in the pulse, the conditions will converge to the space-charge limited solution, assuming that the cathode can emit adequate current. The time constant for this transition is typically the transit time of the space-charge limited particles across the gap of the diode, at most a fraction of a nanosecond for electrons. Accurate simulation of pulses shorter than the transit time must use the particle-in-cell approach, while pulses longer than the transit time can use the more economic electrostatic programs.

## CHARACTERISTICS OF DIFFERENT PROGRAMS

Since there are several authors of gun design programs at this conference, it will not be necessary to include much detail about other programs. The program SNOW, written by Jack Boers<sup>2</sup> of Varian Associates, is predominantly a program for the design of ion sources. Boers is reporting on the development of a three-dimensional version of SNOW at this meeting.

The Darwin model of solving for electromagnetic fields has been used by John Boyd<sup>3</sup> of LLNL to develop a program called DPC, for Darwin Particle Code. This program bridges a gap between electrostatic programs, of the type discussed here, and the fully electromagnetic PIC codes. The Darwin method relaxes the so-called "Courant Condition" for electromagnetic programs and is especially useful for applications involving pulses of a few tens of nanoseconds in length.

The best example of the triangular mesh approach is in a program by Richard True<sup>4</sup> of Litton Industries. This is a general purpose program which is particularly useful for the design of high area-convergence guns, such as for TWT's, because of the feature allowing the mesh to be concentrated near the axis.

The program EBQ, written by Art Paul<sup>5</sup> of LLNL, uses the direct cancellation of self-magnetic and space-charge forces and is thus especially suited for intense beam transport.

A pair of general purpose ion and electron source programs from GSI have been reported by Spädke.<sup>6</sup> They are AXCEL-GSI and KOBRA3, the latter being a three-dimensional code.

## EGUN

We will now briefly describe the organization of the program EGUN<sup>1</sup> and also describe in some detail some of the important algorithms. In operation, the program starts by reading and checking the input boundary data. The program then solves Laplace's equation, i.e., Poisson's equation without space-charge. The result of this calculation, together with all the boundary information is then printed.

The Poisson solver in EGUN is a column matrix inversion routine that compares favorably in speed to conventional point-by-point over relaxation schemes. Boundary interpolation, both for the Poisson solver and for partial differentiation of potentials near boundaries, is based on

$$V_w = V_p + \frac{V_b - V_p}{\Delta x} \quad ,$$

where  $V_w$  = potential of a mesh point behind the boundary,

$V_p$  = potential of a mesh point nearest to the boundary,

$V_b$  = potential assigned to the boundary,

$\Delta x$  = vector distance to the boundary from the point at 'p'.

Next, the first iteration of electron trajectories is started. These are initiated by one of four schemes:

1. "GENERAL" cathode in which electrons are started assuming Child's law holds near a surface designated as the cathode. This surface can be of any arbitrary shape and may include holes and shadow grids.
2. "SPHERE" for a spherical cathode (cylindrical in rectangular coordinates) in which the electrons are assumed to be emitted at right angles to the surface defined by a radius of curvature and a radial limit. Child's law for space-charge limited current is again used.
3. "CARDS" in which the specific starting conditions for each ray are specified in an 80-column card format.
4. "GENCARD" which combines the versatility of "CARDS" with the assumptions of Child's law from "GENERAL." This is especially useful for cases involving very nonuniform current emission.

The three methods for initiating space-charge limited flow all include a Busch's Law calculation to account for magnetic flux through the cathode.

On the first iteration cycle, space-charge forces are calculated from the assumption of paraxial flow. As the rays are traced through the program, space-charge is computed and stored in a separate array. After all the electron trajectories have been calculated, the program begins the second cycle by solving Poisson's equation with the space-charge from the first cycle. For problems meeting the paraxial assumptions, especially if relativistic electron beams are involved, this one cycle may be sufficient to solve the entire problem. For other problems in which space-charge is negligible, e.g., spectrometers and phototubes, a single cycle is usually adequate.

Subsequent iteration cycles (as many as are requested) follow the above pattern. The Child's law calculations for the starting conditions are remade by averaging the perveance used for the previous cycle with the perveance calculated directly from the solution of Poisson's equation.

