Lawrence Berkeley National Laboratory

Lawrence Berkeley National Laboratory

Title

The "Inverse Problem" to the Evaluation of Magnetic Fields

Permalink

https://escholarship.org/uc/item/1523v5zf

Author

Caspi, S.

Publication Date

1992-12-04

eScholarship.org Powered by the California Digital Library
University of California

SC-MAG-412
LBL 33316

The "INVERSE PROBLEM" to the Evaluation of Magnetic Fields.*

S.Caspi, M.Helm, L.J.Laslett, and V.Brady

Lawrence Berkeley Laboratory

December 4, 1992

* This was supported by the Director, Office of Energy Research, Office of High Energy and Nuclear Physics, High Energy Physics Division, U. S. Department of Energy, under Contract No. DE-AC03–76SF00098.

Abstract

In the design of superconducting magnet elements, such as may be required to guide and focus ions in a particle accelerator, one frequently premises some particular current distribution and then proceeds to compute the consequent magnetic field through use of the laws of Biot and Savart or of Ampere. When working in this manner one of course may need to revise frequently the postulated current distribution before arriving at a resulting magnetic field of acceptable field quality. It therefore is of interest to consider an alternative ("inverse") procedure in which one specifies a desired character for the field required in the region interior to the winding and undertakes then to evaluate the current distribution on the specified winding surface that would provide this desired field.

We may note that in undertaking such an inverse procedure we would wish, on practical grounds, to avoid the use of any "double-layer" distributions of current on the winding surface or interface but would not demand that no fields be generated in the exterior region, so that in this respect the goal would differ in detail from that discussed in [1], in analogy to the distribution sought in electrostatics by the so-called Green's equivalent stratum.

Introduction

In a very simple initial example it was desired to find a distribution of surface current density, on the surface of a *circular cylinder* of radius "a", that would provide in the interior a periodic alternating purely sinusoidal quadrupole field whose scalar magnetic potential would be proportional to $I_2\left(\frac{\pi\rho}{L}\right)\cos\left(\frac{\pi z}{L}\right)\sin 2\phi$ (cylindrical coordinates) – or (more generally) to an expression of the form $\left[\sum_m C_m I_2\left((2m-1)\frac{\pi\rho}{L}\right)\cos\left((2m-1)\frac{\pi z}{L}\right)\right]\sin 2\phi$. In this instance, with a *circular cylinder* selected as the form on which the current windings are to be placed, it may be evident that an analytic solution can readily be obtained — — and that indeed if additional azimuthal harmonics characterized by factors $\sin 6\phi$ or etc. were also present in the desired potential an analytical expression for the required current density could still be provided through superposition.

When a more general form of interface is considered desirable, however — —$\rho = f(z)$, but still of circular cross-section — — the coordinate system for a conventional analytic solution for the required current distribution may be lacking and we may wish to turn to some sort of relaxation process or processes in ρ,z space for computational solutions. We turn now to consideration of this option.

Analysis

With a continuous interface ($\rho = f(z)$, and of circular cross-section, specified) for the surface on which current windings are to be placed the problem may then be specified as follows. [We may treat a single azimuthal component at a time in the course of the computational work, since the ϕ variation will be a separable variable and several harmonics, if present, may have their multipole fields superposed when required.]

The interface $\rho = f(z)$ will separate the ρ,z space into an interior region (Region I) and an exterior region (Region II). In each of these regions the magnetic field may be described by scalar potential functions $\Omega^I(\rho, z)\sin n\phi$ and $\Omega^{II}(\rho, z)\sin n\phi$ (where n represents the azimuthal harmonic number) that should satisfy the differential equation

$$\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial\Omega}{\partial\rho}\right) + \frac{\partial^2\Omega}{\partial z^2} - \frac{n^2}{\rho^2}\Omega = 0$$

[see [2],[3],[4]] or

$$\frac{\partial^2\Omega}{\partial\rho^2} + \frac{1}{\rho}\frac{\partial\Omega}{\partial\rho} + \frac{\partial^2\Omega}{\partial z^2} = \left(\frac{n}{\rho}\right)^2\Omega$$

The function Ω^I may be taken to be a *given function* that will vanish along the axis $\rho = 0$, while the function Ω^{II} (that remains to be found) should tend to zero at remote distances $(\rho \to \infty)$. Each of the functions Ω^I & Ω^{II} should fulfill suitable boundary conditions [e.g., Dirichlet or Neumann $\left(\frac{\partial\Omega}{\partial n}\right) = 0$] at the side boundaries of a relaxation mesh.

The functions Ω^I & Ω^{II} should *not* be expected themselves to be continuous across the interface but we require instead that the *normal derivatives* should be continuous — — i.e., at the interface $\frac{\partial\Omega^{II}}{\partial n}$ should become equal to the prescribed value of $\frac{\partial\Omega^I}{\partial n}$ at that same point (with n having the *same* direction in space on the two sides of the interface).

It appears that we have a mathematically well posed problem in ρ,z space for the function Ω^{II}. We thus may anticipate, in particular, that if the function Ω^I in some z region increases from zero to (say) some positive value as it approaches the interface, there then may be a jump to a negative value for Ω^{II} on the opposite side of the interface at that location and that Ω^{II} then will grow (to less negative values) as ρ increases further — — thus maintaining continuity at the interface of the normal derivatives of potential and permitting Ω^{II} to tend towards zero at large ρ.

If the problem thus posed becomes solved, as by a relaxation process applied to the function Ω^{II}, the values for the surface-current density on the interface then can be found. Thus, specifically, the value of the longitudinal component of current density (i.e., the component running along the interface in the ρ,z plane) is given by the discontinuity in the ϕ component of field [i.e, by the difference between $-\frac{1}{\mu_0}\frac{\partial\Omega^{II}}{\rho\partial\phi}\sin n\phi \quad and \quad +\frac{1}{\mu_0}\frac{\partial\Omega^I}{\rho\partial\phi}\sin n\phi$, or $\frac{n}{\mu_0}\frac{1}{\rho}\left(\Omega^I - \Omega^{II}\right)\cos n\phi$], while the discontinuity in the longitudinal derivative or longitudinal component of filed similarly gives the ϕ component of current density as $\frac{1}{\mu_0}\left(\frac{\partial\Omega^{II}}{\partial s} - \frac{\partial\Omega^I}{\partial s}\right)\sin n\phi$ — — so as together will describe a current with zero surface divergence (as desired).

We might expect that the relaxation solution of the problem posed for the function Ω^{II} might be achieved by a slightly modified version of the program POISSON. We may now first mention, however, that the boundary condition $\Omega^{II} \to 0$ as $(\rho \to \infty)$ may not be easily realized on a necessarily finite mesh, so that one may need to have a recourse to some approximate treatment of this matter (such as imposing a boundary value $\Omega^{II} = 0$ at the outer edge of a quite extended mesh, or by some more sophisticated special treatment). A possible difficulty with regard to employing an available relaxation program, to solve the problem posed above for determining the function $\Omega^{II}(\rho, z)$, will arise if the program can accommodate a Neumann type of boundary condition *only if* in such cases the value specified for the normal derivative is *zero*

(in contrast to the present requirement that the normal derivative of Ω^{II} shell be taken as equal to the known (prescribed) normal derivative of Ω^{I}).

In recognition of the possible occurrence of this difficulty, we now suggest a possible means of circumventing this difficulty, so that one could proceed by use of an available relaxation program for solving the relevant differential equation [subject to the provision the program "editor" will permit one to obtain correct values of *normal derivatives (right up to any boundary)* of solutions $\Omega^{II}(\rho, z)$ obtained by the relaxation process]. The suggested method may well be regarded as inefficient from the point-of-view of computer usage, but none-the-less its adoption may be regarded as appropriate method and we then may go on to illustrate the method by an extremely simple example that may serve to lend some confidence to the belief that the overall process will be convergent.

Implementation

The method to be outlined will omit the need to apply a Neumann boundary condition at the interface $\rho = f(z)$ in performing a relaxation sweep throughout the mesh wherein the function Ω^{II} is to be evaluated. The method instead will employ the assignment of "provisional" values for Ω^{II} at the node points along the interface, thus in effect introducing a Dirichlet type boundary condition at this boundary while relaxation processes are underway in Region II. Subsequent to the execution of several relaxation passes through Region II these provisional Dirichlet boundary values will be revised, in light of currently available provisional estimates of $\frac{\partial \Omega^{II}}{\partial n}$, in such a way that the desired continuity of $\frac{\partial \Omega}{\partial n}$ across the interface may become more closely attained and the relaxation process then will be resumed. Such readjustments can be performed repeatedly until a suitable close degree of convergence is attained and the then-available values of Ω^{II} and its derivative $\frac{\partial \Omega^{II}}{\partial s}$ employed (together with the corresponding known values of Ω^{I} *and* $\frac{\partial \Omega^{I}}{\partial s}$) to evaluate the implied values of current density on the interface surface.

Specifically we suggest that, in following this procedure, suitable *initial* provisional values for Ω^{II} at the interface may be taken simply as $-\Omega^{I}$ at the interface vertices. Moreover, when subsequently revising such provisional values of Ω^{II} at points on the interface, we propose that the values be scaled up simply by a provisional factor that is the average ratio of the known desired normal derivative of Ω^{I} to the normal derivative of the provisional present function Ω^{II} at that same point. ["over-relaxation factors" for these relaxation and revision process may well be acceptable, and even appropriate, but need not be regarded as necessary.]

