RPN75

RADIATION PROJECT PROGRESS REPORT NO. 3

Computer-Assisted Magnetic Field Design

Part I

by

Richard T. Close

Electron Beams Branch
Plasma Physics Division

7 March 1966

Computer-Assisted Magnetic Field Design

by

Richard T. Close

7 March 1968

I. INTRODUCTION

The purpose of this note is to provide documentation on the procedures now being used to design an electrode configuration that will provide satisfactory magnetic fields for electron trapping in circular orbits. The material is not original but since it exists in a number of scattered references, there appears to be some merit in gathering it in one place. Section II of this note deals with the basic equations that must be solved and the parameters of interest in magnetic field design. This material is essentially classical electrodynamics and should be familiar to the technical reader. Section III deals with the computer program that has been written to perform the necessary calculations. It outlines the relaxation technique employed and enumerates the various features now available in the program. A separate note has been prepared which gives specific instructions on the use of the computer programs.

II. BASIC EQUATIONS FOR FIELD DESIGN

To describe the steady state magnetic field $\vec{H}$ set up by a current density $\vec{j}$, the field equations:

$$\nabla \times \vec{H} = \frac{4\pi}{c} \vec{j} \tag{1}$$

$$\nabla \cdot \vec{B} = 0 \tag{2}$$

are used. In view of equation (2) it is possible to obtain the magnetic field from a vector potential function $\vec{A}$ defined by:

$$\vec{B} = \nabla \times \vec{A} \ . \tag{3}$$

For the axially symmetric case in cylindrical coordinates, it is only necessary to specify the θ component of the vector potential to obtain the radial and longitudinal magnetic fields. In this case the fields are given (in a vacuum, $\vec{B} = \vec{H}$) by:

$$H_r = - \frac{\partial A_\theta}{\partial z} \tag{4}$$

and

$$H_z = \frac{1}{r} \frac{\partial (rA_\theta)}{\partial r} \ . \tag{5}$$

The equation that A_θ obeys is obtained by substituting equation (3) into equation (1). Here there is really not much value in expanding the vector products involved since the apparently simple relation obtained ($\nabla^2 \vec{A} = \frac{4\pi}{c} \vec{j}$) is fairly complicated in anything but rectangular coordinates. It is better to work in generalized curvilinear coordinates to insure that the gradient operation on unit vectors is correctly performed. This being the case, one evaluates the θ component as:

$$\left[\nabla \times (\nabla \times \vec{A}) \right]_\theta = \frac{\partial^2 (rA_\theta)}{\partial z^2} + r \frac{\partial}{\partial r} \left(\frac{1}{r} \frac{\partial (rA_\theta)}{dr} \right) . \tag{6}$$

It is convenient to define a function Ψ such that:

$$\Psi = rA_\theta \ . \tag{7}$$

In a current free region, Ψ obeys the elliptic partial differential equation:

$$\frac{\partial^2 \Psi}{\partial z^2} + r \frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial \Psi}{\partial r}\right) = 0 \ . \tag{8}$$

The design of a suitable magnetic field configuration is reduced to the solution of equation (8) for various specified electrode shapes. It is instructive to formulate equations (4) and (5) in terms of Ψ.

$$H_r = -\frac{1}{r}\frac{\partial \Psi}{\partial z} \tag{9}$$

$$H_z = \frac{1}{r}\frac{\partial \Psi}{\partial r} \ . \tag{10}$$

From equation (10) one obtains:

$$\Psi = rA_\theta = \int_0^r rH_z \, dr \ . \tag{11}$$

This relation allows the identification of Ψ with the flux. More precisely the flux Φ is given by:

$$\Phi = 2\pi \Psi \ . \tag{12}$$

Once the fields as given by equations (9) and (10) have been obtained, the current I (in emu) can be calculated from equation (1).

$$\int \nabla \times \vec{H} \cdot d\vec{A} = 4\pi \int \vec{J} \cdot d\vec{A} \quad .$$

From Stokes' theorem this becomes:

$$\oint \vec{H} \cdot d\vec{\ell} = 4\pi I \quad . \tag{13}$$

The current I is then simply related to the line integral of $\vec{H}$ about any closed path encircling the electrode system. From the current one may calculate the inductance and stored energy of the system. In practice one usually sets the flux function Y equal to unity on the metallic boundaries of the electrode configuration. In this case, the inductance L is simply:

$$L = \frac{\Phi}{I} = \frac{2\pi Y}{I} = \frac{2\pi}{I} \quad . \tag{14}$$

Similarly, the stored energy E becomes:

$$E = \frac{1}{2} LI^2 = \pi I \quad . \tag{15}$$

Finally the quantity that determines the stability of the electron beam in the magnetic field is of interest. This quantity known as the field index n is defined as:

$$n = -\frac{r}{H_z} \frac{\partial H_z}{\partial r} \tag{16}$$

along the median plane. For values of n less than unity the field should be suitable for the trapping of particles in stable circular orbits.

