Title
The "Inverse Problem" to the Evaluation of Magnetic Fields

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

Author
Caspi, S.

Publication Date
2011-01-12
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 \( \sum_m C_m I_2 \left( (2m - 1) \frac{\pi \rho}{L} \right) \cos \left( (2m - 1) \frac{\pi z}{L} \right) \sin 2\phi \). 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 \( \rho, z \) 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 \( \phi \) 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 \( \rho, z \) 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 \( \Omega^I \) may be taken to be a given function that will vanish along the axis \( \rho = 0 \), while the function \( \Omega^{II} \) (that remains to be found) should tend to zero at remote distances \( (\rho \to \infty) \). Each of the functions \( \Omega^I \) & \( \Omega^{II} \) 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 \( \Omega^I \) & \( \Omega^{II} \) 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 \( \frac{\partial \Omega^I}{\partial n} \) should become equal to the prescribed value of \( \frac{\partial \Omega^{II}}{\partial n} \) 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 \( \rho, z \) space for the function \( \Omega^{II} \). We thus may anticipate, in particular, that if the function \( \Omega^I \) 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 \( \Omega^{II} \) on the opposite side of the interface at that location and that \( \Omega^{II} \) then will grow (to less negative values) as \( \rho \) increases further — thus maintaining continuity at the interface of the normal derivatives of potential and permitting \( \Omega^{II} \) to tend towards zero at large \( \rho \).

If the problem thus posed becomes solved, as by a relaxation process applied to the function \( \Omega^{II} \), 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 \( \rho, z \) plane) is given by the discontinuity in the \( \phi \) component of field [i.e., by the difference between \(- \frac{1}{\mu_0} \frac{\partial \Omega^{II}}{\partial \phi} \sin n\phi \) and \(+ \frac{1}{\mu_0} \frac{\partial \Omega^I}{\partial \phi} \sin n\phi \), or \( \frac{n}{\mu_0 \rho} (\Omega^I - \Omega^{II}) \cos n\phi \)], while the discontinuity in the longitudinal derivative or longitudinal component of field similarly gives the \( \phi \) component of current density as \( \frac{1}{\mu_0} \left( \frac{\partial \Omega^{II}}{\partial \phi} - \frac{\partial \Omega^I}{\partial \phi} \right) \sin n\phi \) —— 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 \( \Omega^{II} \) 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 $\Omega^{II}$ shall be taken as equal to the known (prescribed) normal derivative of $\Omega^I$).

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 $\Omega^{II}$ is to be evaluated. The method instead will employ the assignment of "provisional" values for $\Omega^{II}$ 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 $\Omega^{II}$ and its derivative $\frac{\partial \Omega^{II}}{\partial s}$ employed (together with the corresponding known values of $\Omega^I$ 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 $\Omega^{II}$ at the interface may be taken simply as $-\Omega^I$ at the interface vertices. Moreover, when subsequently revising such provisional values of $\Omega^{II}$ 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 $\Omega^I$ to the normal derivative of the provisional present function $\Omega^{II}$ 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 $0 \leq z \leq \frac{L}{4}$ (or $-\frac{L}{4} \leq z \leq \frac{L}{4}$) with a local slope angle $\gamma$ 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

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

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

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

For each of the functions $\Omega_n^I$ & $\Omega_n^{II}$ 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

$$\begin{align*}
\frac{\partial \Omega_n}{\partial z} |_{z=0} &= 0 \\
\Omega_n |_{z=a} &= 0
\end{align*}$$

The requisite connection between the functions $\Omega_n^I$ & $\Omega_n^{II}$ occurs at $\rho_w = f(z)$ and explicitly is

$$\frac{\partial \Omega_n^I}{\partial n} = \frac{\partial \Omega_n^{II}}{\partial n}$$

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:

$$\begin{align*}
\frac{\partial \Omega}{\partial n} &= \cos \gamma \frac{\partial \Omega}{\partial r} - \sin \gamma \frac{\partial \Omega}{\partial z} \\
\frac{\partial \Omega}{\partial s} &= \sin \gamma \frac{\partial \Omega}{\partial r} + \cos \gamma \frac{\partial \Omega}{\partial z}
\end{align*}$$

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 potentials 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:

\[ \Omega_{i,new}^{I} = \Omega_{i,old}^{I} [1 + \lambda(\text{Factor} - 1)] \]

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

\[
J_s = \frac{n}{\mu_0 \rho_w} \left( \frac{\Omega^I_n - \Omega^{II}_n}{\cos n\phi} \right)
\]

\[
J_\phi = -\frac{1}{\mu_0} \left( \frac{\partial \Omega^I_n}{\partial s} - \frac{\partial \Omega^{II}_n}{\partial s} \right) \sin n\phi
\]

(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

\[
\frac{d\phi}{ds} = \frac{1}{\rho_w J_s} J_\phi
\]

\[
= -\frac{1}{\rho_w n} \left( \frac{\partial \Omega^I_n}{\partial s} - \frac{\partial \Omega^{II}_n}{\partial s} \right) \tan n\phi
\]

\[
= g(r, z) \tan n\phi
\]

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 \( r \) )

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

POISSON output provides tables of \( \rho, z, \) and \( g(\rho, z) \), 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 \( \rho = a \)

\[
J = \cos \frac{\pi z}{L} \cos 2\phi e_x + \frac{\pi a}{2L} \sin \frac{\pi z}{L} \sin 2\phi e_\phi
\]

