Lawrence Berkeley National Laboratory
Recent Work

Title
MAGNETIC FIELD COMPUTATION FOR TWO-DIMENSIONAL AIR COIL SYSTEM

Permalink
https://escholarship.org/uc/item/0z43g6rc

Authors
Yourd, Roland
Halbach, Klaus.

Publication Date
1968-07-01

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

UCRL-18378
Cy. 2

RECEIVED
LAWRENCE
RADIATION LABORATORY
SEP 25 1968
LIBRARY AND
DOCUMENTS SECTION

University of California

Ernest O. Lawrence Radiation Laboratory

TWO-WEEK LOAN COPY

This is a Library Circulating Copy which may be borrowed for two weeks. For a personal retention copy, call Tech. Info. Division, Ext. 5545

MAGNETIC FIELD COMPUTATION FOR TWO-DIMENSIONAL AIR COIL SYSTEM

Roland Yourd and Klaus Halbach

July 1968

Berkeley, California

UCRL-18378
Cy. 2

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

UCRL-18378
UC-28 Particle Accel. and
High Volt. Mach.
TID-4500 (52nd Ed.)

UNIVERSITY OF CALIFORNIA

Lawrence Radiation Laboratory
Berkeley, California

AEC Contract No. W-7405-eng-48

MAGNETIC FIELD COMPUTATION FOR TWO-DIMENSIONAL AIR COIL SYSTEM

Roland Yourd and Klaus Halbach

July 1968

MAGNETIC FIELD COMPUTATION FOR TWO-DIMENSIONAL AIR COIL SYSTEM*

Roland Yourd and Klaus Halbach

Lawrence Radiation Laboratory
University of California
Berkeley, California

July 1968

ABSTRACT

Analytic formulas suitable for numerical computation are derived for two-dimensional air coil systems. The computer code, FIELDS, is described, and evaluation of a sextupole magnet is included as a sample problem. A complete listing of the program coding is also provided in an appendix.

I. INTRODUCTION

A method of calculating two-dimensional magnetic fields in vacuum will be developed consisting of evaluation of analytical complex formulas. A description of the resulting computer code, FIELDS, is included, and user instructions are provided including a sample sextupole magnet problem. The method treats infinite straight conductors with arbitrary cross section and uniform current density, j, in each conductor, and no permeable material may be present. Field calculations may be obtained both inside and outside the coil area without special treatment. In addition a multipole expansion of the fields is provided.

II. MAGNETIC FIELD EQUATIONS FOR TWO-DIMENSIONAL COILS

Complex algebra will be used in the following field derivations, and MKS units will be used throughout. Magnetic fields can be described by the complex quantity $H = H_x + iH_y$. Since only the complex conjugate of H, namely $H^* = H_x - iH_y$, is an analytic function of $z = x + iy$, this is the quantity used.

For a current filament with positive current I (positive in the $\vec{x} \times \vec{y}$ direction, or directed out of the page) shown in Fig. 1 at point z, the magnetic field at a point z_o has a strength equal to $I/(2\pi |z - z_o|)$. The field direction (perpendicular to the vector $z_o - z$) may be given by $i(z_o - z)/|z_o - z|$.

In equation form,

$$H = \frac{I}{2\pi |z - z_o|} \frac{i(z_o - z)}{|z_o - z|}$$

This may be rewritten as

$$H = \frac{I}{2\pi} \frac{i(z_o - z)}{(z_o - z)(z_o^* - z^*)} = \frac{Ii}{2\pi (z_o^* - z^*)} \tag{2}$$

where z^* and z_o^* are complex conjugates of z and z_o respectively. Introducing H^* in (2)

$$H^* = \frac{Ii}{2\pi (z - z_o)} \tag{3}$$

For several current filaments the fields at z_o may be obtained by superposition. Extending this to a conductor of arbitrary cross section with uniform current density j, the problem becomes one of integration over the cross section.

$$H^* = \frac{ji}{2\pi} \int \frac{1}{z - z_o} \, dx\, dy \tag{4}$$

The integral in (4) is of the form $\int F(z)\, dx\, dy$, where $F(z)$ is analytic and single valued over the integration region, and may have one singularity at $z = z_o$. (If no field values are to be evaluated inside the coil, z_o is outside the integration region, and $F(z)$ has no singularities). For a more general $F(z)$ (with "n" first order singularities inside the integration region) it is shown in Appendix A that the integral of (4) may be represented by

$$\frac{1}{2i} \left(\oint F(z)\, z^*\, dz - \sum_{\text{Sum over "n" poles}} \oint F(z)\, z_n^*\, dz \right)$$

The contour integrations in the sum are performed for infinitesimal circles about each pole.

In (4) $F(z) = 1/(z - z_o)$, and H^* may be written as

$$H^* = \frac{j}{4\pi} \oint \frac{z^* - z_o^*}{z - z_o} \, dz \qquad (5)$$

Equation (5) is valid for field points both inside and outside the coil region. This formulation differs slightly from that developed by R. A. Beth[1] in that he considers interior and exterior points separately. Equation (5) is more convenient to use for numerical computation since no checks are required to determine the type of point. Another advantage of (5) is that the absolute value of the integrand is always unity. For field points near the coil Beth's technique requires taking differences between large numbers over part of the integration contour. Use of (5) avoids both problems, and is used for calculating all field values in the computer code. For completeness, we derive (5) in the appendix with a procedure that differs from Beth's, and allows extension to other problems.

III. MULTIPOLE COEFFICIENTS

Multipole coefficients are obtained by expanding the fields in a power series over a region excluding all currents. In such a region the complex potential $G(z) = A + iV$ (where A and V are, respectively, the vector and scalar potentials) is analytic and $H^*(z)$ can be obtained from $G(z)$ through $H^*(z) = iG'(z)$. Therefore $G(z)$ may be expanded in a converging Taylor Series of the complex variable z_o.

$$G(z_o) = a_o + a_1 z_o + a_2 z_o^2 + \text{-----} + a_n z_o^n \qquad (7)$$

$$G(z_o) = \sum_{n=o}^{\infty} a_n z_o^n$$

Similarly

$$H^*(z_o) = i \sum_{n=1}^{\infty} n\, a_n\, z_o^{n-1} = \sum_{n=1}^{\infty} b_n\, z_o^{n-1} \tag{8}$$

where a_n and b_n are in general complex constants.