An additional starting option is "LAPLACE" intended for any application of Laplace's equation not involving electron ray tracing. In this case the number of cycles is used simply to improve the accuracy of the solution of Laplace's equation. The "LAPLACE" option includes a provision for inputting arbitrary data in the "space-charge" array. The output from LAPLACE includes a list of the fields on the entire boundary. This can be used to find local peak field strengths and to calculate the capacity of part or all of some configuration.

The program always operates in two dimensions; either R and Z in cylindrical coordinates or Y and X in rectangular coordinates. The rectangular coordinate output retains the R and Z labels, however. Electron orbits are calculated through azimuthal changes (labeled "PHI") referenced to the Z axis. In rectangular coordinates, PHI is actually the third Cartesian coordinate.

EGUN uses a four step 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 auxiliary 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

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

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

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

where

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

The constant  $\alpha = e\lambda/m_0c^2$  where  $e$  is the charge on the particle,  $m_0c^2$  is the rest energy of the particle, and  $\lambda$  is the constant of proportionality between the real coordinates and the dimensionless coordinates. Thus,

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

By an arbitrary choice,  $\lambda = 5.11 \times 10^5$  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.<sup>2</sup> Then,  $cB$  has units of volts per meter. To convert to program fields of volts per mesh unit, magnetic fields are multiplied by the value UNIT in meters per mesh unit. Magnetic field input to the program is in gauss, which is the common engineering unit, and is internally converted to webers/meter<sup>2</sup>.

The azimuthal magnetic field  $B_\phi$  comes from the current in the electron beam and is called the self-magnetic field of the beam. The magnetic field created by an axial current is

$$B_\phi = \frac{\mu_0 I}{2\pi r} \text{ webers/meter}^2 \quad .$$

The field is assumed to be due to an infinite conductor which is a good approximation in the area in which the field is significant. After multiplying  $B_\phi$  by the scale factor and expressing  $r$  in meters which requires multiplying  $r$  by the scale factor also, the scale factor cancels as might be expected. Thus, the scale factor only enters for external magnetic fields. The current  $I$  is the summation of the current in the trajectories at lower radii than the trajectory being calculated, but including the one 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 in most practical cases and so is neglected.

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

$$\nabla^2 V = \frac{\rho}{\epsilon_0} = \frac{J}{v\epsilon_0} .$$

The element of area for  $J$  is  $r \times 1.0$  square mesh units where  $r$  is the particle radius. The velocity is only the Z-component since the space-charge is being spread between adjacent points on the same column. The one mesh unit space between adjacent points accounts for the 1.0 in the area expression above.

In the finite difference form, the right-hand side becomes

$$RO = \frac{36\pi \times 10^9 I(K) \times 10^{-6}}{ABS(ZDOT) \times 3 \times 10^8} = \frac{(3.77 \times 10^{-4}) I(K)}{ABS(ZDOT)} ,$$

where  $RO$  is to be spread between two adjacent mesh points in inverse ratio to the distance from the ray to each point,  $I(K)$  is the current in the one radian segment of the ray (in microamperes) and  $ZDOT$  is the velocity in units of  $c$ . If the angle of inclination,  $dR/dZ$ , exceeds  $45^\circ$ , the calculation is made for  $RDOT$ . The absolute value of  $ZDOT$  is used to allow a negative  $ZDOT$ . The explicit value of  $R$  is canceled by the  $R$  which would convert the current to current density, thus avoiding special problems as  $R \rightarrow 0$ .

In practice, however, there are still some space-charge problems near the axis. In rectangular coordinates, if the axis is a plane of symmetry, then any trajectory between  $R = 0$  and  $R = 1$  has a mirror image between  $R = 0$  and  $R = -1$ . (A reminder...when in rectangular coordinates, the axis still retain their cylindrical labels.) To account for all the space-charge on the axis, the calculated charge is doubled. In cylindrical coordinates, the algorithm for distributing the space charge proportionately to the distance between the adjacent points is not a very accurate solution within one mesh unit of the axis. Good smooth laminar flow near the axis results by simply making the space-charge on the axis equal to that found for the first row.

Magnetic fields, except for the self-magnetic field of a beam, are input directly in one of three ways:

1. by specifying the field along the Z-axis,
2. by specifying a set of coils (giving position, radius and current), or
3. by using the vector potential output from a magnet program such as Poisson. It is interesting to note that Colman<sup>7</sup> has converted several accelerator physics programs including Poisson to run on the IBM-AT.