Given an interface $\rho_w = f(z)$ for $0 \leq z \leq \frac{L}{4}$ $\left(or\ -\frac{L}{4} \leq z \leq \frac{L}{4}\right)$ with a local slope angle γ given by

$$\tan \gamma_i = T_i = \frac{d\rho_w}{dz}$$

for which

$$\sin \gamma_i = \frac{T_i}{\sqrt{1 + T_i^2}}$$

$$\cos \gamma_i = \frac{1}{\sqrt{1 + T_i^2}}$$

The magnetic scalar potential in the inner region $\rho < \rho_w$ (region I) is of the form

$$V^I = \sum_n \Omega_n^I(\rho, z) \sin n\phi \ , \ with \ \ \Omega_n^I(0, z) = 0$$

In the region $\rho > \rho_w$ (region II) we shall similarly expect to write

$$V^{II} = \sum_n \Omega_n^{II}(\rho, z) \sin n\phi \ , \ with \ \ \Omega_n^{II}(\infty, z) \to 0$$

For each of the functions Ω_n^I & Ω_n^{II} we expect them to satisfy the differential equation

$$\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial\Omega_n}{\partial\rho}\right) + \frac{\partial^2\Omega_n}{\partial z^2} - \frac{n^2}{\rho^2}\Omega_n = 0$$

and to satisfy the boundary conditions

$$\begin{cases} \frac{\partial\Omega_n}{\partial z}|_{z=0} = 0 \\ \Omega_n|_{z=\frac{L}{4}} = 0 \end{cases} \quad or \quad \begin{cases} \Omega_n|_{z=-\frac{L}{4}} = 0 \\ \Omega_n|_{z=\frac{L}{4}} = 0 \end{cases}$$

The requisite connection between the functions Ω_n^I & Ω_n^{II} occurs at $\rho_w = f(z)$ and explicitly is $\frac{\partial\Omega^I}{\partial n} = \frac{\partial\Omega^{II}}{\partial n}$ where n denotes distance in the *normal* direction (in the same sense), or

$$\cos\gamma_i\frac{\partial\Omega_n^{II}}{\partial\rho} - \sin\gamma_i\frac{\partial\Omega_n^{II}}{\partial z} = \cos\gamma_i\frac{\partial\Omega_n^I}{\partial\rho} - \sin\gamma_i\frac{\partial\Omega_n^I}{\partial z}$$

when written in cylindrical coordinates, for each and every location z_i along the interface $\rho_{w,i} = f(z_i)$. We have used the following relations to calculate derivatives of potentials normal and tangent to the interface :

$$\frac{\partial\Omega}{\partial n} = \frac{\partial\Omega}{\partial r}\cos\gamma - \frac{\partial\Omega}{\partial z}\sin\gamma$$
$$\frac{\partial\Omega}{\partial s} = \frac{\partial\Omega}{\partial r}\sin\gamma + \frac{\partial\Omega}{\partial z}\cos\gamma$$

Modifying "POISSON"

The program POISSON was modified to solve the revised differential equation, and to make the necessary provisional revisions on the interface. With the aide of the mesh generator program AUTOMESH we generate two regions (I & II), solve the inner region I first, and obtain the normal derivatives on each interface point. Second we turn off region I and turn on region II assuming the initial potentials on the interface are the negative of those in region I. Following several relaxation cycles (10 to 50), the potentials on the interface are multiplied by a common *Factor* (see below) and the updated potetials held constant during the next iteration cycle. The update *Factor* is the average ratio of the normal derivative on both sides of the interface.

$$Factor = \frac{1}{N}\sum_{i=1}^{N}\frac{\cos\gamma_i\frac{\partial\Omega_n^I}{\partial\rho} - \sin\gamma_i\frac{\partial\Omega_n^I}{\partial z}}{\cos\gamma_i\frac{\partial\Omega_n^{II}}{\partial\rho} - \sin\gamma_i\frac{\partial\Omega_n^{II}}{\partial z}}$$

N is the total number of points on the interface. Accordingly the interface potentials are revised :

$$\Omega^{II}_{i,new} = \Omega^{II}_{i,old}[1 + \lambda(Factor - 1)]$$

with λ being the relaxation factor. As the process converges the value of Factor tends towards 1. and $\Omega^{II}_{i,new} = \Omega^{II}_{i,old}$. The resulting current density on the interface can now be obtained using the potentials and derivatives on both sides of the interface.

$$J_s = \frac{n}{\mu_0}\frac{\Omega^I_n - \Omega^{II}_n}{\rho_w}\cos n\phi$$
$$J_\phi = -\frac{1}{\mu_0}\left(\frac{\partial\Omega^I_n}{\partial s} - \frac{\partial\Omega^{II}_n}{\partial s}\right)\sin n\phi$$

(with $\mu_0 = \frac{4\pi}{10}$ in "Poisson units" of cm , amp, gauss), which should prove to be such that the surface-divergence of this surface-current density vanishes. Lines of current flow (or wire direction) is given by the differential equation

$$\begin{aligned}\frac{d\phi}{ds} &= \frac{1}{\rho_w}\frac{J_\phi}{J_s} \\ &= -\frac{1}{\rho_w n}\left(\frac{\frac{\partial\Omega^I_n}{\partial s} - \frac{\partial\Omega^{II}_n}{\partial s}}{\Omega^I_n - \Omega^{II}_n}\right)\tan n\phi \\ &= g(r,z)\tan n\phi\end{aligned}$$

where s is a location distance along the interface curve $\rho_w = f(z)$. The above differential equation can be rewritten as a function of z and r (instead of s and r)

$$\frac{d\phi}{dz} - g(\rho,z)\sqrt{1 + \left(\frac{d\rho}{dz}\right)^2_w}\tan n\phi = 0$$

POISSON output provides tables of ρ, z, and g(ρ, z), which are used to calculate wire locations.

An alternating cylindrical quadrupole — EXAMPLE

To illustrate the procedure just outlined and possibly to give a sense of any issues concerning convergence, we apply this method to a simple problem in which a purely sinusoidal AG quadrupole field is to be formed by current windings placed on a circular cylinder. It will be recalled that for such a simple configuration it was known that for a current distribution at ρ=a

$$J = \cos\frac{\pi z}{L}\cos 2\phi\, e_z + \frac{\pi a}{2L}\sin\frac{\pi z}{L}\sin 2\phi\, e_\phi$$

it was determoned by analytical means that the fields were derived from scalar potential functions

$$V^I = \Omega^I \sin 2\phi = -\frac{2\pi}{10}\frac{\pi a^2}{L}K_2'\left(\frac{\pi a}{L}\right)I_2\left(\frac{\pi\rho}{L}\right)\cos\left(\frac{\pi z}{L}\right)\sin 2\phi \quad for \quad \rho < a$$
$$V^{II} = \Omega^{II} \sin 2\phi = -\frac{2\pi}{10}\frac{\pi a^2}{L}I_2'\left(\frac{\pi a}{L}\right)K_2\left(\frac{\pi\rho}{L}\right)\cos\left(\frac{\pi z}{L}\right)\sin 2\phi \quad for \quad \rho > a$$

In the program we accordingly employ a mesh with the type of boundary condition indicated (see AUTOMESH input file in Appendix) and from the formula shown for Ω^I the values of Ω^I are readily specified as fixed numerical values along the interface $\rho = a$. With this potential values, an input file into POISSON (see Appendix) can be generated. Results of the potentials and derivatives are given in the Appendix and are also plotted in Fig. 1. The flux plot in regions I and II are shown in Figures 2 and 3. Figures 4 and 5 are the current density calculated by POISSON, and Fig. 6 contains the g(s) function from which wire location are calculated (Fig. 7 and 8), (see Appendix for programs).