This Taylor Series has a convergence radius equal to the distance from the expansion origin to the closest singularity (the closest coil dimension, in this case). The series converges, then, only within a circle extending to the nearest coil boundary. In such a region (5) may be written as

$$H^* = \frac{j}{4\pi} \oint \frac{z^*}{z - z_o}\, dz = \frac{j}{4\pi} \oint \frac{z^*}{z(1 - \frac{z_o}{z})}\, dz \tag{9}$$

and for $\left| {}^{z_o}/_z \right| < 1$ this becomes

$$H^* = \frac{j}{4\pi} \oint \frac{z^*}{z} \sum_{n=1}^{\infty} \frac{z_o^{n-1}}{z^{n-1}}\, dz, \tag{9a}$$

Equating the two representations for H^* in this circular region, namely (8) and (9a)

$$\sum_{n=1}^{\infty} b_n\, z_o^{n-1} = \frac{j}{4\pi} \oint \frac{z^*}{z} \sum_{n=1}^{\infty} \frac{z_o^{n-1}}{z^{n-1}}\, dz, \tag{10}$$

and solving (10) for all values of b_n, one obtains

$$b_n = \frac{j}{4\pi} \oint \frac{z^*}{z^n}\, dz \tag{11}$$

Although certain coil symmetries produce only real or only imaginery coefficients, b_n in general may be complex coefficients.

IV. DESCRIPTION OF FIELDS PROGRAM

A Fortran IV computer code has been written to evaluate magnetic fields and multipole coefficients using equations (5) and (11). Input data requires only specification of coil boundary and current information, certain coil symmetry conditions, and a description of the desired grid for field evaluation points.

Coil areas are approximated by inscribed polygons to enable numerical treatment of the line integrals in (5) and (11). A "Trapezoid rule" technique is currently used for the integration. The user may change the number of polygon line segments at will, and also could easily implement additional integration methods (Simpson's rule among others). The symmetry order is treated automatically and only coordinate points for a single coil are actually specified.

The total current, I in amperes, per coil must be given, and a current density is obtained by dividing I by the calculated polygon area. Thus although the calculated model and physical current densities may differ slightly (since true coil area and polygon areas are not quite identical), the total current is identical for both.

A typical problem is shown in Fig. 2. The input data for this sextupole first must specify the total current per coil, I_1. The coil boundary is next specified (in a counter clockwise direction for a positive current element) by giving end points $((r_1, \theta_1)$ and $(r_2, \theta_2))$ for each coil side. A code K = 1 or K = 2 is used to indicate whether the coil side is a straight line segment or an arc segment respectively. Finally a value for N is specified to indicate a desired division of the coil segment into N equal length polygon sides.

Actual data cards for the coil boundary shown would be:

$$2,\ r_1,\ \theta_1,\ r_2,\ \theta_2,\ 50$$
$$1,\ r_3,\ \theta_3,\ 10$$
$$2,\ r_4,\ \theta_4,\ 60$$
$$1,\ r_1,\ \theta_1,\ 10$$

where K is either 1 or 2, units for r and θ are cm and radians respectively, and the desired division shown results in a 130 sided polygon approximation. Note that after the first boundary segment is described by first and last points, only the last point is required for each succeeding sector.

Two additional data cards are required to set up the rectangular edit grid. The first card specifies (in order) the number of field points in the x direction, the initial value for x, and the incremental distance between the points. The second card gives identical quantities for y.

All input data is read by subroutine DATA which allows complete field freedom in punching data on the cards. The user should become familiar with the use of this routine as described by Bill Benson's write up: "Subroutine Data - A Fortran 66 Format Free Input Routine", June 9, 1965. (Ref. 2)

The symmetry order of the coil system is specified through a Fortran data statement in the main program. The variable name NCOIL is used, and it represents the number of rotationally symmetrical coils (dipole = 2, sextupole = 6, etc.). In some cases, the user may wish to describe more than one coil in a symmetrical set (provision for small tuning coils, variable current densities over portions of the total coil, etc.). The variable name IGRP represents the number of separate coils comprising a single symmetrical group or set. If IGRP $\neq$ 1, then separate input cards must describe each of the coil boundaries in the set.

Multipole coefficients are obtained from (11) and multiplied by r_a^{n-1}, where r_a is the radius of the expansion aperture (or a value specified by the user). The thus normalized multipole coefficients describe, according to (8), the contribution of the individual multipoles to the field at the normalization radius. It can easily be shown that rotational symmetries will result in certain "b_n" values being exactly zero for particular "n" in the series of (11). If NCOIL represents the number of rotationally symmetrical coil sets, a general expression for the particular "n" orders that may exist in many coil types is

$$n = \frac{\text{NCOIL}}{2} \quad (2M - 1)$$

where M = 1, 2, 3, ---- .

The user may wish to include expressions for additional coil types.

A complete listing of the program coding and the results from the sextupole sample problem are in Appendix B.

APPENDIX

A. Evaluation of $\int F(z)\,dx\,dy$

The function F(z) must be analytic, single valued over the integration region, and must have no singularities higher than first order inside the boundary. It is convenient to represent F(z) as the derivative of another function G(z).

$$F(z) = G'(z)$$

$$I = \int F(z)\,dx\,dy = \int G'(z)\,dx\,dy \qquad \text{(A1)}$$

Since the integration is over a total area, and since integration over an infinitesimal circular area containing a simple pole contributes only infinitesimally to this total, the region in Fig. A1 may be divided into strips, thus excluding the area inside of the infinitesimal circles around each pole. Integrating first over x, (A1) becomes

$$I = \sum_{\text{sum for all strips}} \int (G(z, \text{right}) - G(z, \text{left})\;dy = \sum_{\text{strips}} \oint G(z)\;dy \qquad \text{(A2)}$$

Once the contour is defined, G(z) becomes a function of y alone since $z = x(y) + iy$ for points on the boundary. The line integral of (A2) is integrated by parts.

Let

$$u = G(z) \qquad du = G'(z)\ (x'y + i)\ dy = G'(z)\ dz$$

$$dv = dy \qquad v = y$$

then,

$$\oint G(z)\ dy = G(z)\ y \Big|_{\text{start}}^{\text{end}} - \oint G'(z)\ y\ dz \tag{A3}$$

Since G(z) is regular and single valued within each loop (strip), and since the path starts and ends at the same point, $G(z)\ y \Big|_{\text{start}}^{\text{end}} = 0$.