The problem is then clear. One must solve equation (8) subject to various specified boundary conditions to determine the flux function Y. Having obtained Y, the field components H_x and H_z are easily calculated. Finally, one can obtain the inductance, stored energy and field index which go to completely describe the magnetic field configuration. After a few computer runs it should be possible to devise a suitable system of electrodes with the characteristics desired. In this regard, one would like to minimize the stored energy E and possibly arrange that the variation in field index follow a particular form.

III. COMPUTER ALGORITHM FOR THE SOLUTION OF THE MAGNETIC FIELD DESIGN PROBLEM

Since equation (8) is an elliptic partial differential equation, some sort of relaxation method is indicated for its numerical solution. In preparing such a problem for solution one draws out the electrode structure and then superimposes a finite difference mesh. The spacing between nodal points in the mesh is set essentially by the size of the fast memory of the computer being used. For most purposes a mesh containing about 1000 nodes has been found to be sufficient. If the need arises, however, it is probably possible to double or even triple this number and still remain within the limits of the usual 32K machines. In regard to the actual number of nodes used, there are several considerations. Intuitively one feels that a very fine mesh is the most accurate but a definitive theorem on this subject is not available. Further a very fine mesh sometimes leads to unstable solutions and/or extremely long running times. On the other hand, on relatively naive grounds the maximum accuracy obtainable is roughly one over the total number of nodes. Thus, no hard rules can

be given for the positioning of the finite difference net. In practice it is usually a simple matter to follow an experimental approach to this problem. After several computer runs a workable mesh normally becomes evident.

The numerical procedure is to solve a finite difference equation at every nodal point. The exact form of the finite difference approximation depends on the method used and the accuracy desired. With reference to figure 1, one can arrive at truncated expressions for the derivatives at point zero of the mesh by considering a Taylor series expansion. Neglecting terms containing derivatives of order greater than two one finds:

$$\left(\frac{\partial Y}{\partial x}\right)_0 = \frac{1}{h_3 h_4 (h_3 + h_4)\,\Delta x}\left\{h_3^2 Y_2 - h_4^2 Y_4 - (h_3^2 - h_4^2) Y_0\right\}$$

$$\left(\frac{\partial^2 Y}{\partial x^2}\right)_0 = \frac{1}{\Delta x^2 h_3 h_4 (h_3 + h_4)}\left\{h_4 Y_2 + h_3 Y_4 - (h_3 + h_4) Y_0\right\} \qquad (17)$$

$$\left(\frac{\partial Y}{\partial z}\right)_0 = \frac{1}{h_1 h_2 (h_1 + h_2)\,\Delta z}\left\{h_2^2 Y_1 - h_1^2 Y_2 - (h_2^2 - h_1^2) Y_0\right\}$$

$$\left(\frac{\partial^2 Y}{\partial z^2}\right)_0 = \frac{2}{\Delta z^2 h_1 h_2 (h_1 + h_2)}\left\{h_2 Y_1 + h_1 Y_2 - (h_1 + h_2) Y_0\right\}$$

If one then substitutes equations (17) into equation (8) and solves for Y_0, the result is:

$$Y_0 = C_1 Y_1 + C_2 Y_2 + C_3 Y_3 + C_4 Y_4 \qquad (18)$$

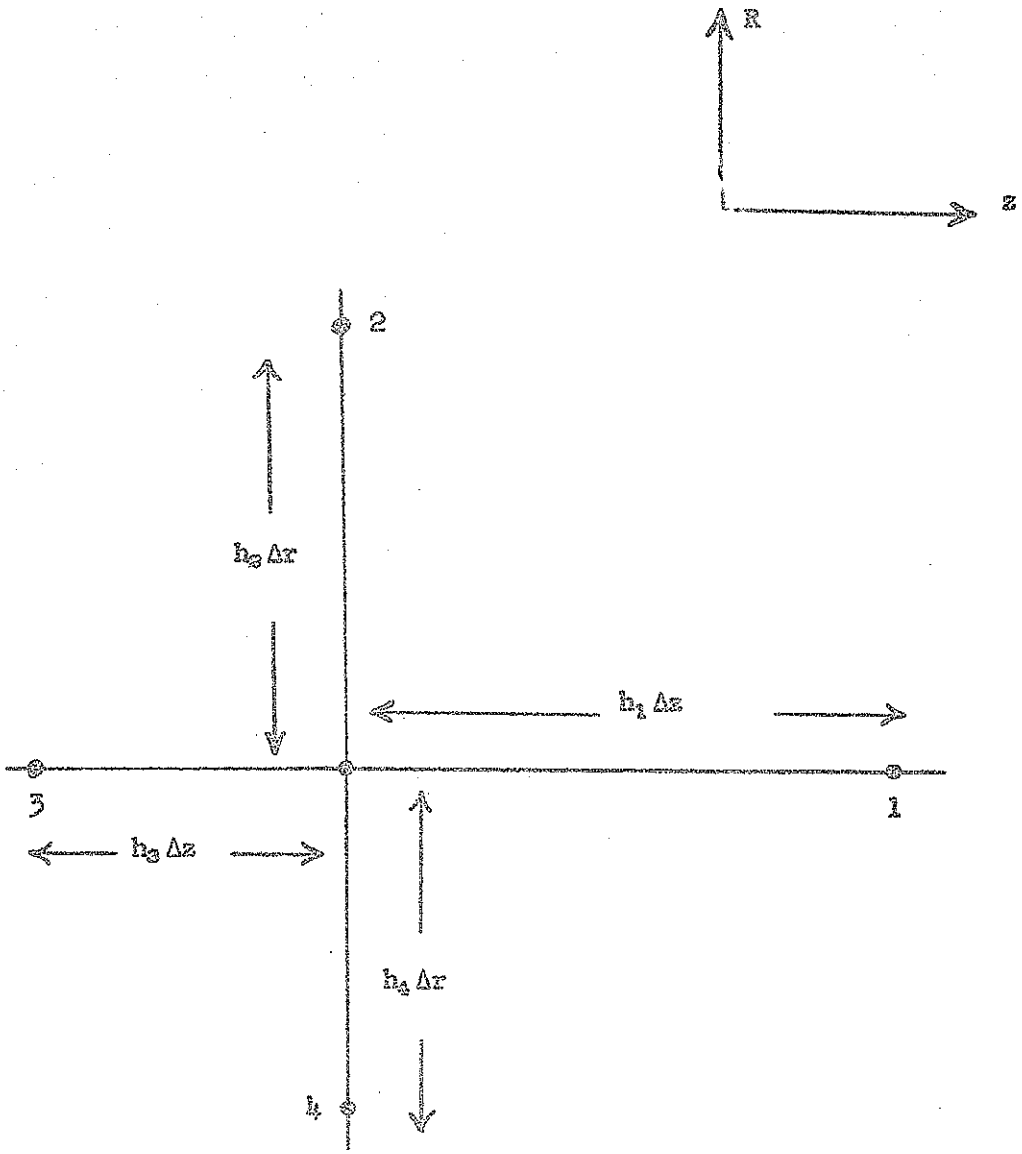

Figure 1

where
$$C_1 = \frac{2\lambda h_3}{K} \tag{19}$$

$$C_2 = \frac{\mu h_4}{K}\left(2 - \frac{\Delta r}{r} h_4\right) \tag{20}$$

$$C_3 = \frac{2\lambda h_1}{K} \tag{21}$$

$$C_4 = \frac{\mu h_3}{K}\left(2 + \frac{\Delta r}{r} h_3\right) \tag{22}$$

$$\lambda = \frac{1}{\Delta z^2\, h_1 h_3 (h_1 + h_3)} \tag{23}$$

$$\mu = \frac{1}{\Delta r^2\, h_3 h_4 (h_3 + h_4)} \tag{24}$$

$$K = \mu(h_4 + h_3)\left\{2 - \frac{\Delta r}{r}(h_4 - h_3)\right\} + 2\lambda(h_3 + h_1) \quad . \tag{25}$$

Hence, it is seen that the flux function Ψ is given at every node of the mesh by equation (18) with the coefficients $\{C_i\}$ as given by equations (19) through (25). The numerical procedure repeatedly solves equation (18) at each node until the solution relaxes. That is, when the maximum change in Ψ at any mesh point is less than a prescribed amount, the calculation of Ψ ceases and the program moves on to other computations.