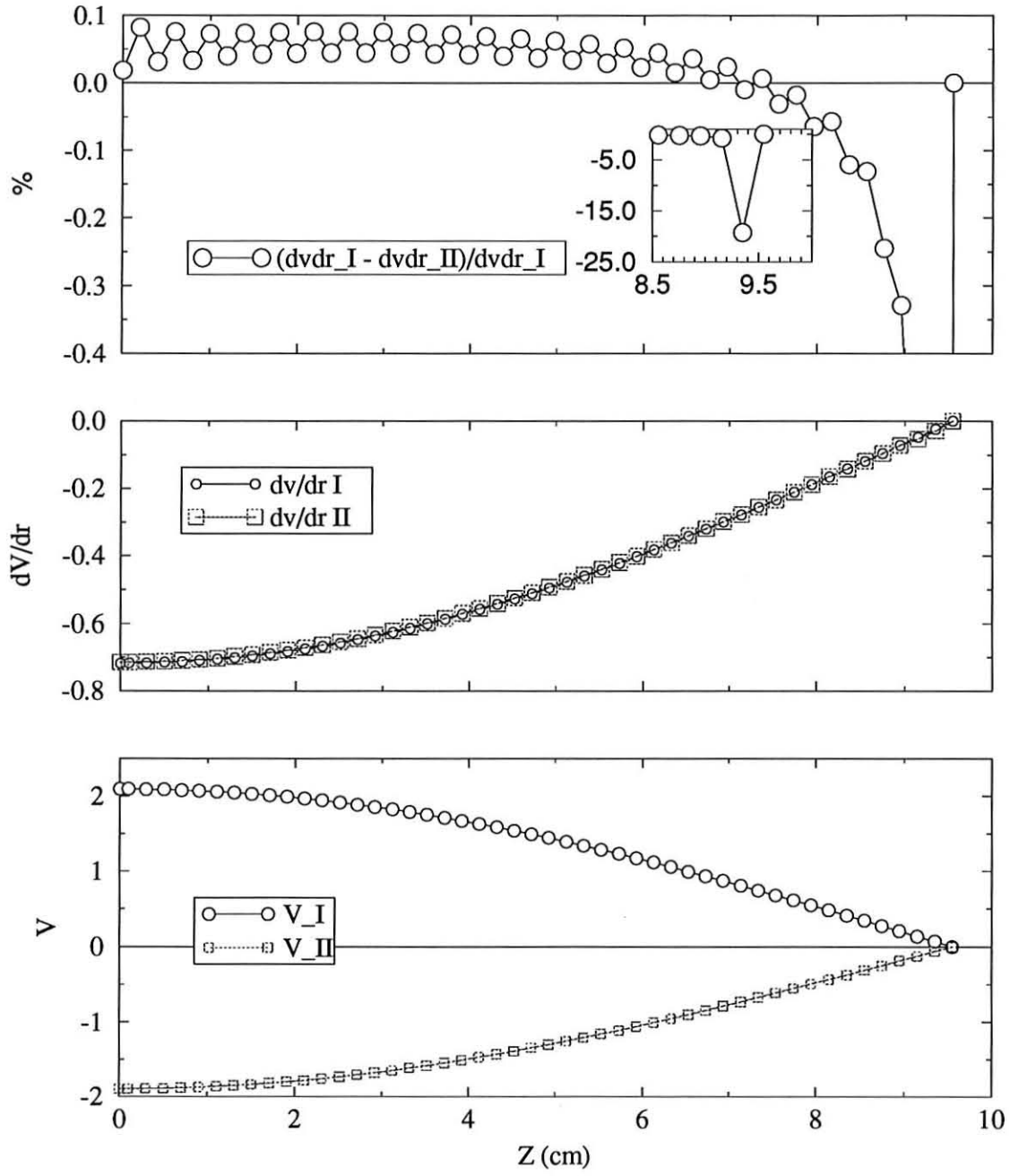

Figure 1 Potentials and normal derivatives on the interface.

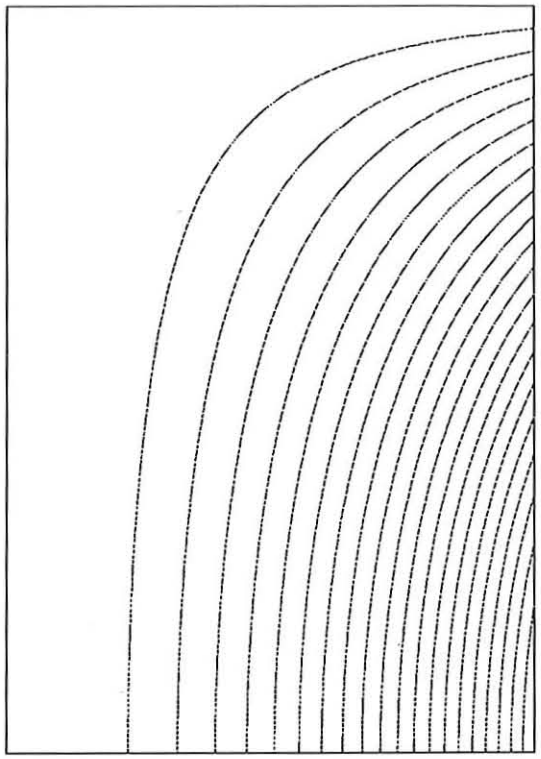

Figure 2 Flux plot in inner Region I.

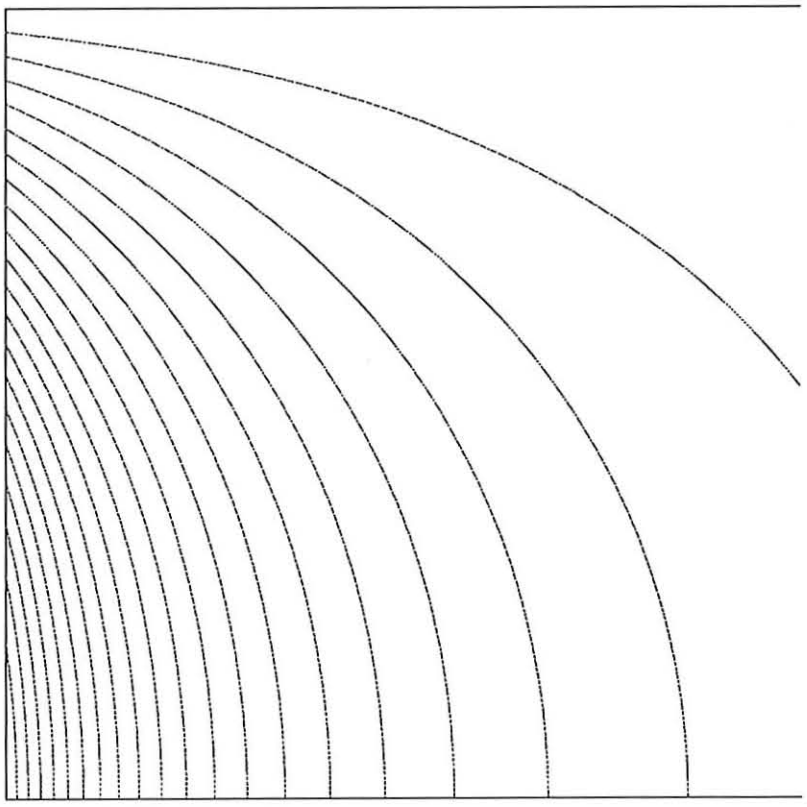

Figure 3 Flux plot in outer Region II.

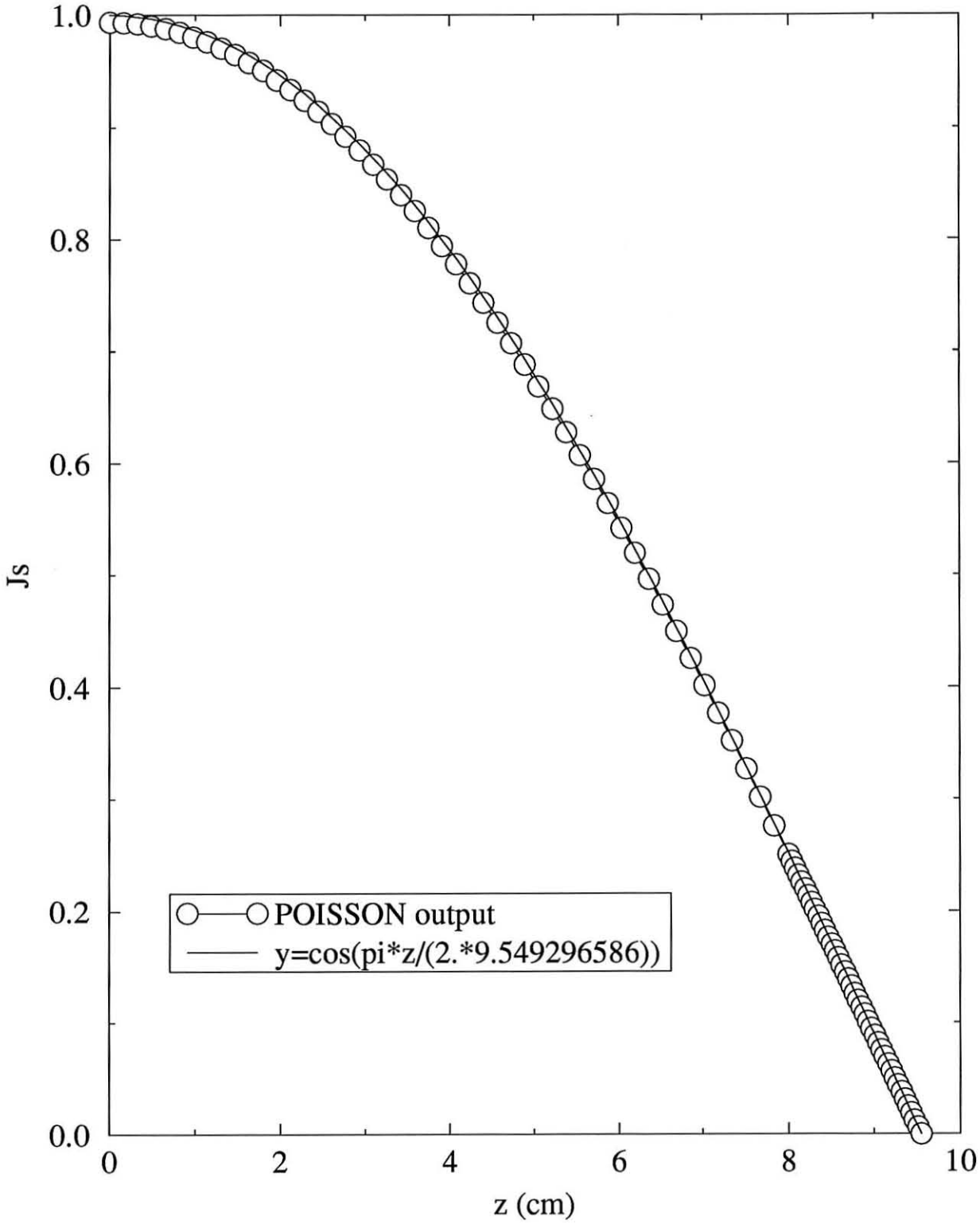

Figure 4 Current density in the Z direction.