Replacing y by $\frac{1}{2i}(z^* - z)$ and G'(z) by F(z), (A3) becomes

$$\oint G(z)\ dy = \frac{1}{2i} \oint F(z)\ (z^* - z)\ dz \tag{A4}$$

Since all poles have been excluded from the integration strips,

$\frac{1}{2i} \oint F(z)\ z\ dz$ must be zero from Cauchy's integral formula.

Therefore

$$I = \frac{1}{2i} \sum_{\substack{\text{sum over}\\ \text{all strips}}} \oint F(z)\ z^*\ dz = \frac{1}{2i} \oint_{\substack{\text{outer}\\ \text{contour}}} F(z)\ z^*\ dz - \frac{1}{2i} \sum_{\substack{\text{each pole}\\ \text{inside contour}}} \oint F(z)\ z^*\ dz \tag{A5}$$

The integrals about each pole must be taken over infinitesimal circular areas.

It is easy to show that for this case

$$\oint_{\text{pole}} F(z)\ z_n^*\ dz = z_n^* \oint_{\text{pole}} F(z)\ dz, \tag{A6}$$

If there is one or no singularity inside the outer contour, (A5) can be expressed as

$$I = \frac{1}{2i} \oint_{\substack{\text{outer}\\\text{contour}}} F(z)\,(z^* - z_n^*)\,dz \tag{A7}$$

It becomes evident by inspection that if there are n singularities, not all of which have to be inside the contour, I becomes

$$I = \int_{\substack{\text{area}\\\text{inside contour}}} F(z)\,dx\,dy = \frac{1}{2i} \oint_{\text{contour}} F(z)\,\Big(z^* - \sum_n z_n^* R_n(z)\Big)\,dz \tag{A8}$$