In cylindrical coordinates, the field is interpreted as an axial magnetic field with radial terms as required by Maxwell's equations. The off-axis fields can be made by either a sixth-order expansion from the axial fields or, for the case



of a set of coils, by directly using the appropriate elliptic functions. When the vector potential input has been used, local interpolation is used in place of the expansion.

In rectangular coordinates the magnetic field can be defined to be principally in any one of the three Cartesian directions. Off-median plane expansions are made in the direction of the field on the median plane. If the median plane is the R-Z plane, then the field is in the PHI direction and the field extends to infinity in the R-direction. This fits the configuration of the pole face of a dipole magnet. (Remember that R, Z and PHI are here taken to be orthogonal Cartesian coordinates.) If the median plane lies normal to the plane of the problem, through the Z-axis, then the field extends to infinity in the PHI direction. In this case, the direction of the field on the median plane can be either in the Z-direction or in the R-direction, depending on the symmetry of the coils that produce the field. The off-median-plane expansions in rectangular coordinates satisfy Maxwell's equations to second order.

Self-magnetic fields are calculated for both coordinate systems from the current in the rays on the present cycle. A built-in sort routine insures that the rays are sequentially numbered from the axis outwards. The self-magnetic field calculation assumes all the current from the previous rays lies on the axis in an infinitely long conductor. If the ray being calculated crosses the last preceding ray, then the current from that ray is dropped. If the ray continues to cross other rays, then the current from those rays is only dropped if the ray goes below the minimum radius of a previous ray. Note that if the self-magnetic field is very significant, then almost by definition, one is dealing with an intense relativistic beam. This problem is generally better suited to the paraxial ray approach, as solved in the first cycle, in which the space-charge is offset by the self-magnetic field directly, rather than by the offsetting effects of two large terms. Best results can be obtained for such problems if the electron gun region can be separated from the drift and focusing regions in which the self fields are so important. For cases where the beam is already relativistic in the gun, a new option allows the user to define a kinetic energy above which the direct cancellation of space-charge by the self magnetic field is used, as described earlier, instead of the normal separate terms. This permits the Child's Law calculation to be used near the cathode and the paraxial calculation to be used when the beam is at higher energy.

In rectangular coordinates, the self-magnetic field assumes symmetry about the  $Y = 0$ , ( $R = 0$ ) plane. If this is not correct, or if for other reasons it is desired to turn off the self-magnetic field, then an external field of strength zero can be specified. In any case, in rectangular coordinates, the self-magnetic field functions only if there is no external field.

A single variable controls plotting. Normally, at least the last cycle is plotted. The first cycle may also be plotted or one may even plot every cycle. Ray tracing plots may include equipotential plots, either separate or overlaid with the trajectory plots. Figure 1 is an example of the graphic output showing a Pierce diode with equipotential lines and trajectory paths. If there is an external magnetic field, then this field is also plotted, overlaid on the trajectory plots. A feature that is especially useful if a magnetic field is present, is an option that allows one to choose a single trajectory for which the azimuthal position PHI, is plotted as a function of Z. Finally, there are a pair of simple plots; current density versus radius and alpha versus radius, where  $\text{Alpha} = \arctan dR/dZ$ . The latter plot is equivalent to a phase-space plot.