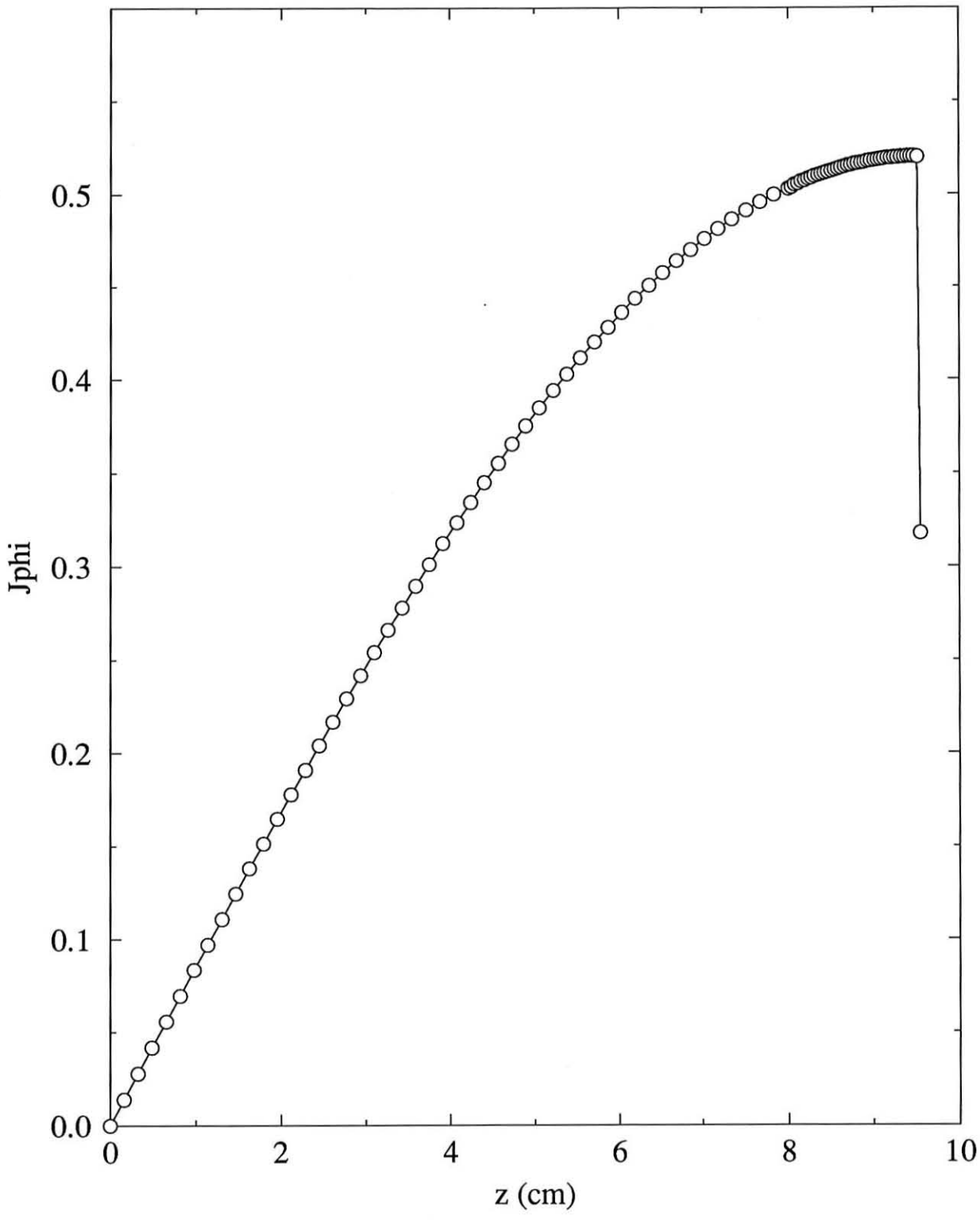

Figure 5 Current density in the ϕ direction.

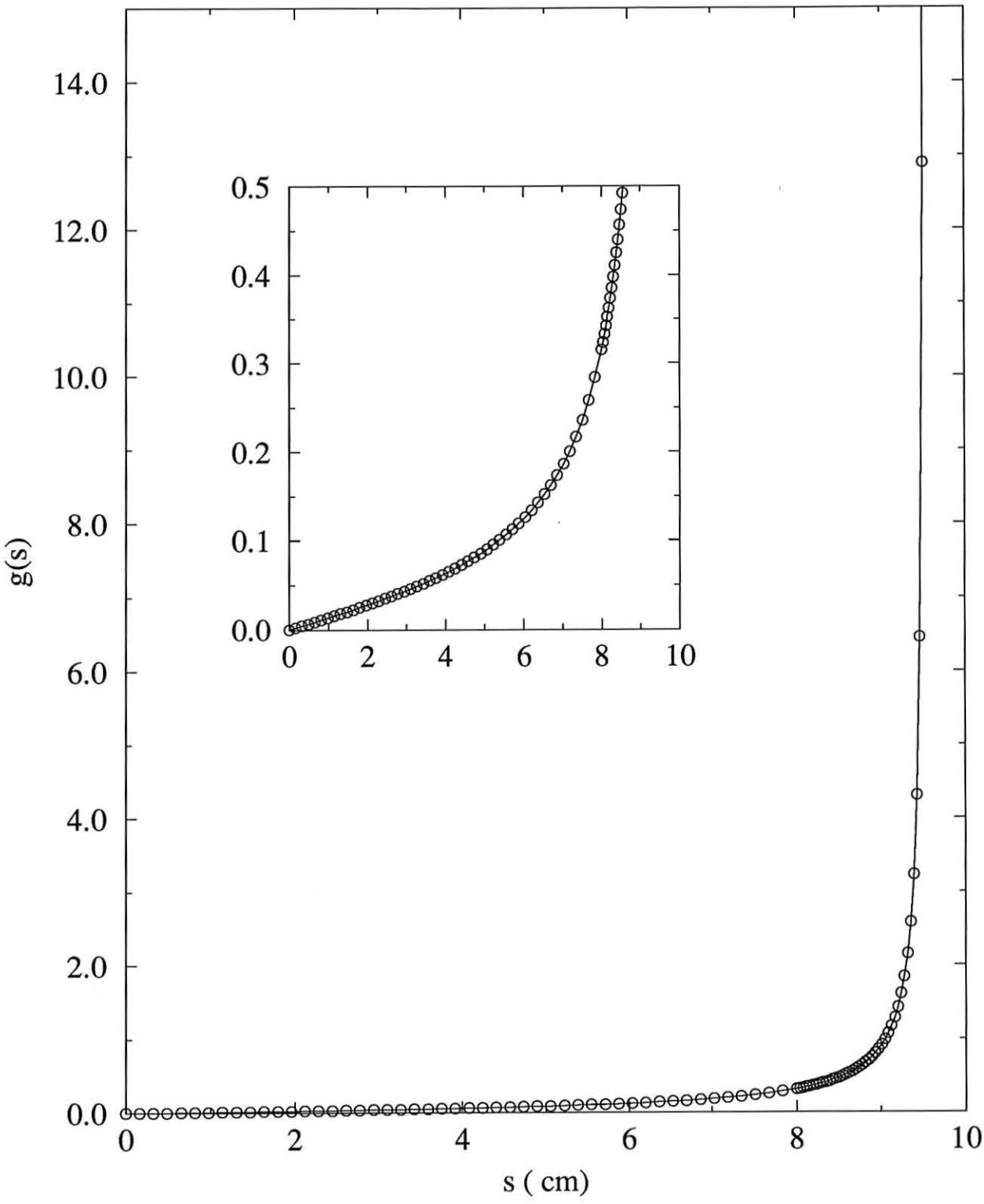

Figure 6 The function g(s).