with

$$R_n(z) = \frac{P_n(z)}{P_n(z_n)}\ , \quad \text{and} \quad P_n(z) = \frac{\prod_\gamma (z - z_\gamma)}{z - z_n}$$

```
      PROGRAM FIELDS(INPUT,OUTPUT)

      COMMON/BDRY/X(2000),Y(2000),NPTS(5),Z(2000),ZS(2000)
      COMMON/GRID/X0,Y0,Z0,Z0S
      COMMON/MPOL/RNORM
      COMMON/XY/IXY

      DATA IDIM/2000/
      DIMENSION CUR(5), CURDEN(5)

C     IGRP REPRESENTS NUMBER (UP TO 5) OF COILS PER GROUP OR SET
      DATA IGRP/1/

C     NCOIL REPRESENTS NUMBER OF ROTATIONALLY SYMMETRICAL SETS
C     (DIPOLE = 2, SEXTUPOLE = 6, ETC)
      DATA NCOIL/4/

C     IALT = -1 IF SIGNS OF CURRENT ELEMENTS ALTERNATE, OTHERWISE +1
      DATA IALT/-1/

C     RN IS THE NORMALIZING RADIUS USED IN THE MULTIPOLE ANALYSIS.
C     IF RN IS NEGATIVE THE POINT ON COIL NEAREST ORIGIN IS USED FOR
C     NORMALIZATION.
      DATA RN/-1./

C     IF SENTINEL IXY EQUALS 1, THE USER MAY INPUT DATA IN (X,Y)
C     INSTEAD OF (R,THETA) COORDS.
      DATA IXY/1/

      DATA (CUR(I), I=1,5)/5*0.0/
      COMPLEX SUM, HSTAR, ROT
      COMPLEX Z, ZS, Z0, Z0S

C     FIRST DATA CARD CONTAINS CURRENTS -- C1, C2, ... S
C     AN S IS REQUIRED IF LESS THAN 5 CURRENT ELEMENTS ARE DESCRIBED.
C     DESCRIBE BOUNDARY OF COIL IN A COUNTER CLOCKWISE DIRECTION.
C     ENCLOSE POSITIVE CURRENT CARRYING ELEMENT
C     REGION MUST CLOSE -- FIRST PT AND LAST PT COORDS MUST BE IDENTICAL.
C     FIRST CARD READ MUST CONTAIN (IN ORDER), THE KIND OF CURVE
C     SEGMENT  ( 1 FOR STRAIGHT LINE,  2 FOR CIRCULAR ARC ),  R1,  PHI1,
C     R2, PHI2, AND THE NUMBER OF INTERVALS DESIRED.
C     FOLLOWING CARDS CONTAIN ONLY THE KIND,  R2,  PHI2,  AND NUMBER
C     OF INTERVALS.

      PI = 3.1415926536898

C     READ CURRENTS FOR COILS IN GROUP
      CALL DATA(CUR,5)

      DO 700 N=1,IGRP

      PRINT 600, N
  600 FORMAT(1H1*    DATA CARDS FOR COIL BDRY *I3* ARE LISTED BELOW*///)

      CALL DATA(KIND,1,R1,1,PHI1,1,R2,1,PHI2,1,NINT,1)

      R0 = R1
      PHI0 = PHI1

      IF(N.EQ.1) GO TO 8
```

```
      NPTS(N) = NPTS(N-1)
      GO TO 9
    8 NPTS(1)=1

    9 M = NPTS(N)
      X(M)=R1
      Y(M)=PHI1
      IF(IXY.EQ.1) GO TO 10
      X(M) = R1 * COS(PHI1)
      Y(M) = R1 * SIN(PHI1)

   10 IF(KIND .EQ. 1) CALL LINE(R1,PHI1,R2,PHI2,NINT,N)
      IF(KIND .EQ. 2) CALL  ARC(R1,PHI1,R2,PHI2,NINT,N)

C     HAS REGION BEEN CLOSED
   20 IF((R2.EQ.R0).AND.(PHI2.EQ.PHI0)) GO TO 1000
      R1 = R2
      PHI1 = PHI2
      CALL DATA(KIND,1,R2,1,PHI2,1,NINT,1)

      IF(KIND .EQ. 1) CALL LINE(R1,PHI1,R2,PHI2,NINT,N)
      IF(KIND .EQ. 2) CALL  ARC(R1,PHI1,R2,PHI2,NINT,N)

      IF(NPTS(N) .GT. IDIM) GO TO 25
      GO TO 20

C     DIMENSIONS HAVE BEEN EXCEEDED
   25 PRINT 26, NPTS(N), IDIM
   26 FORMAT(///*  DIMENSIONS OF X AND Y EXCEEDED*/*  NPTS = *I5
     **  , MAX ALLOWED IS*I5 //)
      STOP

 1000 CONTINUE
  700 CONTINUE

C     CALCULATE COIL AREAS AND CURDENS
      DO 18 I=1,IGRP
C     FIRST AND LAST PTS ARE SAME -- SUBTRACT ONE FROM NPTS
      NPTS(I) = NPTS(I) - 1
      CALL AREA(A,I)
      CURDEN(I) = CUR(I)/A
      PRINT 66,  CURDEN(I)
   66 FORMAT(/*   CURDEN = *E15.6* A/CM2*)
   18 CONTINUE

C     CALCULATE Z AND ZSTAR FROM X,Y PAIRS
C     ALSO OBTAIN NORMALIZING RADIUS FOR MULTIPOLES
      RNORM = 10000.
      IST = NPTS(IGRP)
      DO 19 K= 1,IST
      R = SQRT(X(K)*X(K) + Y(K)*Y(K))
      IF(R.LT.RNORM) RNORM = R
      Z(K) = CMPLX(X(K), Y(K))
      ZS(K) = CMPLX(X(K), -Y(K))
   19 CONTINUE
      IF(RN.GT.0.) RNORM = RN

C     SET UP EDIT GRID
      PRINT 602
  602 FORMAT(///*   DATA CARDS READ TO SET UP GRID ARE LISTED BELOW*///)
      CALL DATA(NX,1,X01,1,DELX0,1)
```

```
      CALL DATA(NY,1,Y01,1,DELY0,1)

      XP = X01
      YP = Y01

C     CALCULATE POWERS FOR MULTIPOLES

      ROT = CMPLX(0.,1.)
      PRINT 301
  301 FORMAT(1H1 *  MULTIPOLE ANALYSIS*//)
      PRINT 302, RNORM
  302 FORMAT(2X,*THE NORMALIZING RADIUS IS*, F10.4, *CM*//)
      PRINT 303
  303 FORMAT(47H  BSTAR =  SUM OVER N ( CN*((Z0/R) ** (N-1) ))     ///)

      DO 11 M=1,10
C     ANY EXPRESSION FOR MULTIPOLE ANALYSIS (IPOWER) MAY BE SUBSTITUTED
C     FOR THE GENERAL ONE USED BELOW
      IPOWER = (NCOIL/2)*(2*M-1)
      HSTAR = CMPLX(0.,0.)
      DO 899 I=1,IGRP
      CALL SUMMM(SUM, I, IPOWER)
      HSTAR = HSTAR - CURDEN(I) * 0.1*ROT*SUM
  899 CONTINUE
      HSTAR = FLOAT(NCOIL) * HSTAR * RNORM
      PRINT 300, IPOWER, HSTAR
  300 FORMAT(/*  N = *I3, 10X,  *    CN = * 2E15.6)
   11 CONTINUE

      PRINT 201
  201 FORMAT(1H1,14X,*X(CM)*,5X,*Y(CM)*,12X,*BX(G)*13X,*BY(G)*,13X,*BT(G
