Brookhaven National Laboratory

BNL-104815-2014-TECH

AGS/AD/Tech Note No. 399;BNL-104815-2014-IR

FIELD OF A HELIX

A. Luccio

August 1994

Collider Accelerator Department

Brookhaven National Laboratory

U.S. Department of Energy

USDOE Office of Science (SC)

DISCLAIMER

Accelerator Division
Alternating Gradient Synchrotron Department
BROOKHAVEN NATIONAL LABORATORY
Upton, New York 11973

Accelerator Division
Technical Note

AGS/AD/Tech. Note No. 399

FIELD OF A HELIX

A. Luccio

August 15, 1994

Field of a Helix

A.Luccio

Siberian Snakes

It has been proposed (Shatunov et Al.) to build Siberian snakes snd spin rotators for RHIC with an arrangement of superconducting helical dipoles. In a helical, or twisted, dipole the magnetic field on axis is perpendicular to the axis and rotates around it as we proceed along the dipole. No such magnet has been built until now. An engineering study is in progress in the RHIC department (E.Willen et Al.) to build a 4 Tesla prototype with a 10 cm bore, with an a coil geometry derived, by twisting, from the geometry of the cosine superconducting dipoles for RHIC (Blewett).

Only field measurements on the prototype will tell the detailed structure of the magnetic field of such an helix, its multipole content and the end (fringe) field. For the time being we have tried to find an analytical expression for such a field, to be used to calculate the spin precession in the helix, the particle trajectories, and the optical characteristics of the magnet considered as insertions in the lattice of the accelerator. In particular, the theoretical field can enable one to calculate the integrated multipoles through the magnet.

An expression for the field in the helix body, i.e. excluding the fringe field, can be found under the rather general conditions that the variables are separable. This expression exactly satisfies Maxwell's div and curl equations for a stationary field

(1) $$\nabla \cdot \mathbf{B} = \mathbf{0}, \quad \nabla \times \mathbf{B} = \mathbf{0}.$$

In cartesian coordinates, x and y transverse, and z longitudinal, the field of an helix of constant pitch is on axis ($x = y = 0$)

(2) $$\begin{cases} B_x = B_0 \cos kz \\ B_y = B_0 \sin kz \,, \\ B_z = 0 \end{cases}$$

with

(3) $$k = \frac{2\pi}{\lambda}.$$

The field of Eq. (2) obviously satisfies (1).

Superposition of Wigglers

Eq. (2) can be thought as the superposition of two transverse wigglers rotated by 90° around the longitudinal axis z and out of phase by $\pi/2$. A general expression for a field of such a wiggler can be found assuming that the variables are separable (K. Halbach). Consider the following expression for the vertical (y) field

(4) $$B_y(x,y,z) = B_0 X(x) Y(y) \sin kz,$$

and apply Maxwell' equations (1). From curl = 0 obtain

(5) $$\begin{cases} \frac{\partial B_x}{\partial y} = \frac{\partial B_y}{\partial x} = \frac{\partial X}{\partial x} Y \sin kz \\ \frac{\partial B_z}{\partial y} = \frac{\partial B_y}{\partial z} = kXY \cos kz \end{cases},$$

or

(6) $$\begin{cases} B_x = \frac{\partial X}{\partial x}\left(\int Y dy\right) \sin kz \\ B_z = kX\left(\int Y dy\right) \cos kz \end{cases}.$$

From div = 0 obtain

(7) $$\frac{\partial^2 X}{\partial x^2}\left(\int Y dy\right) \sin kz + X \frac{\partial Y}{dy} \sin kz - k^2 X \left(\int Y dy\right) \sin kz = 0,$$

and, by differentiating with respect to y

(8) $$\frac{\partial^2 X(x)}{\partial x^2}Y(y)+X(x)\frac{\partial^2 Y(y)}{\partial y^2}-k^2X(x)Y(y)=0.$$

Eq.(8) is equivalent to the two formally identical equations

(9) $$\begin{cases}\dfrac{\partial^2 X(x)}{\partial x^2}-k_x^2X(x)=0\\[2mm] \dfrac{\partial^2 Y(y)}{\partial y^2}-k_y^2Y(y)=0\end{cases}.$$

with

(10) $$k_x^2+k_y^2=k^2.$$

The general integral of Eqs. (9) is

(11) $$\begin{cases}X(x)=a\,\mathrm{Ch}(k_x x)+b\,\mathrm{Sh}(k_x x)\\ Y(y)=c\,\mathrm{Ch}(k_y y)+d\,\mathrm{Sh}(k_y y)\end{cases}.$$

The field of this sinusoidal dipole must obey some symmetry requirements, i.e.