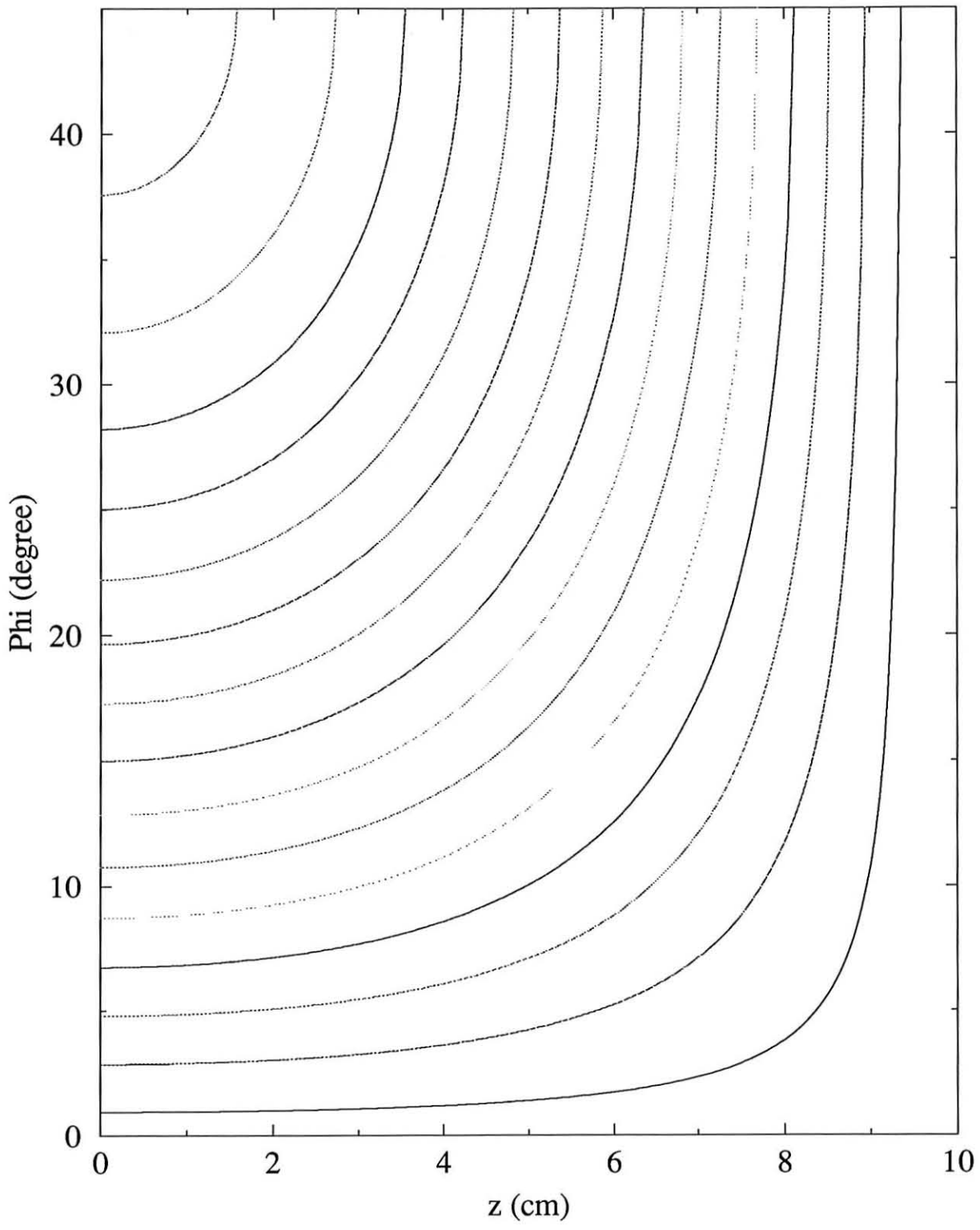

Figure 7 The azimuthal angle ϕ of each wire on the cylindrical surface.

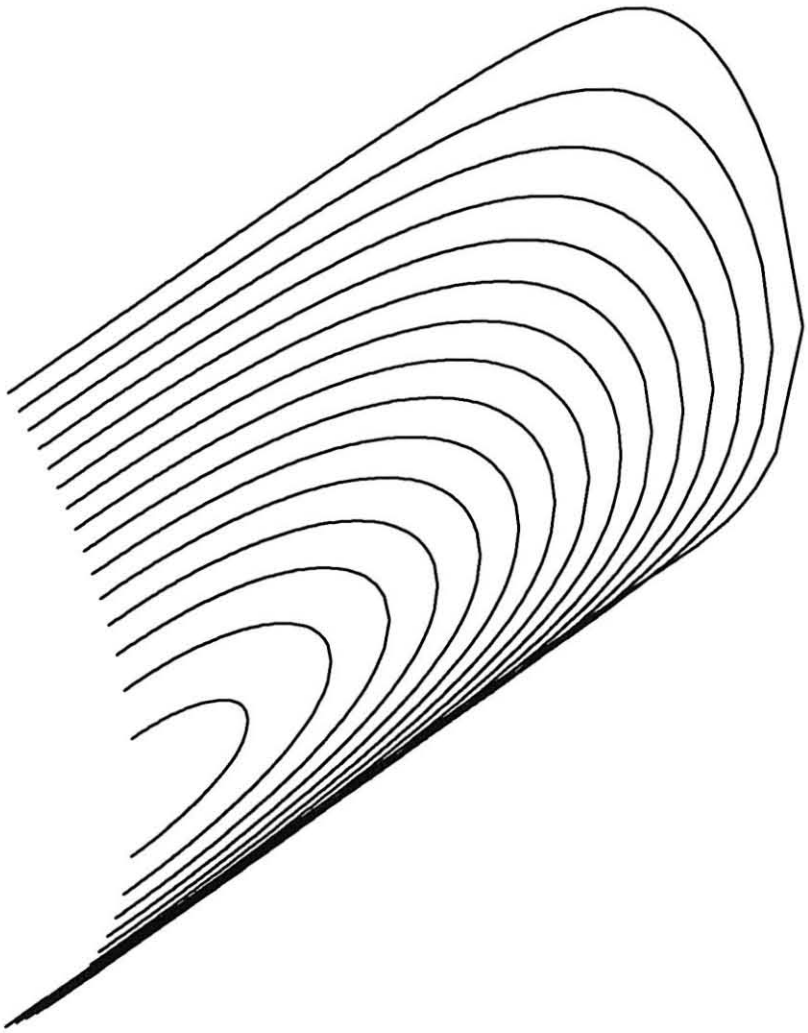

Figure 8 A 3D CAD drawing of the quad wire distribution.

Bibliography

[1] *J. Appl, Physics 37 (No. 6), 2361-2363*, May 1966.

[2] V.Brady. Three-dimensional field components. *Lawrence Berkeley Laboratory, HIFAR Note-261*, January 1990.

[3] L.J.Laslett S.Caspi M.Helm and V.Brady. Three dimensional magnetic field produced by an axisymmetric iron yoke. *Lawrence Berkeley Laboratory, LBL-29810, SC-MAG-315*, June 1991.

[4] S.Caspi M.Helm and L.J.Laslett. The use of a relaxation method to calculate the 3d magnetic field contribution of an iron yoke. *Lawrence Berkeley Laboratory, LBL-29826, SC-MAG-317*, July 1991.

Appendix A

Interpolation & Differentiation in POISSON

In the reprograming of POISSON the interpolation and interpolation-differetiation are based on polynomials of the type shown in the Table on Page 4 of Ref. [2]. These functions are designed to be individually solutions of the differential equation (Page 2) $\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial\Omega}{\partial\rho}\right)+\frac{\partial^2\Omega}{\partial z^2}-\frac{n^2}{\rho^2}\Omega=0$ for which (for specified n) solutions have been sought by means of relaxation.

The functions of that Table are each of such a form that (for n>0) they *vanish* at r=0 and so (as intended) appear appropriate for use in an interior region that includes the *axis* of the cylindrical coordinate system. [The coordinate Z in these functions could be replaced, if desired, by $z\text{-}z_0$ due to the translational character of the coordinate Z, but such a replacement would appear to have merit only in regard to arithmatic accuracy as affected by register-length limitations. Such a shift of origin may be understood to be applied in cartesian coordinate use of the unmodefied POISSON (see Ref. [1], esp. footnote to Table B.13.2.I, page 28) but it would not be appropriate for the cylindrical coordinate radial coordinate r.] For puposes of interpolation and/or interpolation-differentiation such function are to be fit (by "least square") to values of Ω at neighboring vertices of the mesh.

In application to the modefied POISSON, the polynomials of the Table in Ref. [2], and Ref. [1], (Table B.13.2.II page 35) for the scalar potential, do *NOT* appear particularly suitable however for fitting values of the function Ω in *EXTERIOR* regions that extend essentially to ∞ where Ω may be expected to become zero. [It may be noted that the polynomials of this table do, in fact, continue formally to satisfy the differential equation if the index n is replaced by —n, but such a replacement (i) results in some terms still growing in magnitude without limit as r becomes large and (ii) revised-denominator factors (such as 2–n for n=2 or etc.), unless removed by multiplication, may lead to computational blow-up if n is assigned some integer value.]

In review of the circumstances noted above, the evaluation (or estimation) of interpolated Ω values and derivatives *in an EXTERNAL region* may require a basic replacement of functions so as to be expressed in terms of reference radius r_0 as (possibly truncated) power series developed in powers of $\delta = r - r_0$ and z (with coefficients such that these series expressions constitute at least approximate (truncated) solutions of the differential equation with which we are concerned.

If one enters n < 0, use Such expressions should individually satisfy

K	$\Omega_k^n(r,z)$
1	r^n
2	$r^n z$
3	$(n+1)r^n z^2 - \frac{1}{2}r^{n+2}$
4	$(n+1)r^n z^3 - \frac{3}{2}r^{n+2}z$
5	$(n+1)(n+2)r^n z^4 - 3(n+2)r^{n+2}z^2 + \frac{3}{4}r^{n+4}$
6	$(n+1)(n+2)r^n z^5 - 5(n+2)r^{n+2}z^3 + \frac{15}{4}r^{n+4}z$
7	$(n+1)(n+2)(n+3)r^n z^6 - \frac{15}{2}(n+2)(n+3)r^{n+2}z^4 + \frac{45}{4}(n+3)r^{n+4}z^2 - \frac{15}{8}r^{n+6}$

$\frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial\Omega}{\partial\rho}\right) + \frac{\partial^2\Omega}{\partial z^2} - \frac{n^2}{\rho^2}\Omega = 0$ and even with n a negative integer should not blow up.

AUTOMESH

To approach a solution to such a problem by an iterative procedure, we have modified the program POISSON and adopted the following procedure :

We made use of the program AUTOMESH to generate two regions (inner region I and outer region II). The inner region is assigned material 1 and the outer region material 1025 (which is equivalent to material 0 in other cases), also we assigned each region a current density of —0.7957747 (or $\frac{10}{4\pi}$) to take care of the modified differential equation (see input file below).

```
Quad cylinder
$geo NREG=2 XMIN=0.0 XMAX=50.0 YMIN=0.0 YMAX=9.549296586
dx=0.2 dy=0.2 $
$reg mat = 1 den=-0.7957747 npoint=7  $
$PO  x=0.0 y=0.0 $
$PO  x=6.366197724 y=0.0 $
$PO  x=50.0 y=0.0 $
$PO  x=50.0 y=9.549296586 $
$PO  x=6.366197724 y=9.549296586 $
$po  x=0.0 y=9.549296586 $
$PO  x=0.0 y=0.0 $
$reg mat = 1025 den=-0.7957747 npoint=5  $
$PO  x=6.366197724 y=0.0 $
$PO  x=50.0 y=0.0 $
$PO  x=50.0 y=9.549296586 $
$PO  x=6.366197724 y=9.549296586 $
$PO  x=6.366197724 y=0.0 $
```

Next we have used the output file from AUTOMESH (which is also the input file to LATTICE) to obtain the interface points. The physical coordinates of the interface points obtained by the program "lat2newpoi" are used in the sample program "potfile" to calculate potentials on

the interface.

```
lat2newpoi < in.latt > pot.rz
potfile < pot.rz > pot.pot
poisson < in.pois > out.pois
```

The file pot.pot contains the logical mesh point on the interface and their region I potential values.