     X)*//)

      DO 500 J=1,NY

      DO 501 JJ=1,NX
      HSTAR = CMPLX(0.,0.)

      DO 499 JI = 1,NCOIL
      KNT = JI + 1
      ISN = (IALT)**KNT
      ANG = FLOAT(JI-1)*2.*PI/FLOAT(NCOIL)
      ROT = CMPLX(0.,-ANG)

      Z0 = CMPLX(XP,YP)
      Z0 = Z0 * CEXP(ROT)
      Z0S = CMPLX(XP,-YP)
      Z0S = Z0S * CEXP(ROT)
      X0 = REAL(Z0)
      Y0 = AIMAG(Z0)

      DO 550 I=1,IGRP
      CALL SUMS(SUM,I)

      IF(CABS(SUM) .EQ. 0.) GO TO 77

      SUM = SUM * CURDEN(I)* CEXP(ROT)* 0.1
      HSTAR = HSTAR + SUM*FLOAT(ISN)

  550 CONTINUE

  499 CONTINUE
```

```
      BX = REAL(HSTAR)
      BY = -AIMAG(HSTAR)
      B = SQRT(BX*BX + BY*BY)

      PRINT 200, XP, YP, BX, BY, B
  200 FORMAT(/10X, 2F10.4,5X,3(E15.6,3X))

      GO TO 78
   77 PRINT 202, XP, YP
  202 FORMAT(/10X,2F10.4,10X* EDIT PT IS ON COIL BDRY*)
   78 XP = XP + DELX0
  501 CONTINUE

      XP = X01
      YP = YP + DELY0
  500 CONTINUE

      END
```

```
      SUBROUTINE AREA(A,I)

      COMMON/BDRY/X(2000),Y(2000),NPTS(5),Z(2000),ZS(2000)
      COMMON/GRID/X0,Y0,Z0,Z0S

      IF(I.EQ.1) GO TO 1
      IS = NPTS(I-1) + 1
      GO TO 2

    1 IS = 1
    2 IE = NPTS(I)

      AS1 = X(IS)*(Y(IS + 1)  -  Y(IE))
      AS2 = X(IE)*(Y(IS)  -  Y(IE - 1))
      A = AS1 + AS2

      JS = IS + 1
      JE = IE - 1

      DO 3 J= JS, JE

      AS = X(J) * (Y(J+1) - Y(J-1))
      A = A + AS

    3 CONTINUE

      A = A * 0.5

      RETURN
      END
```

```
      SUBROUTINE SUMS(SUM,I)

      COMMON/BDRY/X(2000),Y(2000),NPTS(5),Z(2000),ZS(2000)
      COMMON/GRID/X0,Y0,Z0,Z0S

      COMPLEX Z, ZS, Z0,Z0S
      COMPLEX SUM, AS

      SUM = CMPLX(0.,0.)
      AS  = CMPLX(0.,0.)

      IF(I.EQ.1) GO TO 1
      IS = NPTS(I-1) + 1
      GO TO 2
    1 IS = 1
    2 IE = NPTS(I)
      IF(CABS(Z(IS) - Z0) .EQ. 0.     .OR.   CABS(Z(IE)-Z0) .EQ. 0.) GO
     * TO 8

      SUM = ((ZS(IS) - Z0S     )/(Z(IS)-Z0     ))*(Z(IS+1)-Z(IE))
      AS = ((ZS(IE)-Z0S     )/(Z(IE)-Z0     ))*(Z(IS)-Z(IE-1))
      SUM = SUM+AS

      JS = IS + 1
      JE = IE - 1

      DO 3 J = JS, JE

      IF(CABS(Z(J) - Z0) .EQ. 0.)  GO TO 8
      AS = ((ZS(J)-Z0S    )/(Z(J)-Z0     ))*(Z(J+1)-Z(J-1))

      SUM = SUM + AS

    3 CONTINUE

      SUM = SUM * 0.5

      RETURN
    8 SUM = CMPLX(0.,0.)
      RETURN
      END
```

```
      SUBROUTINE SUMMM(SUM,I,IPOWER)

      COMMON/BDRY/X(2000),Y(2000),NPTS(5),Z(2000),ZS(2000)
      COMMON/GRID/X0,Y0,Z0,Z0S
      COMMON/MPOL/R

      COMPLEX Z, ZS, Z0,Z0S
      COMPLEX SUM, AS

      SUM = CMPLX(0.,0.)
      AS  = CMPLX(0.,0.)

      IF(I.EQ.1) GO TO 1
      IS = NPTS(I-1) + 1
      GO TO 2
    1 IS = 1
    2 IE = NPTS(I)

      SUM = ((ZS(IS)/R)/((Z(IS)/R)**IPOWER))*(Z(IS+1)/R-Z(IE)/R)
      AS = ((ZS(IE)/R)/((Z(IE)/R)**IPOWER))*(Z(IS)/R-Z(IE-1)/R)
      SUM = SUM+AS

      JS = IS + 1
      JE = IE - 1

      DO 3 J = JS, JE

      AS = ((ZS(J)/R)/((Z(J)/R)**IPOWER))*(Z(J+1)/R-Z(J-1)/R)

      SUM = SUM + AS

    3 CONTINUE

      SUM = SUM * 0.5

      RETURN
      END
```

```
      SUBROUTINE LINE(R1,PHI1,R2,PHI2,NINT,N)
      COMMON/BDRY/X(2000),Y(2000),NPTS(5),Z(2000),ZS(2000)
      COMMON/GRID/X0,Y0,Z0,Z0S
      COMMON/XY/IXY

      IEND = NPTS(N)   + NINT
      X(IEND) = R2
      Y(IEND) = PHI2
      IF(IXY.EQ.1) GO TO 10
      X(IEND)=R2*COS(PHI2)          $  Y(IEND)=R2*SIN(PHI2)

   10 NN = NPTS(N)
      XDIFF = (X(IEND) - X(NN)        )/FLOAT(NINT)
      YDIFF = (Y(IEND) - Y(NN)        )/FLOAT(NINT)
      NSTART = NPTS(N) + 1
      IK = IEND - 1
      DO 1 I=NSTART,IK

      X(I) = X(I-1) + XDIFF
      Y(I) = Y(I-1) + YDIFF
    1 CONTINUE

      NPTS(N) = IEND
      RETURN
      END
```

```
      SUBROUTINE ARC(R1,PHI1,R2,PHI2,NINT,N)
      COMMON/BDRY/X(2000),Y(2000),NPTS(5),Z(2000),ZS(2000)
      COMMON/GRID/X0,Y0,Z0,Z0S
      COMMON/XY/IXY

      IF(IXY.EQ.1) GO TO 200
      IF(R1.EQ.R2) GO TO 2
C     PTS NOT ON ARC
      PRINT 100, R1,R2
  100 FORMAT(///*  FIRST AND LAST PTS OF SEGMENT DONT HAVE SAME RADIUS*/
     **   R1 = *F10.4,*  R2 = *F10.4///)
      STOP
  200 PRINT 201
  201 FORMAT(///*  MUST USE (R,THETA) COORD SYSTEM TO DESCRIBE ARCS,  (I
     *XY MAY NOT BE 1)*//)
      STOP

    2 CONTINUE

      IEND = NPTS(N) + NINT
      X(IEND)=R2*COS(PHI2)    $  Y(IEND)=R2*SIN(PHI2)
      PHIDIF = (PHI2 - PHI1)/FLOAT(NINT)
      NSTART = NPTS(N) + 1
      PHI = PHI1
      IK = IEND - 1
      DO 1 I=NSTART,IK

      PHI = PHI+PHIDIF

      X(I)=R2*COS(PHI)