(12) $$\begin{bmatrix}B_x(-x)=-B_x(x) & B_x(-y)=-B_x(y)\\ B_y(-x)=B_y(x) & B_y(-y)=B_y(y)\end{bmatrix}.$$

Finally, the field of a single (vertical) sinusoidal wiggler is obtained

(13) $$\begin{cases}B_x=B_0\dfrac{k_x}{k_y}S_xS_y\sin kz\\[2mm] B_y=B_0C_xC_y\sin kz\\[2mm] B_z=B_0\dfrac{k}{k_y}C_xS_y\cos kz\end{cases},$$

with

(14) $$\begin{bmatrix} S_x = \mathrm{Sh}k_x x & C_x = \mathrm{Ch}k_x x \\ S_y = \mathrm{Sh}k_y y & C_y = \mathrm{Ch}k_y y \end{bmatrix}.$$

It is easy to check that the field of Eq. (13) obeys Eqs. (1).

A field of two mutually perpendicular sinusoidal wigglers, out of phase by 90° that reduces to Eq. (2) on axis, is immediately found from Eq. (13). It is

(15) $$\begin{cases} B_x = B_0\left[C_x C_y \cos kz + \dfrac{k_x}{k_y} S_x S_y \sin kz\right] \\ B_y = B_0\left[C_x C_y \sin kz + \dfrac{k_y}{k_x} S_x S_y \cos kz\right] \\ B_z = B_0\left[\dfrac{k}{k_y} C_x S_y \cos kz - \dfrac{k}{k_x} S_x C_y \sin kz\right] \end{cases}.$$

However, the field (15) is not the field that we could expect from a twisted dipole. A more general helical dipole field is obtained by the superposition of many infinitesimal wiggler fields, continuously rotated by an angle θ around the axis z (figure 1)

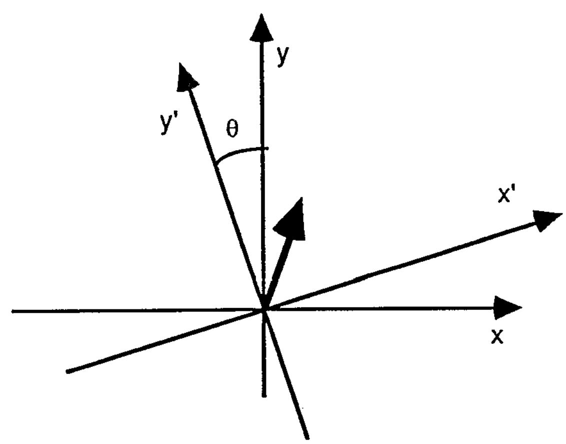

Fig. 1

Using the transformation between rotated frame (primed quantities) and laboratory (unprimed)

(16)
$$\begin{cases} B_x = B_{x'}\cos\theta - B_{y'}\sin\theta \\ B_y = B_{x'}\sin\theta + B_{y'}\cos\theta \end{cases} \qquad \begin{cases} x' = x\cos\theta + y\sin\theta \\ y' = -x\sin\theta + y\cos\theta \end{cases}$$

the components of the field obtained by the integral sum of many infinitesimal transverse sinusoidal wigglers, that exactly satisfy Maxwell's, are

(17)
$$\begin{cases} b_x = \dfrac{B_x}{B_0} = b_1\sin kz + b_2\cos kz \\ b_y = \dfrac{B_y}{B_0} = b_3\sin kz + b_4\cos kz, \\ b_z = \dfrac{B_z}{B_0} = b_5\sin kz + b_6\cos kz \end{cases}$$

$$b_i = \frac{1}{\sqrt{2}}\left(-a_1 + \frac{k_x}{k_y}a_2,\ \ a_3 - \frac{k_y}{k_x}a_4,\ \ a_3 + \frac{k_x}{k_y}a_4,\ \ a_1 + \frac{k_y}{k_x}a_2,\ \ -\frac{k}{k_x}a_5,\ \ \frac{k}{k_y}a_6,\right),$$