For equation (8) it is not critical, but for elliptic-type equations in axially symmetric geometries one must specify an additional boundary condition along the axis of symmetry ($r=0$). For this case the condition is that:

$$\left(\frac{\partial \Psi}{\partial z}\right)_{r=0} = 0 \quad . \tag{26}$$

Of course, since Y is defined as rA_θ, this boundary condition is redundant in the sense that it simply reproduces zero values of Y along the axis. For other elliptic equations, for example, Laplace's equation, conditions such as equation (26) become very important and care must be used in dealing with them.

If the relaxation method indicated by equation (18) were to be used as it stands, one would expect that the procedure would converge rather slowly. A method known as the extrapolated Liebmann method or the method of successive over-relaxation[1] can be used to speed up the problem. Using this procedure, one modifies equation (18) to read:

$$Y_o^{K+1} = Y_o^K + \beta\left(C_1 Y_1^K + C_2 Y_2^K + C_3 Y_3^{K+1} + C_4 Y_4^{K+1} - Y_o^K\right) . \qquad (27)$$

In equation (27) the superscripts refer to the K^{th} and $(K+1)^{th}$ iteration and the quantity β is called the accelerating factor. In all but the simplest cases β must be specified or determined in some empirical way. According to theory, equation (27) should converge for:

$$0 < \beta < 2 . \qquad (28)$$

In practice it has been found that values of β near 1.5 are about optimum for this particular problem. One should notice that if one sets β equal to unity in (27), equation (18) is recovered. The somewhat peculiar ordering of the iteration number is due to the fact that it is convenient to allow the computer to relax each nodal point

in turn. This is in contrast to the usual manual methods where the point at which the residue is largest is relaxed first.

For good choices of β, it has been found that problems typically converge about twice as fast as the $\beta=1$ case.

Once the values of Y are calculated to the desired accuracy, the program continues on to compute the magnetic field components as given by equations (9) and (10). Several more sophisticated finite difference formulations are possible, but it has been found adequate to compute the derivatives involved by the simplest formulae:

$$\left(\frac{\partial Y}{\partial z}\right)_0 = \frac{Y_1 - Y_3}{2\Delta z} = \left(-r\,H_r\right)_0 \tag{29}$$

and

$$\left(\frac{\partial Y}{\partial r}\right)_0 = \frac{Y_4 - Y_2}{2\Delta r} = \left(r\,H_z\right)_0 \quad . \tag{30}$$

Some ambiguity is encountered along the axis where $r=0$ and values of Y_4 are not available. This has been remedied by the somewhat unsophisticated procedure of setting H_z along the axis equal to H_z along the first mesh row ($z = \Delta z$). The region along the axis is not of great concern in the design problem and it is found in practice that this procedure produces reasonable if not entirely correct results.

Once the field components have been calculated, the line integral (equation (13)) is evaluated. The simplest paths that can

be used are rectangular and follow the integration mesh. The computer program is written so that one need only specify the mesh nodal point number at which to start (on the lower left of the net) and the width of the path desired (in terms of mesh nodes). Since the integral is known to be independent of path the choice of the starting point and the width should not be critical. The usual procedure is to use several different paths and to average the results. In no case tried so far has a variation of greater than one percent been observed between various paths. The practice of averaging is seen to be of little value in this case but it has been retained in the program. From the line integral, the current, inductance and stored energy are easily calculated.

The only remaining quantity of interest is the field index along the median plane as defined in equation (16). Some anxiety was expressed about the calculation of this parameter. This arises principally from the fact that it involves the second derivative of the flux function Y which was itself calculated by a finite difference method. Small errors in Y might be greatly magnified in quantities that are calculated by finite difference approximations to derivatives.

So far this fear has been groundless and apparently satisfactory results have been obtained using only the simplest finite difference equations.

One additional feature of this computer program is that it will punch various parameters on cards so that they can be used in other calculations. At the present time Y is punched on cards if a control index NPCH is set equal to unity in the input data. These cards are then used in trajectory calculations.

REFERENCES

1. Forsythe, G.E. and Wasow, W.R., "Finite Difference Methods for Partial Differential Equations," Wiley, 1960.