      Y(I)=R2*SIN(PHI)
    1 CONTINUE
      NPTS(N) = IEND
      RETURN
      END
```

```
      SUBROUTINE DATA(X1,N1,X2,N2,X3,N3,X4,N4,X5,N5,X6,N6,X7,N7)
C 53
      COMMON/REDMAN/FLTING,TYPX,PRINX,VALUED,CFLAG,REPEAT,RC,VALUE,
     # STAR,K,R,DOLLAR,S,C,
     # COLCT,COLMAX,TYPTAB(7),DECIDE(10),XPA(4),XPB(10),XPC(10),
     # BLANK,BITS,MINUS,A,ALLBIT,TOOBIG,CARDCT
     COMMON /CCNAP/ I
      INTEGER TYPX,PRINX,DELTAJ,VALU,GOTX,CFLAG,REPEAT,RC,STAR
      INTEGER K,R,DOLLAR,S,C,CARDCT,COLCT
$$$$$
      INTEGER COLMAX
$$$$$
      DATA STAR,K,R,DOLLAR,S,C,COLCT,COLMAX/0000047,0000013,0000022,
     # 0000053,0000023,0000003,82,81/
$$$$$# 0000053,0000023,0000003,74,73/
      DATA TYPTAB/00050507050405050505B,05070505050505050707B,
     # 05050505050505010101B,01010101010101030307B,
     # 07070707070607020000B,0,0/
      DATA DECIDE/00503021212011413000,00504151515151513000,
     # 00615151515151513000,00613131313131313000,
     # 00506170717171713000,00617170717171713000,
     # 01117101313071313000,01117131313101313000,
     # 01117171717171713000,02121211212221211300/
      DATA XPA/1.0,1.0,1.0/
      DATA XPB/1.0,1.0E10,1.0E20,1.0E30,1.0E40,1.0E50,1.0E60,1.0E70,
     # 1.0E80,1.0E90/
      DATA XPC/1.0,1.0E1,1.0E2,1.0E3,1.0E4,1.0E5,1.0E6,1.0E7,1.0E8,
     # 1.0E9/
      DATA BLANK,BITS,MINUS,A,ALLBIT,TOOBIG/0000055,07700000000000000,
     # 0000046,0000001,037770000000000000000,000010000000000000000/
      DATA CARDCT/0/
      CFLAG=0
      REPEAT=0
      COLCT=COLMAX+1
$$$$$ COLCT=74
     CALL CCNARG
      IF(I.LE. 1)GO TO 103
      CALL DATAI(X1,N1)
      IF(I.LE. 3)GO TO 103
      CALL DATAI(X2,N2)
      IF(I.LE. 5)GO TO 103
      CALL DATAI(X3,N3)
      IF(I.LE. 7)GO TO 103
      CALL DATAI(X4,N4)
      IF(I.LE. 9)GO TO 103
      CALL DATAI(X5,N5)
      IF(I.LE.11)GO TO 103
      CALL DATAI(X6,N6)
      IF(I.LE.13)GO TO 103
      CALL DATAI(X7,N7)
103   PRINT 301,CARDCT,COLCT
301   FORMAT(1H 85X23HDATA POSITIONED AT CARDI4,8H, COLUMNI3)
$$$$$ FORMAT(24H DATA POSITIONED AT CARDI4,8H, COLUMNI3)
      RETURN
      END
```

```
      ASCENTF SUBROUTINE IDBYTE(I,J)
C 54
          PS                                                   000000000000
          PS                                                   000000000000
          PS                                                   000000000000
          PS                                                   000000000000
RDBYTE    PS                                                   000000000000
          SA2=B2              .X2=J                            56220
          SX3=1               .X3=1                            7130 000001
          IX4=X2-X3           .X4=J-1                          37423
          SX5=10              .X5=10                           7150 000012
          PX7=B0,X4           .X7=FLOAT(J-1)                   27704
          PX0=B0,X5           .X0=10.                          27005
          NX0=B0,X0                                            24000
          FX7=X7/X0           .X7=(J-1)/10                     44770
          UX7=B7,X7                                            26777
          LX7=B7,X7                                            22777
          PX7=B0,X7                                            27707
          UX1=B0,X7           .X1=N                            26107
          SA2=X1+B1           .X2=N+I=I(N+1)=II                53211
          SA3=B2              .X3=J                            56320
          PX7=B0,X5           .X7=10.                          27705
          PX0=B0,X1           .X0=FLOAT(N)                     27001
          DX7=X7*X0           .X7=10*N                         42770
          UX7=B0,X7                                            26707
          IX4=X3-X7           .X4=J-10*N                       37437
          SX1=6               .X1=6                            7110 000006
          PX7=B0,X1           .X7=6.                           27701
          PX0=B0,X4           .X0=FLOAT(J-10*N)                27004
          DX7=X7*X0           .X7=6*(J-10*N)                   42770
          UX6=B0,X7           .X6=MKK                          26607
          SB3=X6              .B3=MKK                          63360
          SX3=77B             .X3=A MASK                       7130 000077
          LX6=B3,X2           .SHIFT II  MKK PLACES            22632
          BX6=X6*X3           .MASK THE LOWER 6 BITS           11663