(18)
$$a_i = \begin{pmatrix} \int \mathrm{Ch}(u')\mathrm{Ch}(v')\sin\theta\, d\theta \\ \int \mathrm{Sh}(u')\mathrm{Sh}(v')\cos\theta\, d\theta \\ \int \mathrm{Ch}(u')\mathrm{Ch}(v')\cos\theta\, d\theta \\ \int \mathrm{Sh}(u')\mathrm{Sh}(v')\sin\theta\, d\theta \\ \int \mathrm{Sh}(u')\mathrm{Ch}(v')\, d\theta \\ \int \mathrm{Ch}(u')\mathrm{Sh}(v')\, d\theta \end{pmatrix}.$$

(19)
$$\begin{cases} u' = uc - vs \\ v' = us + vc \end{cases} \quad \begin{cases} u = k_x x \\ v = k_y y \end{cases}. \qquad s = \sin\theta \qquad c = \cos\theta$$

An expansion to quadratic terms in x and y, close to the z axis, in the assumption

(19a) $\theta_1 = \pi/4, \quad \theta_2 = 3\pi/4,$ (field vertical at the helix entrance) ,

is

(17a)
$$\begin{cases} b_x \approx \left[-1 - \frac{1}{3}u^2 + \frac{2}{3}v^2\right]\sin kz - \frac{1}{6}uv\cos kz \\ b_y \approx \frac{1}{6}uv\sin kz + \left[1 + \frac{2}{3}u^2 + \frac{1}{3}v^2\right]\cos kz \\ b_z \approx 2v\sin kz + 2u\cos kz \end{cases}.$$

On axis (x=y= 0), the field of Eq. (17) can be written as

(20)
$$\begin{cases} B_x = B_0 \sin(kz+\phi) \\ B_y = -B_0 \cos(kz+\phi), \\ B_z = 0 \end{cases}$$

with the phase

(21)
$$\tan\phi = \frac{\sin\theta_2 - \sin\theta_1}{\cos\theta_2 - \cos\theta_1},$$

where θ_1 and θ_2 are the rotation angle of the first and last infinitesimal wiggler, respectively (limits of the integrals over θ).

Explicit expressions for the products of hyperbolic functions in the a coeeficients are found with (Abramowitz-Stegun #9.6.34 and 9.6.35, p.376)

(22)
$$\begin{cases} e^{z\cos\theta} = I_0(z) + 2\sum_{k=1}^{\infty} I_k(z)\cos k\theta \\ e^{z\sin\theta} = I_0(z) + 2\sum_{k=0}^{\infty}(-1)^k I_{2k+1}(z)\sin(2k+1)\theta + 2\sum_{k=1}^{\infty}(-1)^k I_{2k}(z)\cos 2k\theta \end{cases},$$

where $I_n(z)$ are modified Bessel functions that satisfy the following symmetry conditions

(23)
$$\begin{array}{ll} I_{-k}(z) = I_k(z) & \text{k = integer} \\ I_k(-z) = I_k(z) & \text{if } k = \text{even}. \\ I_k(-z) = -I_k(z) & \text{if } k = \text{odd} \end{array}$$

From Eqs. (22) we obtain

(24)
$$\begin{cases} \mathrm{Sh}(z\cos\theta) = 2\sum_{k=0}^{\infty} I_{2k+1}(z)\cos(2k+1)\theta \\ \mathrm{Sh}(z\sin\theta) = 2\sum_{k=0}^{\infty}(-1)^k I_{2k+1}(z)\sin(2k+1)\theta \\ \mathrm{Ch}(z\cos\theta) = I_0(z) + 2\sum_{k=1}^{\infty} I_{2k}(z)\cos 2k\theta \\ \mathrm{Ch}(z\sin\theta) = I_0(z) + 2\sum_{k=1}^{\infty}(-1)^k I_{2k}(z)\cos 2k\theta \end{cases},$$