POISSON input

```
0 DUMP
*66 0.0
*19 1
*42 33 33 1 49
*131 2.000000
*136 1
*137 0
*74 1.95
*75 1.96
*20   49 s
        33          1     2.096646424839027
        33          2     2.095523854076131
        33          3     2.092157343864345
        33          4     2.086550499147154
        33          5     2.078709323874369
        33          6     2.068642215134743
        33          7     2.056359950010085
        33          8     2.041875683245411
        33          9     2.025204933324081
        33         10     2.006365534948627
        33         11     1.985377669654441
        33         12     1.962263811778074
        33         13     1.937048725420820
        33         14     1.909759386187659
        33         15     1.880425028341242
        33         16     1.849077063853275
        33         17     1.815749060916343
        33         18     1.780476726210482
        33         19     1.743297794786612
        33         20     1.704252096028298
        33         21     1.663381441056512
        33         22     1.620729595201462
        33         23     1.576342253877383
        33         24     1.530266903552044
        33         25     1.482552904754207
        33         26     1.433251350839591
        33         27     1.382415035158054
        33         28     1.330098421181071
        33         29     1.276357478271286
        33         30     1.221249779518272
        33         31     1.164834335638509
        33         32     1.107171587002677
        33         33     1.048323222375472
        33         34    0.9883522867490240
        33         35    0.9273229985167589
        33         36    0.8653007093828187
        33         37    0.8023518662449801
        33         38    0.7385438130378309
        33         39    0.6739449084515552
        33         40    0.6086243266256957
        33         41    0.5426520144900224
        33         42    0.4760986504511001
        33         43    0.4090354346322512
        33         44    0.3415342133330507
        33         45    0.2736672685671230
        33         46    0.2055073082964059
        33         47    0.1371272514130845
```

```
        33         48     0.6860035520353576E-01
        33         49     0.1283783266788422E-15
1 DUMP
*66 0.0
*19 1
*42 33 33 1 49
*131 2.000000
*136 1
*137 1
*138 1.0
*139 10
*20   0 s

-1 DUMP
```

POISSON output

```
Region I
2.0000000
L       V(SCALAR)          R                Z              BR (G)              BPhi (G)            BZ (G)
 1 2.096646547318E+00 6.366197586060 0.000000000000 -0.7171517941419098 -0.6586809532612223 0.0000000000000000
 2 2.095523834229E+00 6.366197586060 0.198943659663 -0.7170968789564882 -0.6583282425343521 0.0112921689116404
 3 2.092157363892E+00 6.366197586060 0.397887378931 -0.7157290302088136 -0.6572706346635924 0.0225639262943619
 4 2.086550474167E+00 6.366197586060 0.596831083298 -0.7139919359804071 -0.6555091782686451 0.0338649370438848
 5 2.078709363937E+00 6.366197586060 0.795774638653 -0.7111499741188435 -0.6530458207860993 0.0450349958988621
 6 2.068642139435E+00 6.366197586060 0.994718432426 -0.7078719190586976 -0.6498831088638026 0.0562053627686040
 7 2.056360006332E+00 6.366197586060 1.193662166595 -0.7035546417593521 -0.6460245628678970 0.0672939407482033
 8 2.041875600815E+00 6.366197586060 1.392605781555 -0.6987310317100115 -0.6414741525729682 0.0783175828993172
 9 2.025204896927E+00 6.366197586060 1.591549396515 -0.6929188132511985 -0.6362368963733038 0.0892700028554143
10 2.006365537643E+00 6.366197586060 1.790493011475 -0.6865999563440726 -0.6303183369730424 0.1001082473212732
11 1.985377550125E+00 6.366197586060 1.989436864853 -0.6793046958773339 -0.6237247660903953 0.1108650447229844
12 1.962263822556E+00 6.366197586060 2.188380479813 -0.6715229043924935 -0.6164633742604611 0.1214788167321808
13 1.937048792839E+00 6.366197586060 2.387324094772 -0.6627769629408211 -0.6085418388774856 0.1319857602950749
14 1.909759402275E+00 6.366197586060 2.586267709732 -0.6535636450887881 -0.5999686238004914 0.1423334007179633
15 1.880424976349E+00 6.366197586060 2.785211563110 -0.6434074112654756 -0.5907529419025737 0.1525417813046619
16 1.849076986313E+00 6.366197586060 2.984155178070 -0.6328011707059862 -0.5809046801694929 0.1625771114457657
17 1.815749049187E+00 6.366197586060 3.183098793030 -0.6212812494892564 -0.5704343996996754 0.1724443534372167
18 1.780476689339E+00 6.366197586060 3.382042407990 -0.6093269195694623 -0.5593532608028053 0.1821241151014716
19 1.743297815323E+00 6.366197586060 3.580986022949 -0.5964934146552936 -0.5476731728026399 0.1916088353647939
20 1.704252123833E+00 6.366197586060 3.779929876328 -0.5832423348210232 -0.5354066067834909 0.2008902648914298
21 1.663381576538E+00 6.366197586060 3.978873729706 -0.5691509624074282 -0.5225667453930392 0.2099532165223456
22 1.620729684830E+00 6.366197586060 4.177817344666 -0.5546593190010648 -0.5091672581381128 0.2187968367571791
23 1.576342225075E+00 6.366197586060 4.376760959625 -0.5396962792555930 -0.4952225260889092 0.2273991035152649
24 1.530266880989E+00 6.366197586060 4.575704574585 -0.5236999750140144 -0.4807475295268838 0.2357650917529443
25 1.482552886009E+00 6.366197586060 4.774648189545 -0.5072785382208010 -0.4657577355926394 0.2438702323923388
26 1.433251380920E+00 6.366197586060 4.973592281342 -0.4904989835104039 -0.4502692106380372 0.2517234681177605
27 1.382415056229E+00 6.366197586060 5.172535896301 -0.4730150559145631 -0.4342985078740854 0.2592982031869214
28 1.330098390579E+00 6.366197586060 5.371479511261 -0.4551964824971802 -0.4178627422723469 0.2666046438925219
29 1.276357412338E+00 6.366197586060 5.570423126221 -0.4367251283476145 -0.4009795156635321 0.2736153024330798
30 1.221249818802E+00 6.366197586060 5.769366741180 -0.4179451590261981 -0.3836669541882021 0.2803435061846677
31 1.164834260941E+00 6.366197586060 5.968310356140 -0.3985649252552599 -0.3659434835925471 0.2867611308850846
32 1.107171535492E+00 6.366197586060 6.167253971100 -0.3789033704589824 -0.3478282037354231 0.2928815947664912
33 1.048323273659E+00 6.366197586060 6.366197586060 -0.3586980356004351 -0.3293404766306112 0.2986782300536143
34 9.883522987366E-01 6.366197586060 6.565141677856 -0.3382400406246867 -0.3105000387989290 0.3041657086844798
35 9.273231029510E-01 6.366197586060 6.764085292816 -0.3172949803760822 -0.2913271510710452 0.3093174710446227
36 8.653007149696E-01 6.366197586060 6.963028907776 -0.2961271637775334 -0.2718422428057948 0.3141481001722067
37 8.023518323898E-01 6.366197586060 7.161972522736 -0.2745320244970684 -0.2520662676718636 0.3186317238283186
38 7.385438084602E-01 6.366197586060 7.360916137695 -0.2527460666529304 -0.2320203853168075 0.3227843928878915
39 6.739448904991E-01 6.366197586060 7.559859752655 -0.2305932667831558 -0.2117260362684598 0.3265814641393620
40 6.086243391037E-01 6.366197586060 7.758803367615 -0.2082821101535752 -0.1912049793856347 0.3300383905600971
41 5.426520109177E-01 6.366197586060 7.957747459412 -0.1856662989490925 -0.1704791607806644 0.3331327486168544
42 4.760986864567E-01 6.366197586060 8.156690597534 -0.1629253294224231 -0.1495708168088346 0.3358799409084603
43 4.090354442596E-01 6.366197586060 8.355634689331 -0.1399414223092047 -0.1285022774521897 0.3382580201254559
44 3.415341973305E-01 6.366197586060 8.554577827454 -0.1168664746977446 -0.1072961348477000 0.3402837926186755
45 2.736672759056E-01 6.366197586060 8.753521919250 -0.0936080876984228 -0.0859751122097983 0.3419349715675055
46 2.055073082447E-01 6.366197586060 8.952465057373 -0.0702918356059436 -0.0645620263796764 0.3432367368863155
47 1.371272355318E-01 6.366197586060 9.151409149170 -0.0468365076718379 -0.0430797925066235 0.3441459267248196
48 6.860034912825E-02 6.366197586060 9.350353240967 -0.0233634333967867 -0.0215514357513705 0.3446684048762722
49 1.283783384240E-16 6.366197586060 9.549296379089 -0.0376122077137689  0.0000000000000000 0.3285225789159964
```