          JP    RDBYTE        .RETURN                          0200 L00001
          END
```

```
      SUBROUTINE DATAI(X,N)
C 55
      COMMON/REDMAN/FLTING,TYPX,PRINX,VALUED,CFLAG,REPEAT,RC,VALUE,
     * STAR,K,R,DOLLAR,S,C,
     * COLCT,COLMAX,TYPTAB(7),DECIDE(10),XPA(4),XPB(10),XPC(10),
     * BLANK,BITS,MINUS,A,ALLBIT,TOOBIG,CARDCT
      INTEGER TYPX,PRINX,DELTAJ,VALU,GOTX,CFLAG,REPEAT,RC,STAR
      INTEGER K,R,DOLLAR,S,C,CARDCT,COLCT
      DIMENSION X(1)
      EQUIVALENCE (VALUE,VALU)
      IF(N.LE.0)GO TO 101
      J=1
      DELTAJ=1
      DO 100 I=1,N
1     IF(CFLAG.EQ.0)GO TO 2
      CFLAG=0
      GO TO 10
2     IF(REPEAT.EQ.0)GO TO 3
      RC=RC-1
      IF(RC.GT.0)GO TO 10
      REPEAT=0
3     CALL RDNUM(VALUE)
      IF(TYPX)3,4,10
4     GOTX=7
      IF(VALU.EQ.STAR)GOTX=1
      IF(VALU.EQ.K)GOTX=2
      IF(VALU.EQ.R)GOTX=3
      IF(VALU.EQ.DOLLAR)GOTX=4
      IF(VALU.EQ.S)GOTX=5
      IF(VALU.EQ.C)GOTX=6
      GO TO (11,12,13,14,101,16,3),GOTX
11    CALL RDNUM(J)
      IF(TYPX.EQ.1.AND.FLTING.EQ.0.0)GO TO 3
      GO TO 17
12    CALL RDNUM(DELTAJ)
      IF(TYPX.EQ.1.AND.FLTING.EQ.0.0)GO TO 3
      GO TO 17
13    CALL RDNUM(RC)
      VALUE=VALUED
      REPEAT=1
      IF(TYPX.EQ.1.AND.FLTING.EQ.0.0.AND.RC.GT.0)GO TO 2
      GO TO 17
14    PRINX=1
      CALL RDNUM(VALUE)
      IF(VALU-DOLLAR)14,1,14
16    VALU=I-1
      CFLAG=1
      GO TO 101
C     ERROR###
17    CONTINUE
      PRINT 302,CARDCT,COLCT
302   FORMAT(15H ERROR AT CARD I4,5H,COL I2)
C     ERROR###
10    X(J)=VALUE
      VALUED=VALUE
      J=J+DELTAJ
100   CONTINUE
101   RETURN
      END
```

```
      SUBROUTINE RDNUM(ISIT)
C 56
       COMMON/REDMAN/FLTING,TYPX,PRINX,DUMMY(11),
     * COLCT,COLMAX,TYPTAB(7),DECIDE(10),XPA(4),XPB(10),XPC(10),
     * BLANK,BITS,MINUS,A,ALLBIT,TOOBIG,CARDCT
       INTEGER TYPX,PRINX,COLCT,COLMAX,BLANK,BITS,MINUS,A,ALLBIT
       INTEGER TOOBIG,CARDCT
       INTEGER VALUE,STAGE,CHARAC,CHTYPE,OLDCH,         WORD,EXPON
       INTEGER DIGITS,ESIGN,NSIGN
       DIMENSION BUFFER(8)
       EQUIVALENCE(X,I)
1      STAGE=1
       N=0
       DIGITS=0
       EXPON=0
       WORD=0
       TESDC=0
       NSIGN=0
       ESIGN=0
       FLTING=0.
11     OLDCH=CHARAC
       IF(COLCT-COLMAX)3,6,2
6      CHARAC=BLANK
       GO TO 7
2      READ 500,BUFFER
500    FORMAT(8A10)
$$$$$
       PRINT 510,BUFFER
510    FORMAT(1H 8A10)
$$$$$
       CARDCT=CARDCT+1
5      COLCT=1
3      CHARAC=IDBYTE(BUFFER,COLCT)
7      COLCT=COLCT+1
       CHTYPE=IDBYTE(TYPTAB,CHARAC+1)
       IF(CHTYPE.EQ.0)CHTYPE=8
130    STAGE=IDBYTE(DECIDE(STAGE),CHTYPE)
       GO TO(11,12,11,11,15,16,17,18,19,20,101,102,103,104,105,106,107),
     * STAGE
12     IF(CHARAC.EQ.MINUS)NSIGN=1
       GO TO 11
15     IF(CHARAC.LT.27.OR.CHARAC.GT.36)GO TO 25
24     I=N+N
       I=I+I+N
       I=I+I+(CHARAC-27)
       IF(I.GT.TOOBIG)GO TO 25
26     N=I
       GO TO 11
25     DIGITS=DIGITS+1
       GO TO 11
16     DIGITS=DIGITS-1
C      DECIMAL POINT ENCOUNTERED. SET FOR FLOATING POINT.
       FLTING=1.
       GO TO 15
C      E ENCOUNTERED. SET FOR FLOATING POINT.
17     FLTING=1.
       GO TO 11
18     IF(CHARAC.EQ.MINUS)ESIGN=1
       GO TO 11
19     EXPON=EXPON*10+(CHARAC-27)
       GO TO 11
```

```
20    I=WORD.AND.BITS
      IF(I)11,28,11
28    WORD=64*WORD
      WORD=WORD.OR.CHARAC
      GO TO 11
C     SCAN COMPLETE.PREPARE FOR EXIT.
101   I=COLCT-1
      IF(PRINX.EQ.0)PRINT 502,I,CHARAC,CARDCT
      GO TO 1
502   FORMAT(19H INPUT ERROR COLUMN,I3,11H CHARACTER ,O2,6H, CARD,I4)
103   COLCT=COLCT-1
      CHARAC=OLDCH
102   ISIT=CHARAC
      TYPX=0
114   RETURN
105   COLCT=COLCT-1
104   DIGITS=DIGITS+(1-FSIGN-ESIGN)*EXPON
      TYPX=1
      IF(NSIGN.NE.0)N=-N
108   IF(FLTING)124,109,124
109   ISIT=N
      IF(DIGITS)126,114,112
112   DO 120 I=1,DIGITS
      N=ISIT+ISIT
      N=N+N+ISIT
120   ISIT=N+N
      IF(ISIT.LT.TOOBIG)GO TO 114
      IF(PRINX.EQ.0)PRINT 503
503   FORMAT(18H INTEGER TOO LARGE)
      ISIT=ALLBIT
      GO TO 114
126   DIGITS=-DIGITS
      DO 125 I=1,DIGITS
125   ISIT=ISIT/10
      GO TO 114
124   X=N
      K=DIGITS
      IF(DIGITS.LT.0)K=-DIGITS
      J1=K/100
      K=K-100*J1
      J2=K/10
      K=K-J2*10
      Y=XPA(J1+1)*XPB(J2+1)*XPC(K+1)
      IF(DIGITS)115,116,117
115   X=X/Y
116   ISIT=I
      GO TO 114
117   X=X*Y
      GO TO 116
107   COLCT=COLCT-1
106   TYPX=-1
      ISIT=WORD
      GO TO 114
C     DIGIT=NUMBER OF DIGITS TO RIGHT OF THE DECIMAL POINT.
C     N=INTEGER VALUE.
C     EXPON =EXPONENT VALUE.
      END
```