and then

$$(25)\quad \begin{cases} Sh(u') = Sh(uc - vs) = Sh(uc)Ch(vs) - Ch(uc)Sh(vs) \\ Sh(v') = Sh(us + vc) = Sh(us)Ch(vc) + Ch(us)Sh(vc) \\ Ch(u') = Ch(uc - vs) = Ch(uc)Ch(vs) - Sh(uc)Sh(vs) \\ Ch(v') = Ch(us + vc) = Ch(us)Ch(vc) + Sh(us)Sh(vc) \end{cases},$$

$$(26)\quad \begin{cases} Sh(uc)Ch(vs) = 2\sum_{m=1}^{\infty}(-1)^m \sum_{k=-\infty}^{\infty}(-1)^k I_{2k-1}(u)I_{2k-2m}(v)\cos(2m+1)\theta \\ Ch(uc)Sh(vs) = 2\sum_{m=1}^{\infty}(-1)^m \sum_{k=-\infty}^{\infty}(-1)^k I_{2k}(u)I_{2k+2m+1}(v)\sin(2m+1)\theta \\ Ch(uc)Ch(vs) = \frac{1}{2}\left[I_0(u)I_0(v) - 2\sum_{k=1}^{\infty}(-1)^k I_{2k}(u)I_{2k}(v)\right] \\ \qquad +2\sum_{m=1}^{\infty}(-1)^m \sum_{k=-\infty}^{\infty}(-1)^k I_{2k}(u)I_{2k+2m}(v)\cos 2m\theta \\ Sh(uc)Sh(vs) = \sum_{m=1}^{\infty}(-1)^m \sum_{k=-\infty}^{\infty}(-1)^k I_{2k+1}(u)I_{2k-2m+1}(v)\sin 2m\theta \end{cases}.$$

Using the expressions above, the field of an helix can be expressed in a trigonometric series with coefficients containing the Bessel functions.

Expansion. Integrated multipoles.

Let us write an expansion of Eqs. (17) in u, v useful to evaluate analytically the multipole content of the field. For the integrals in Eqs. (18) it is to second order

$$(27)\quad \begin{cases} a_1^{(2)} = \left[1 + \frac{1}{2}(u^2 + v^2)\right]I_{10} \\ a_2^{(2)} = (u^2 - v^2)I_{12} + uv(I_{03} - I_{21}) \\ a_3^{(2)} = \left[1 + \frac{1}{2}(u^2 + v^2)\right]I_{01} \\ a_4^{(2)} = (u^2 - v^2)I_{21} + uv(I_{12} - I_{30}) \\ a_5^{(2)} = uI_{01} - vI_{10} \\ a_6^{(2)} = uI_{10} + vI_{01} \end{cases},$$

Terms of higher order (up to fourth) are

$$(28)\qquad \begin{cases} a_1^{(4)} = a_{11}(u^2+v^2)^2 + a_{12}(u^2-v^2)^2 + a_{13}u^2v^2 + a_{14}uv(u^2-v^2) \\ a_2^{(4)} = a_{21}(u^4-v^4) + a_{22}uv(u^2+v^2) \\ a_3^{(4)} = a_{31}(u^2+v^2)^2 + a_{32}(u^2-v^2)^2 + a_{33}u^2v^2 + a_{34}uv(u^2-v^2) \\ a_4^{(4)} = a_{41}(u^4-v^4) + a_{42}uv(u^2+v^2) \\ a_5^{(4)} = a_{51}u^3 + a_{52}u^2v + a_{53}uv^2 + a_{54}v^3 \\ a_6^{(4)} = a_{61}u^3 + a_{62}u^2v + a_{63}uv^2 + a_{64}v^3 \end{cases},$$