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

\[
V^I = \Omega^I \sin 2\phi = -\frac{2\pi}{10} \frac{\pi a^2}{L} K_2^2 \left( \frac{\pi a}{L} \right) L_2 \left( \frac{\pi \rho}{L} \right) \cos \left( \frac{\pi z}{L} \right) \sin 2\phi \quad \text{for} \quad \rho < a
\]

\[
V^{II} = \Omega^{II} \sin 2\phi = -\frac{2\pi}{10} \frac{\pi a^2}{L} L_2 \left( \frac{\pi a}{L} \right) K_2 \left( \frac{\pi \rho}{L} \right) \cos \left( \frac{\pi z}{L} \right) \sin 2\phi \quad \text{for} \quad \rho > a
\]
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 $\Omega^I$ the values of $\Omega^I$ 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.](image)
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 $\phi$ direction.
Figure 6 The function $g(s)$. 
Figure 7 The azimuthal angle $\phi$ of each wire on the cylindrical surface.
Figure 8  A 3D CAD drawing of the quad wire distribution.
Bibliography


Appendix A

Interpolation & Differentiation in POISSON

In the reprograming of POISSON the interpolation and interpolation-differentiation 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 \( z - z_0 \) due to the translational character of the coordinate \( Z \), but such a replacement would appear to have merit only in regard to arithmetic accuracy as affected by register-length limitations. Such a shift of origin may be understood to be applied in cartesian coordinate use of the unmodified 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 purposes of interpolation and/or interpolation-differentiation such function are to be fit (by "least square") to values of \( \Omega \) at neighboring vertices of the mesh.

In application to the modified 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 \( \Omega \) in EXTERIOR regions that extend essentially to \( \infty \) where \( \Omega \) 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 \( \Omega \) 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
<table>
<thead>
<tr>
<th>K</th>
<th>$\Omega^n_k(r,z)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$r^n$</td>
</tr>
<tr>
<td>2</td>
<td>$r^n z$</td>
</tr>
<tr>
<td>3</td>
<td>$(n+1)r^n z^2 - \frac{1}{2} r^{n+2}$</td>
</tr>
<tr>
<td>4</td>
<td>$(n+1)r^n z^3 - \frac{3}{2} r^{n+2} z$</td>
</tr>
<tr>
<td>5</td>
<td>$(n+1)(n+2)r^n z^4 - 3(n+2)r^{n+2} z^2 + \frac{3}{4} r^{n+4}$</td>
</tr>
<tr>
<td>6</td>
<td>$(n+1)(n+2)r^n z^5 - 5(n+2)r^{n+2} z^3 + \frac{15}{4} r^{n+4} z$</td>
</tr>
<tr>
<td>7</td>
<td>$(n+1)(n+2)(n+3)r^n z^6 - \frac{15}{2} (n+2)(n+3)r^{n+2} z^4 + \frac{45}{4} (n+3)r^{n+4} z^2 - \frac{15}{8} r^{n+6}$</td>
</tr>
</tbody>
</table>

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

**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

\[
\text{Quad cylinder}
= \text{geo NREG=2 XMIN=0.0 XMAX=50.0 YMIN=0.0 YMAX=9.549296586}
\]
\[
\text{dx=0.2 dy=0.2}
\]
\[
\text{reg mat = 1 den=-0.7957747 npoint=7}
\]
\[
\text{PO x=0.0 y=0.0}
\]
\[
\text{PO x=6.366197724 y=0.0}
\]
\[
\text{PO x=50.0 y=0.0}
\]
\[
\text{PO x=50.0 y=9.549296586}
\]
\[
\text{PO x=6.366197724 y=9.549296586}
\]
\[
\text{PO x=0.0 y=9.549296586}
\]
\[
\text{PO x=0.0 y=0.0}
\]
\[
\text{reg mat = 1025 den=-0.7957747 npoint=5}
\]
\[
\text{PO x=6.366197724 y=0.0}
\]
\[
\text{PO x=50.0 y=0.0}
\]
\[
\text{PO x=50.0 y=9.549296586}
\]
\[
\text{PO x=6.366197724 y=9.549296586}
\]
\[
\text{PO x=6.366197724 y=0.0}
\]

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 33 149
*131 2.0000000
*136 1
*137 0
*74 1.95
*75 1.96
*20 49 9
   33   1   2.096646424839027
   33   2   2.095523854076131
   33   3   2.092157343864345
   33   4   2.086550499147154
   33   5   2.078709232374369
   33   6   2.068642215134743
   33   7   2.056359950010085
   33   8   2.041875683245411
   33   9   2.025304933324081
  10   2.006365534948627
  11   1.9853776696954441
  12   1.962263811778074
  13   1.937048725420820
  14   1.909759386187659
  15   1.880425028341242
  16   1.849077063853275
  17   1.815749060916343
  18   1.780476726210482
  19   1.743297794786612
  20   1.704252096028298
  21   1.663381441056512
  22   1.620725959520142
  23   1.576342253877383
  24   1.530266903552044
  25   1.482552904754207
  26   1.433251350833951
  27   1.3824150535158054
  28   1.33098421181071
  29   1.276357478271286
  30   1.221249779518272
  31   1.16483435638509
  32   1.107171587002677
  33   1.048232222375472
  34   0.9883522867490240
  35   0.9273229985167589
  36   0.8653007093828187
  37   0.8023518662449801
  38   0.738543810378309
  39   0.6739449084515552
  40   0.608624236265957
  41   0.5426520144900224
  42   0.476096504511001
  43   0.409034346322512
  44   0.3415342133330507
  45   0.2736672685671230
  46   0.2055073082964059
  47   0.1371272514130845
```
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.1t.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.')</nend
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.1e.s1) then
  r=r1
  drdz=(r2-r1)/step
g=g1
return
defif
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
dend
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,
$.1666666666666667DO /
*
* 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 'li'
    > /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 'e' -e 's/^ *$/d' -e 's/&/LINE - END/' -e 's/LINE/LINE/' $1
sed -f /tmp/baf.$$ sed | sed -e '$d' |
awk '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