Sample Sextupole Magnet
See Fig. 2

```
DATA CARDS FOR COIL BDRY    1 ARE LISTED BELOW

2  6.  .8726646  6.  .1745329  30

1  8.  .1745329  10

2  8.  .8726646  30

1  6.  .8726646  10

CURDEN =     1.023231E+02 A/CM2

DATA CARDS READ TO SET UP GRID ARE LISTED BELOW

8  0.  1.

3  0.  1.
```

MULTIPOLE ANALYSIS

THE NORMALIZING RADIUS IS 6.0000CM

BSTAR = SUM OVER N (CN.((Z0/R)..(N-1)))

N = 3	CN =	8.175091E-06	-1.064536E+02
N = 9	CN =	4.784895E-12	7.582603E-08
N = 15	CN =	-2.960939E-06	7.711295E+00
N = 21	CN =	-2.712846E-06	5.046552E+00
N = 27	CN =	5.418791E-13	-5.314642E-08
N = 33	CN =	3.503158E-06	-4.147010E+00
N = 39	CN =	4.286323E-06	-4.293479E+00
N = 45	CN =	-5.827757E-13	9.476247E-08
N = 51	CN =	-6.424398E-06	4.920979E+00
N = 57	CN =	-7.745811E-06	5.308615E+00

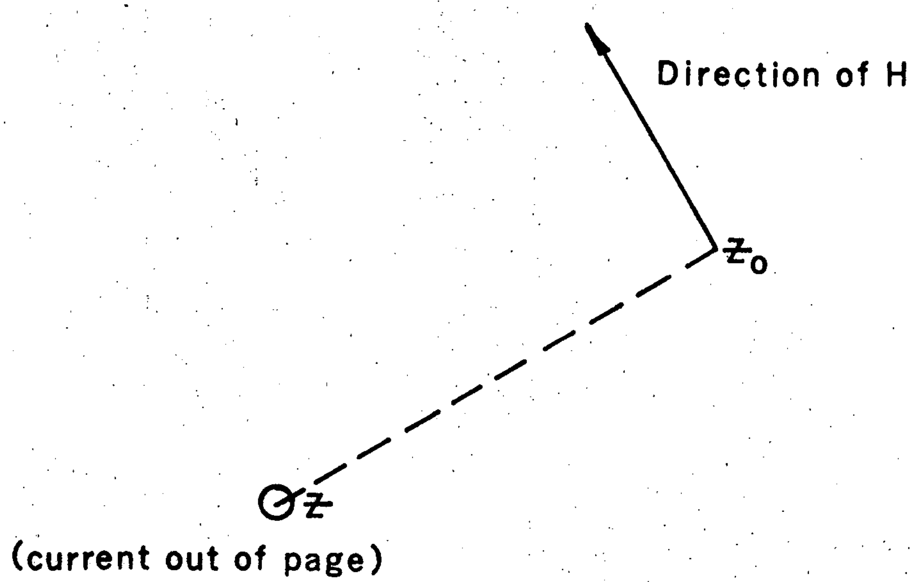
Note: The sextupole geometry described in this example (namely with each coil area equal to twice the area between successive coils) should contain no field contribution for N = 9, 27, 45, --- etc. Also, all components should be purely imaginary numbers. These features may be noted in the above calculated results.

X(CM)	Y(CM)	BX(G)	BY(G)	BT(G)
0.	0.	-3.231719E-09	1.136868E-13	3.231719E-09
1.0000	0.	2.956433E+00	1.058579E-03	2.956433E+00
2.0000	0.	1.182577E+01	4.167218E-03	1.182577E+01
3.0000	0.	2.660847E+01	7.712695E-03	2.660847E+01
4.0000	0.	4.728758E+01	-4.854259E-03	4.728758E+01
5.0000	0.	7.330725E+01	-1.693231E-01	7.330745E+01
6.0000	0.	9.779448E+01	-3.102370E-01	9.779497E+01
7.0000	0.	1.003171E+02	4.013926E-01	1.003179E+02
0.	1.0000	-2.957656E+00	1.059564E-03	2.957656E+00
1.0000	1.0000	-1.219125E-03	-5.911968E+00	5.911968E+00
2.0000	1.0000	8.867920E+00	-1.182276E+01	1.477896E+01
3.0000	1.0000	2.364744E+01	-1.773296E+01	2.955773E+01
4.0000	1.0000	4.435790E+01	-2.366153E+01	5.027416E+01
5.0000	1.0000	7.158070E+01	-2.944211E+01	7.739919E+01
6.0000	1.0000	1.162422E+02	-2.408271E+01	1.187107E+02
7.0000	1.0000	1.145786E+02	1.647872E+01	1.157575E+02
0.	2.0000	-1.183066E+01	4.304832E-03	1.183066E+01
1.0000	2.0000	-8.874152E+00	-1.182307E+01	1.478295E+01
2.0000	2.0000	-3.825049E-03	-2.364743E+01	2.364743E+01
3.0000	2.0000	1.477795E+01	-3.545759E+01	3.841391E+01
4.0000	2.0000	3.534466E+01	-4.724030E+01	5.899907E+01
5.0000	2.0000	6.107473E+01	-5.983869E+01	8.550317E+01
6.0000	2.0000	7.690362E+01	-4.029403E+01	8.682036E+01
7.0000	2.0000	6.581025E+01	3.264225E+01	7.346091E+01

*This work was done under the auspices of the U. S. Atomic Energy Commission.

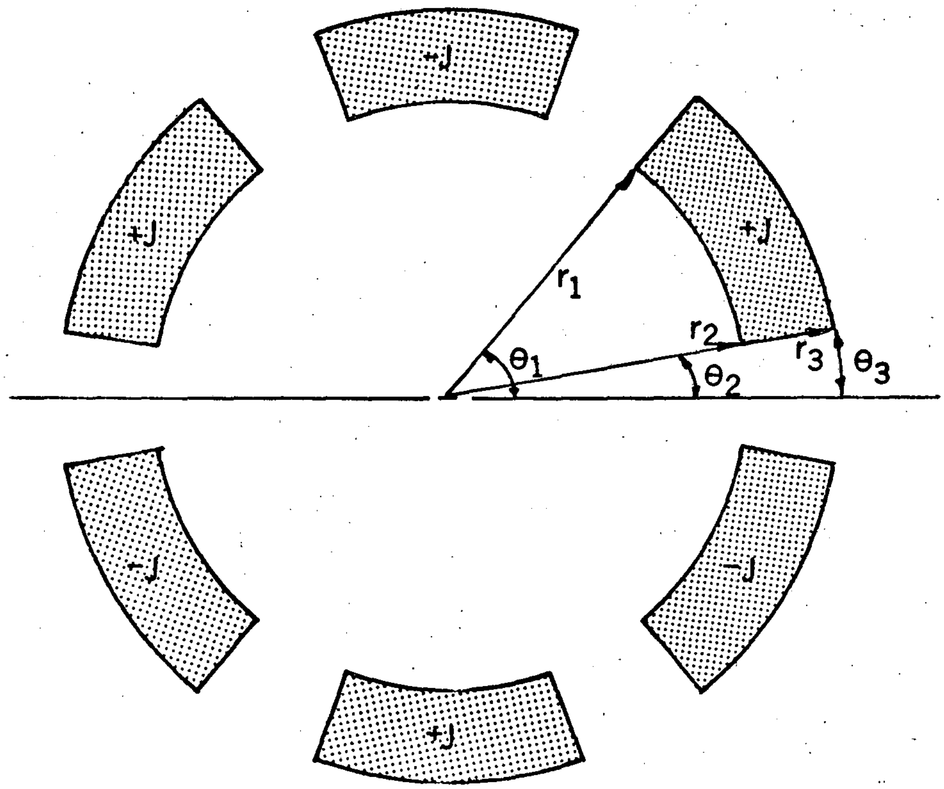
1. R. A. Beth, "Some Extensions of Complex Methods For Two-Dimensional Fields", Proceedings of the Sixth International Conference on High Energy Accelerators, Cambridge Electron Accelerator, 1967.

2. W. Benson, "Subroutine DATA - A Fortran 66 Format Free Input Routine", I3BKYDATA, June 9, 1965, Lawrence Radiation Laboratory, Berkeley, Calif.



Infinite straight current filament
Fig. 1

XBL 688 4902



Model sextupole Geometry
Fig. 2

XBL 688 4903

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.