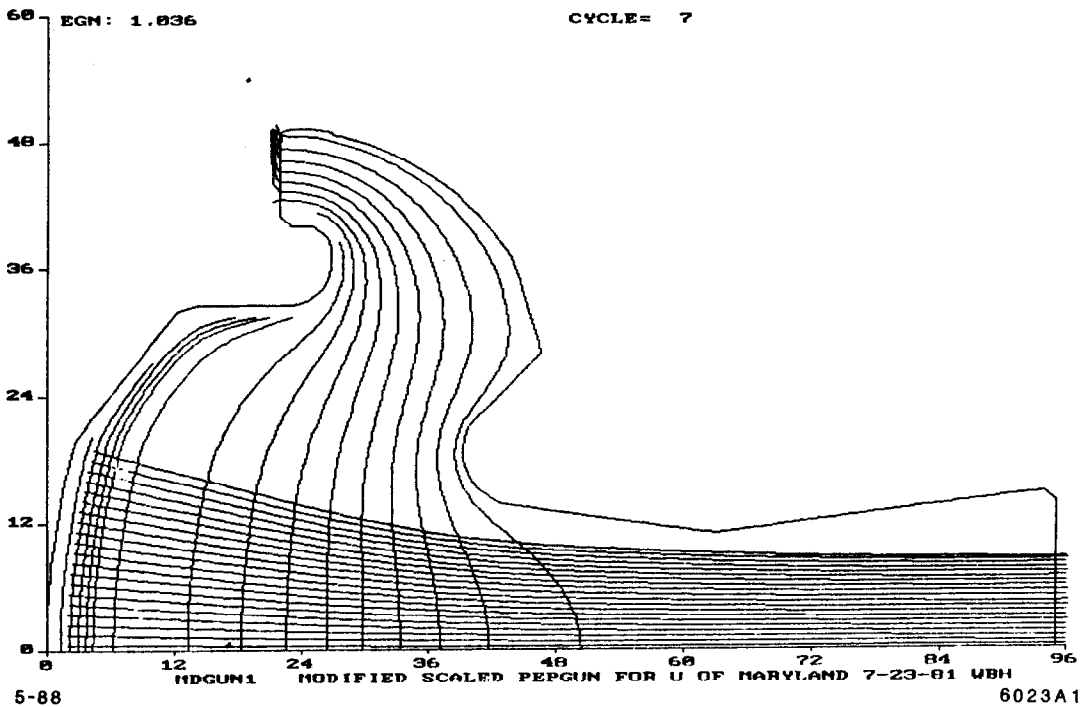


Figure 1. The computer drawn simulation of a Pierce diode, as described in the text. This particular sample was plotted on a dot matrix printer using data sent directly from the PC, not using a screen dump routine.

## PERFORMANCE

EGUN has been available for many years in various Fortran versions and is now available from the author as EGN87c, written in C and including an integrated plotting package for the IBM class of PC's. The problem shown in Fig. 1 required 22 seconds to run on the IBM 3080 and 23 minutes on an AT-equivalent machine with 8 MHz clock. The PC had the Intel<sup>8</sup> 80287 (R) Math co-processor installed and 1.1 Mbyte storage, which permits writing the program output directly to virtual disk space. This factor of roughly 60 between the PC and the IBM 3080 seems to hold for a variety of configurations for this program. Faster AT clones and PS-2 machines, both based on the 80286 processor, reduce this time by about half while 80386 based machines with the 32-bit architecture reduce the time a further factor or two to three. IBM XT-type PC's based on the 8086 processor are only about 50% slower.

As usual with a space-charge-dominated configuration such as this, the problem is run for several cycles in order to achieve convergence. In this case, the problem iterated for seven cycles which has long been the default standard for EGUN. There are several convergence processes going on simultaneously in the program. First there is the solution to Poisson's equation, which is the only one that is even remotely guaranteed mathematically to converge. Then there is the convergence of the Child's Law problem on the cathode. This can be divided into two parts, local and global. The global convergence of perveance is achieved by constantly averaging the perveance found by solving Poisson's equation, with the perveance used in the previous cycle. The perveances found on successive cycles for the problem in Fig. 1 illustrate this process. The numbers (all in units of microperveance, defined as  $IV^{-3/2} \times 10^6$ ) are: 1.06, 0.84, 0.71, 0.70, 0.76, 0.76, 0.75. The final value of 0.75 agrees to within the expected tolerance of the observed value for this tube of 0.70. The perveance value before averaging is obtained by defining a starting surface at least two mesh units in front of the cathode and calculating the current that would flow to that surface given the voltage found on the surface by solving Poisson's equation. For the first cycle, this initial value of the perveance is then divided by two. The user can specify an initial value of perveance, and can also request that this value be maintained for a specified number of cycles. This is one way in which temperature limited emission can be modeled. Sometimes knowing the desired perveance helps the program to converge faster, but the method described above, dividing the no-space-charge value by two, frequently converges faster than the alternatives.

Usually the local perveance convergence, that is, convergence of local current density along the cathode surface from cycle to cycle, follows the global calculation without showing hot spots. Exceptions may occur for particularly "bad" designs, or for cases when the starting surface is too close to the

cathode. It is necessary to define a starting surface that is at least two, and preferably three mesh units from the cathode.