Region II
-2.0000000

L	V(SCALAR)	R	Z	BR (G)	BPhi (G)	BZ (G)
1	-1.896162465414E+00	6.366197586060	0.000000000000	-0.7146012835643133	-0.5956970200125967	0.0000000000000000
2	-1.895147107617E+00	6.366197586060	0.198943659663	-0.7140958474209866	-0.5953780359462094	-0.0102063034109271
3	-1.892102543571E+00	6.366197586060	0.397887378931	-0.7130935225639747	-0.5944215579217058	-0.0204255089910009
4	-1.887031791967E+00	6.366197586060	0.596831083298	-0.7110482375598342	-0.5928285342884022	-0.0305858280305206
5	-1.879940458941E+00	6.366197586060	0.795774638653	-0.7085177983231684	-0.5906007262664564	-0.0407214919996799
6	-1.870835875597E+00	6.366197586060	0.994718432426	-0.7049744165876993	-0.5877404369899222	-0.0508131574319147
7	-1.859728176108E+00	6.366197586060	1.193662166595	-0.7009076122579773	-0.5842508502029318	-0.0608558766726866
8	-1.846628788368E+00	6.366197586060	1.392605781555	-0.6958660912188992	-0.5801355560850393	-0.0708291261575253
9	-1.831552158964E+00	6.366197586060	1.591549396515	-0.6902930529926331	-0.5753990931651146	-0.0807326653348484
10	-1.814514243827E+00	6.366197586060	1.790493011475	-0.6837740553590304	-0.5700464741466850	-0.0905428589514711
11	-1.795533155094E+00	6.366197586060	1.989436864853	-0.6767214474003987	-0.5640833891256475	-0.1002656511920216
12	-1.774629592350E+00	6.366197586060	2.188380479813	-0.6687532952571337	-0.5575163410687407	-0.1098690907203142
13	-1.751825656716E+00	6.366197586060	2.387324094772	-0.6602528472850007	-0.5503522732477428	-0.1193706076796686
14	-1.727145713329E+00	6.366197586060	2.586267709732	-0.6508689689934296	-0.5425988401964197	-0.1287259289011970
15	-1.700616283532E+00	6.366197586060	2.785211563110	-0.6409565315490390	-0.5342643738409057	-0.1379661042757094
16	-1.672265829256E+00	6.366197586060	2.984155178070	-0.6301977768741277	-0.5253578157604664	-0.1470311089084002
17	-1.642124753017E+00	6.366197586060	3.183098793030	-0.6189170205827311	-0.5158887171874991	-0.1559715988512880
18	-1.610225182298E+00	6.366197586060	3.382042407990	-0.6068290153157802	-0.5058671712682972	-0.1647069409321971
19	-1.576601400785E+00	6.366197586060	3.580986022949	-0.5942283445439317	-0.4953039485415211	-0.1733106659603959
20	-1.541289309325E+00	6.366197586060	3.779929876328	-0.5808628636824383	-0.4842103275901088	-0.1816774638227480
21	-1.504326857155E+00	6.366197586060	3.978873729706	-0.5669972150340902	-0.4725982305197484	-0.1899088503741216
22	-1.465753395052E+00	6.366197586060	4.177817344666	-0.5524101698341833	-0.4604800197411670	-0.1978710185750161
23	-1.425610322187E+00	6.366197586060	4.376760959625	-0.5373385695649323	-0.4478687011878426	-0.2056954941074185
24	-1.383940762695E+00	6.366197586060	4.575704574585	-0.5215923834572224	-0.4347778227071476	-0.2132165937524096
25	-1.340789242249E+00	6.366197586060	4.774648189545	-0.5053814307334030	-0.4212213724514938	-0.2206012703139086
26	-1.296202011484E+00	6.366197586060	4.973592281342	-0.4885430740816289	-0.4072138804871886	-0.2276492887565618
27	-1.250226722571E+00	6.366197586060	5.172535896301	-0.4712623410595745	-0.3927703171855791	-0.2345641981889984
28	-1.202912644838E+00	6.366197586060	5.371479511261	-0.4534022550367152	-0.3779061609622881	-0.2411079373689675
29	-1.154310449143E+00	6.366197586060	5.570423126221	-0.4351266575571650	-0.3626373305379781	-0.2475230705051209
30	-1.104472315693E+00	6.366197586060	5.769366741180	-0.4163210420676947	-0.3469802188079706	-0.2535338678370951
31	-1.053451287175E+00	6.366197586060	5.968310356140	-0.3971295677775074	-0.3309514896245329	-0.2594230521157143
32	-1.001302346864E+00	6.366197586060	6.167253971100	-0.3774579578616888	-0.3145684164930660	-0.2648740946109969
33	-9.480812327068E-01	6.366197586060	6.366197586060	-0.3574344094955742	-0.2978485100062984	-0.2702119684549830
34	-8.938447607525E-01	6.366197586060	6.565141677856	-0.3369803325239979	-0.2808096194531436	-0.2750804830408740
35	-8.386512563962E-01	6.366197586060	6.764085292816	-0.3162106957334357	-0.2634700682971730	-0.2798453250386410
36	-7.825595301793E-01	6.366197586060	6.963028907776	-0.2950600230056971	-0.2458483324152408	-0.2841097434626600
37	-7.256299019879E-01	6.366197586060	7.161972522736	-0.2736340587822778	-0.2279633618588596	-0.2882804738310051
38	-6.679232846650E-01	6.366197586060	7.360916137695	-0.2518771910868947	-0.2098342929624431	-0.2919219005700486
39	-6.095013996311E-01	6.366197586060	7.559859752655	-0.2298877684403288	-0.1914805160825434	-0.2954815027841535
40	-5.504268846945E-01	6.366197586060	7.758803367615	-0.2076170537720946	-0.1729217094674690	-0.2984841583792736
41	-4.907629167156E-01	6.366197586060	7.957747459412	-0.1851591369895773	-0.1541777207136207	-0.3014175651365134
42	-4.305735080845E-01	6.366197586060	8.156690597534	-0.1624689709273423	-0.1352686599069422	-0.3037689211543183
43	-3.699229407175E-01	6.366197586060	8.355634689331	-0.1396387546821128	-0.1162147218074232	-0.3060637557594167
44	-3.088762512030E-01	6.366197586060	8.554577827454	-0.1166256522587277	-0.0970363382623826	-0.3077529489447333
45	-2.474988534658E-01	6.366197586060	8.753521919250	-0.0935212222187174	-0.0777540596628036	-0.3093996444012933
46	-1.858564309565E-01	6.366197586060	8.952465057373	-0.0702857894624104	-0.0583885210737155	-0.3104188775036816
47	-1.240149501279E-01	6.366197586060	9.151409149170	-0.0470490036441238	-0.0389604464678961	-0.3113352044928466
48	-6.204069412542E-02	6.366197586060	9.350353240967	-0.0277713247453333	-0.0194906593101272	-0.3154642072825711
49	0.000000000000E+00	6.366197586060	9.549296379089	-0.0896259410725643	0.0000000000000000	-0.2558594443217982