with the integrals

$$(29)\qquad a_{ij} = \begin{bmatrix} \frac{1}{24}I_{10} & \frac{1}{4}I_{32} & \frac{1}{4}(I_{14}-2I_{32}+I_{50}) & \frac{1}{2}(I_{23}-I_{41}) \\ \frac{1}{3}I_{12} & \frac{1}{3}(I_{03}-I_{21}) & & \\ \frac{1}{24}I_{01} & \frac{1}{4}I_{23} & \frac{1}{4}(I_{05}-2I_{23}+I_{41}) & \frac{1}{2}(I_{14}-I_{32}) \\ \frac{1}{3}I_{21} & \frac{1}{3}(I_{12}-I_{30}) & & \\ \frac{1}{3}I_{03}+\frac{1}{2}I_{21} & -\frac{1}{2}I_{30} & \frac{1}{2}I_{03} & -(\frac{1}{3}I_{30}+\frac{1}{2}I_{12}) \\ \frac{1}{3}I_{30}+\frac{1}{2}I_{12} & \frac{1}{2}I_{03} & \frac{1}{2}I_{30} & (\frac{1}{3}I_{03}+\frac{1}{2}I_{21}) \end{bmatrix}$$

$$(30)\qquad I_{mn} = \int_{\Delta\theta} \sin^m\theta \cos^n\theta \, d\theta$$

Explicit expressions for the (indefinite) integrals are

$$(31)\qquad \begin{bmatrix} I_{00}=\theta & I_{30}=-c+\frac{1}{3}c^3 & I_{40}=\frac{3}{8}\theta-\frac{3}{8}sc-\frac{1}{4}s^3c & I_{50}=-\frac{4}{5}c+\frac{4}{15}c^3-\frac{1}{5}s^4c \\ I_{10}=-c & I_{21}=\frac{1}{3}s^3 & I_{31}=\frac{1}{4}s^4 & I_{41}=\frac{1}{5}s^5 \\ I_{01}=s & I_{12}=-\frac{1}{3}c^3 & I_{22}=\frac{1}{8}\theta-\frac{1}{8}sc-\frac{1}{4}s^4 & I_{32}=-\frac{1}{3}c^3+\frac{1}{5}c^5 \\ I_{20}=\frac{1}{2}\theta-\frac{1}{2}sc & I_{03}=s-\frac{1}{3}s^3 & I_{13}=-\frac{1}{4}c^4 & I_{23}=\frac{1}{3}s^3-\frac{1}{5}s^5 \\ I_{11}=\frac{1}{4}-\frac{1}{2}c^2 & & I_{04}=\frac{3}{8}\theta+\frac{3}{8}sc+\frac{1}{4}sc^3 & I_{14}=-\frac{1}{5}c^5 \\ I_{02}=\frac{1}{2}\theta+\frac{1}{2}sc & & & I_{05}=\frac{4}{5}s-\frac{4}{15}s^3+\frac{1}{5}sc^4 \end{bmatrix}$$

There are no quadrupole nor octupolar terms in the transverse field expansion. The expansion of the field near the axis can be used to find the integrated multipoles along the particle trajectory.

Exact equations for the trajectory are

(32) $$\begin{cases} \frac{dx'}{dz} = \Omega_x x' y' - \Omega_y (1 + x'^2) + \Omega_z y' \\ \frac{dy'}{dz} = \Omega_x (1 + y'^2) - \Omega_y x' y' - \Omega_z x' \\ \frac{dx}{dz} = x' \\ \frac{dy}{dz} = y' \end{cases},$$

with the normalized magnetic field

(33) $$\Omega = \frac{e}{mc\gamma}\mathbf{B}$$

Let us calculate the trajectory to zeroth order, using the field on axis

(34) $$\begin{cases} \Omega_x^{(0)} = \Omega_0 \sin(kz + \phi_0) \\ \Omega_y^{(0)} = -\Omega_0 \cos(kz + \phi_0), \\ \Omega_z^{(0)} = 0 \end{cases}$$

(35) $$\tan\phi_0 = -\frac{I_{01}}{I_{10}}.$$

The para-axial trajectory is

(36) $$\begin{cases} x = x_0 + \left(x'_0 - \frac{\Omega_0}{k}\sin\phi_0\right)z - \frac{\Omega_0}{k^2}[\cos(kz + \phi_0) - \cos\phi_0] \\ y = y_0 + \left(y'_0 + \frac{\Omega_0}{k}\cos\phi_0\right)z - \frac{\Omega_0}{k^2}[\sin(kz + \phi_0) - \sin\phi_0] \end{cases}.$$

Consider for simplicity the case

(37) $$k_x = k_y = \frac{k}{\sqrt{2}},$$

and calculate the integrated sextupole along the para-axial trajectory. The expansion for the field of Eqs. (17) is