Another convergence process concerns the final beam configuration. If there are enough rays, i.e., at least one per mesh unit on the starting surface, then usually the seven cycles are enough to give radial convergence to the particle distribution. More cycles may be needed if the problem concerns intense relativistic beams for the reasons discussed earlier. Guns requiring high area convergence ratios may also require more program cycles to find a self-consistent solution to the beam diameter. Strong magnetic fields somewhat aid in the convergence, just as they add stability to the beam in practice. The normal criterion for stability is to examine the output to see whether there is much change from cycle to cycle. The program always reduces the iterative step by a factor of two on the last cycle and also reduces the error criterion for the Poisson solver by a factor of ten. These two measures test for robustness of the solution to calculational limits. In the example of the problem in Fig. 1, the beam radius after seven cycles was 8.7 mesh units. After six it was 8.8 mesh units, a small enough difference to consider the problem converged.

## DIAGNOSTICS

The program contains three classes of diagnostics:

1. Input data diagnostics, particularly for the NAMELIST entries, which must correspond to the list of expected elements, and for the boundary data that must fulfill a variety of criteria to be used to determine difference equations for the Poisson solver. Boundary diagnostics include definite errors that would cause the problem to fail, and warnings about data points that do not fit some predetermined criteria of what boundary data should look like, but which may actually be correct. The user should of course be certain that he fixes or clearly understands why each warning is given. There are also fairly extensive diagnostics provided for magnetic field input data, including a plot of the axial field profile that can be plotted at any chosen radius (but usually on the Z-axis), and a table of the on-axis and off-axis magnetic fields.
2. Program operation diagnostics, including messages about the progress of EGN sent to the PC terminal, tabulation of the time spent in each subroutine and the time required for each program cycle, as well as the total time. Convergence data for the Poisson solver is printed on-line and in the listing.

3. Physics diagnostics including tables of potentials and space-charge, and initial and final data for each trajectory, with an option of printing each iteration step from up to six trajectories. The final trajectory information is used to calculate the emittance of the beam.

Even though, by its very nature, EGUN is not a statistical program, aberrations due to nonlinear fields and nonuniform space-charge distributions cause the beam to fill an area in phase space from which emittance can be derived. For the problem illustrated in Fig. 1, the calculated normalized emittance area is about  $10\pi$  mm-mrad. This is an especially low value for such a gun, which shows that this is an especially good design. For a uniform beam, which is the ideal of an electron gun, the normalized emittance is

$$\epsilon_n = 4\beta\gamma(\langle x^2 \rangle \langle x'^2 \rangle - \langle x \times x' \rangle^2)^{1/2} ,$$

where  $x' = dx/dz$ , the  $\langle \rangle$  brackets signify weighted averages for all the particles, and  $\beta\gamma$  is the relativistic velocity times the relativistic mass. It is the  $\beta\gamma$  product that makes this the "invariant" or "normalized" emittance, as it is usually called. It is invariant because subsequent acceleration does not cause the emittance area to change.

## ACKNOWLEDGEMENTS

We wish to acknowledge the continued support and encouragement of Dr. Roger Miller (SLAC). The origin of EGUN is based on the work of Vladimir Hamza<sup>8</sup> at Stanford University. Finally, our thanks go to all of the users of EGUN, many of whom have made helpful suggestions and provided important benchmarks to test the program.

## REFERENCES

1. W. B. Herrmannsfeldt, SLAC Electron Trajectory Program, SLAC-226, November 1979. Also available as a revised version in draft form, January 1988.
2. Jack Boers, SNOW and SNOW 3-D, papers at this conference.
3. John Boyd and Dennis Hewett, Bull. Am. Phys. Soc. **29**, 1436, (1984).
4. Richard True, IDEM Technical Digest, pp. 257-60, December 1985.
5. A. C. Paul and V. K. Neil, IEEE Trans. on Nucl. Sci. **NS-26**, No. 3, June 1979.
6. Peter Spädke, Computer Simulation of High-Current DC Ion Beams, Proceedings of the 1984 Linac Conference, Seeheim.
7. Judith Colman, The Use of the IBM PC Computer in Accelerator Design Calculations, Proceedings of the 1986 Linear Accelerator Conference, Stanford Linear Accelerator Center, Stanford, CA 94309, June 1986. Ms. Colman's address is Neutral Beam Division, Brookhaven National Laboratory, Upton, NY 11973.
8. Intel is a registered trademark of the Intel Corporation.
9. Vladimir Hamza et al., NASA TN-1323, TN-1665, TN-1711, Lewis Research Center, Cleveland, Ohio (1962).