l	r	z	Js0	Jphi0	dphi/ds = g(r,z)
1	6.366197586059570	0.	0.9982022747606084	0.	0.
2	6.366197586059570	0.198943659663200	0.9976677570276283	0.1710794069530458E-01	0.2693591207576026E-02
3	6.366197586059570	0.397887378931045	0.9960649983974142	0.3420990563197321E-01	0.5394908500426605E-02
4	6.366197586059570	0.596831083297729	0.9933955880074181	0.5128828923823057E-01	0.8109906975339016E-02
5	6.366197586059570	0.795774638652801	0.9896624771129083	0.6824284475626632E-01	0.1083153243347269E-01
6	6.366197586059570	0.994718432426452	0.9848695250476965	0.8516263246146202E-01	0.1358282998307874E-01
7	6.366197586059570	1.193662166595459	0.9790220667732289	0.1019783844942926	0.1636196779167577E-01
8	6.366197586059570	1.392605781555176	0.9721261183098603	0.1186871799614262	0.1917790159524392E-01
9	6.366197586059570	1.591549396514893	0.9641892848153965	0.1352838249064581	0.2203958736921009E-01
10	6.366197586059570	1.790493011474609	0.9552199660163696	0.1517153298462276	0.2494858739043656E-01
11	6.366197586059570	1.989436864852905	0.9452276967374923	0.1680124694665253	0.2792061389119346E-01
12	6.366197586059570	2.188380479812622	0.9342233739212931	0.1841008152251546	0.3095457667185363E-01
13	6.366197586059570	2.387324094772339	0.9222186323241167	0.2000230422040291	0.3406952058304070E-01
14	6.366197586059570	2.586267709732056	0.9092262985553978	0.2157021609003238	0.3726510879486728E-01
15	6.366197586059570	2.785211563110352	0.8952603343227515	0.2311788299864542	0.4056192940793584E-01
16	6.366197586059570	2.984155178070068	0.8803357229221538	0.2463783934562513	0.4396166981886961E-01
17	6.366197586059570	3.183098793029785	0.8644684692379431	0.2613451109847379	0.4748813448298558E-01
18	6.366197586059570	3.382042407989502	0.8476754862330025	0.2759993849277025	0.5114443397913725E-01
19	6.366197586059570	3.580986022949219	0.8299748219683939	0.2903937123326671	0.5495942460460643E-01
20	6.366197586059570	3.779929876327515	0.8113853758288153	0.3044377254615032	0.5893742794434264E-01
21	6.366197586059570	3.978873729705811	0.7919271255422354	0.3182001225075744	0.6311535219645088E-01
22	6.366197586059570	4.177817344665527	0.7716207866504400	0.3315737440180880	0.6749880763125785E-01
23	6.366197586059570	4.376760959625244	0.7504881530384858	0.3446457301902275	0.7213548362344269E-01
24	6.366197586059570	4.575704574584961	0.7285519266699735	0.3572882730295392	0.7703323517086873E-01
25	6.366197586059570	4.774648189544678	0.7058355473223206	0.3696146779050996	0.8225561791776895E-01

```
26 6.366197586059570 4.973592281341553 0.6823633628514890 0.3814727192006888     0.8781479072117103E-01
27 6.366197586059570 5.172535896301270 0.6581604589272583 0.3930032119310564     0.9379597432375204E-01
28 6.366197586059570 5.371479511260986 0.6332527725430416 0.4040248348885582     0.1002191806831087
29 6.366197586059570 5.570423126220703 0.6076669785060698 0.4147087404399114     0.1072006548481333
30 6.366197586059570 5.769366741180420 0.5814305461923003 0.4248461154024210     0.1147766893781889
31 6.366197586059570 5.968310356140137 0.5545713990169615 0.4346395628159276     0.1231095301199438
32 6.366197586059570 6.167253971099854 0.5271184819836450 0.4438478750102748     0.1322652592580781
33 6.366197586059570 6.366197586059570 0.4991011373803042 0.4527084358458642     0.1424786902390510
34 6.366197586059570 6.565141677856445 0.4705492750439835 0.4609494734012290     0.1538750123971094
35 6.366197586059570 6.764085292816162 0.4414935993804526 0.4688408564124689     0.1668095555761726
36 6.366197586059570 6.963028907775879 0.4119650701925724 0.4760784652899361     0.1815256619595133
37 6.366197586059570 7.161972522735596 0.3819954418519292 0.4829653814012340     0.1985992973621732
38 6.366197586059570 7.360916137695312 0.3516167808821093 0.4891677257676419     0.2185285018265267
39 6.366197586059570 7.559859752655029 0.3208615794685162 0.4950219805014384     0.2423408644784115
40 6.366197586059570 7.758803367614746 0.2897628122132800 0.5001623525420927     0.2711366573082435
41 6.366197586059570 7.957747459411621 0.2583537374930758 0.5049590953718078     0.3070162317117941
42 6.366197586059570 8.156690597534180 0.2266680535351235 0.5090163912019857     0.3527453835652479
43 6.366197586059570 8.355634689331055 0.1947396004539154 0.5127349778691294     0.4135790745122910
44 6.366197586059570 8.554577827453613 0.1626026156483072 0.5156912536249081     0.4981752238333208
45 6.366197586059570 8.753521919250488 0.1302915351593355 0.5183156186914784     0.6248819515135644
46 6.366197586059570 8.952465057373047 0.9784093691530989E-01 0.5201626105496894 0.8350998686338192
47 6.366197586059570 9.151409149169922 0.6528554782617585E-01 0.5216153106837943 1.255027274819390
48 6.366197586059570 9.350353240966797 0.3266026151942414E-01 0.5253168416062247 2.526513091590432

49 6.366197586059570 9.549296379089355 0.3209458529977964E-16 0.4650364382616894 2276014904117734.
```

```
      program wire
      implicit real*8(a-h,o-z)
      common /vars/ u(5),rhs(5),z,neqs
      common /intu/ step,hstep,nrk
      open(unit=1,file='fort.91',status='old')
      halfpi=dacos(0.0d0)
      pi=2.0*halfpi
      dpr=180.0/pi
      neqs=1
      en=2.0d0
      print 1000
      read *,nwire,zs,step
      do 500 iwire=1,nwire
      z=zs
      arg=float(2*iwire-1)/float(2*nwire)
      u(1)=dasin(arg)/en
      phi0=u(1)*dpr
      sn=z
      phin=phi0
      test=90.0d0/en
      hstep=0.5*step
      write(2,*) sn,phin
      call getg(r,z,g,step,drdz)
      x=r*dcosd(phin)
      y=r*dsind(phin)
      write(3,*) x,y,z
*  Commence Integration
      do 100 npr=1,500
      do 80 nrk=1,4
      call getg(r,z,g,step,drdz)
      if(g.lt.0.0d0) go to 450
      rhs(1)=g*sqrt(1.0d0+drdz**2)*tan(en*u(1))
      call rkgill
80    continue
      snm1=sn
      phinm1=phin
      sn=z
      phin=dpr*u(1)
      write(2,*) sn,phin
      call getg(r,z,g,step,drdz)
```

```
      x=r*dcosd(phin)
      y=r*dsind(phin)
      write(3,*) x,y,z
*
*  If we have passed phi=90/n, we stop.
      if(phin.ge.test) go to 450
*
*  If we are about to reach phi=90/n, then on the basis of symmetry
*  we interpolate for the z value corresponding to phi=90/n.
      if(phin+3.0*dpr*step*g*tan(en*u(1)).ge.test) then
      pphin=test-phin
      pphinm1=test-phinm1
      s=(snm1*pphin**2-sn*pphinm1**2)/(pphin**2-pphinm1**2)
      phi=test
      write(2,*) s,phi
      call getg(r,s,g,step,drdz)
      x=r*dcosd(phi)
      y=r*dsind(phi)
      write(3,*) x,y,s
      go to 450
      else
      endif
100   continue
450   write(2,*) '&'
      write(3,*) '&'
500   continue
      stop
1000  format(' Enter number of wires, initial z, and integration',
     $' step size.')
      end
      subroutine getg(r,s,g,step,drdz)
*
*  drdz is the derivative of the quadratic passing through
*  the equally spaced points r1 r2 r3.
*
      implicit real*8(a-h,o-z)
      rewind 1
      read(1,*) r1,s1,g1
      read(1,*) r2,s2,g2
      if(s.le.s1) then
      r=r1
      drdz=(r2-r1)/step
      g=g1
```

```
      return
      endif
10    read(1,*,end=100) r3,s3,g3
      if(s.le.s2) then
      g=(g2-g1)*((s-s1)/(s2-s1))+g1
      r=(r2-r1)*((s-s1)/(s2-s1))+r1
      a=(r3+r1-2.0d0*r2)/(2.0d0*step**2)
      drdz=2.0d0*a*s+(r2-r1)/step-(2.0d0*s1+step)*a
      return
      else
      r1=r2
      s1=s2
      r2=r3
      s2=s3
      g1=g2
      g2=g3
      go to 10
      endif
100   g=-1.0
      return
      end
      SUBROUTINE RKGILL
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION G(5)
      SAVE G
      COMMON / VARS / U(5), RHS(5), Z,NEQS
      COMMON / INTU / STEP, HSTEP, NRK
      DATA W1, W2, W3, W4 /
     $.5D0,.2928932188134525D0,1.7071067811865475D0,
     $.1666666666666667D0 /
*
*  W1=1/2
*  W2=(2-SQRT(2))/2
*  W3=(2+SQRT(2))/2
*  W4=1/6
*  FOURTH-ORDER RUNGE-KUTTA-GILL INTEGRATION,
*  NRK DENOTES THE THE INTERNAL STEP TO PERFORM
      GO TO (100,200,300,400) NRK
100   DO 150 N=1,NEQS
      G(N) = STEP*RHS(N)
      U(N) = U(N) + W1*G(N)
150   CONTINUE
      Z = Z + HSTEP
```

```
      RETURN
200   DO 250 N=1,NEQS
      AAAA = STEP*RHS(N)
      U(N) = U(N) + W2*(AAAA - G(N))
      G(N) = G(N) + W2*(AAAA + AAAA - 3.*G(N))
250   CONTINUE
      RETURN
300   DO 350 N=1,NEQS
       AAAA = STEP*RHS(N)
      U(N) = U(N) + W3*(AAAA - G(N))
      G(N) = G(N) + W3*(AAAA + AAAA - 3.*G(N))
350   CONTINUE
      Z = Z + HSTEP
      RETURN
400   DO 450 N=1,NEQS
      AAAA = STEP*RHS(N)
      U(N) = U(N) + W4*(AAAA - 2.*G(N))
450   CONTINUE
      RETURN
      END
```

Converting x,y,z, wires into DXF

```
#!/bin/sh
echo '1i\'            > /tmp/baf.$$.sed
echo 'LINE'          >> /tmp/baf.$$.sed
echo '/LINE-END/a\' >> /tmp/baf.$$.sed
echo 'LINE'          >> /tmp/baf.$$.sed
#
sed -e 's/[DdE]/e/' -e '/^ *$/d' -e 's/&/LINE-END/' -e 's/LINe/LINE/' $1
sed -f /tmp/baf.$$.sed | sed -e '$d' |\
nawk '
BEGIN {
icount = 0
print 0
print "SECTION"
print 2
print "ENTITIES"
}
/LINE/,/LINE-END/ {
if ( $1 ~ "LINE" ) {
   if ( icount > 0 )
```

```
      icount = 0
   }
else {
   if ( icount > 0 ) {
      xlast = x
      ylast = y
      zlast = z
   }
   x = $1
   y = $2
   z = $3
   if ( x == "0." ) x = "0.0"
   if ( y == "0." ) y = "0.0"
   if ( z == "0." ) z = "0.0"
   if ( icount > 0 ) {
      line_hdr()
      print 10
      print xlast
      print 20
      print ylast
      print 30
      print zlast
      print 11
      print x
      print 21
      print y
      print 31
      print z
   }
   icount++
  }
}

END {
print 0
print "ENDSEC"
print 0
print "EOF"
}
function line_hdr(){
print 0
print "LINE"
print 8
```

```
print 0
}' -

rm -f /tmp/baf.$$.sed
```

Bibliography

[1] Reference manual for the poisson/superfish group of codes. *Los Alamos Accelerator Code Group, LA-UR-87-126*, January 1987.

[2] V.Brady. Three-dimensional field components. *Lawrence Berkeley Laboratory, HIFAR Note-261*, January 1990.