(38)
$$\underset{i=1,4}{b_i^{(2)}} = \begin{bmatrix} -I_{10}\left[1+Au^2+2Buv+Cv^2\right] \\ I_{01}\left[1+Du^2+2Euv+Fv^2\right] \\ I_{01}\left[1+Fu^2+2Euv+Dv^2\right] \\ I_{10}\left[1+Cu^2+2Buv+Av^2\right] \end{bmatrix},$$

with

(39)
$$\begin{bmatrix} A=\frac{1}{2}-\frac{I_{12}}{I_{10}} & B=\frac{1}{2}\frac{I_{03}-I_{21}}{I_{10}} & C=\frac{1}{2}+\frac{I_{12}}{I_{10}} \\ D=\frac{1}{2}-\frac{I_{21}}{I_{01}} & E=-\frac{1}{2}\frac{I_{12}-I_{30}}{I_{01}} & F=\frac{1}{2}+\frac{I_{21}}{I_{01}} \end{bmatrix}.$$

Without lack of generality, consider the case

(40)
$$\theta_1 = 0, \quad \theta_2 = \frac{\pi}{2},$$

where it is

(41)
$$A = B = D = -E = \frac{1}{6}, \quad C = F = \frac{5}{6}$$

Now, insert the para-axial trajectory of Eq. (36) in the expansion for the field components of Eqs. (39), in the case of Eq. (41), and integrate along z. The integrals will be a function of the particle initial position and angle, contained in (37). In the simplifying assumptions

(42)
$$\begin{aligned} &kL = 2\pi \\ &x'_0 = \frac{\Omega_0}{k}\sin\phi_0 = -\frac{\Omega_0}{\sqrt{2}k}, \\ &y'_0 = -\frac{\Omega_0}{k}\cos\phi_0 = \frac{\Omega_0}{\sqrt{2}k} \end{aligned}$$

with L the length of the helix, we obtain for the integrated transverse field along the trajectory, to second order in x_o, y_o

(43) $$\begin{cases} \int b_x dz = -\frac{1}{48\pi^2}(\Omega_0 L)^2 L - \frac{1}{12}\Omega_0 L x_0 \\ \int b_y dz = -\frac{1}{48\pi^2}(\Omega_0 L)^2 L + \frac{1}{12}\Omega_0 L y_0 \end{cases} .$$

Eq. (43) shows that, in the assumptions (42), the quadratic terms, in $x_0^2, x_0 y_0, y_0^2$, vanish.

A numerical example is

$$\gamma = 25, \quad B = 4\,T, \quad L = 2\,m$$
$$\Omega_0 L = 0.0978, \quad \tfrac{1}{12}\Omega_0 L = 8.15\,10^{-3}, \quad \tfrac{1}{12\pi^2}(\Omega_0 L)^2 L = 8.08\,10^{-5}\ m$$

Numerical integration.

The field of the helix can be calculated according to the expressions (17) and mapped. Trajectories can be calculated by integration of the equations of motion (33) through the field map. Two computer codes have been written: HEL, to calculate the field map, and TRACK, to calculate trajectories for different initial conditions and evaluate the integrated field along the trajectory. Both codes extensively use routines from "Numerical Recipes".

HEL calculates the *a* integrals and the *b* coefficients of Eqs. (18) using a Romberg integration. HEL also evaluates the coefficients of the expansion of the *b*'s by a linear least-squares fitting. An example of coefficients up to 4-th order, for a 2 meter long helix in the assumption $kL = 2\pi$ is given in Table 1. The expansion has the form

(44) $$b_i = \sum_{j,k=0}^{N} b_{ijk} x^j y^k .$$

The table shows a substantial sextupole and decapole contribution to the transverse field plus *x*-*y* coupling, and no quadrupole or octupole.

The program TRACK calculates the trajectories of a distribution of particles in phase space, using a Predictor Corrector plus Runge-Kutta integration routine. The local field is foun bi a Bi-linear interpolation in the binary field map, created by HEL.

The actual listing of HEL and TRACK and their input files HEL.DAT and TRACK.DAT are given in the Appendix.

Table 1. Example of expansion coefficient of the field of an helix.

b100= -7.07107E-01	b101= 4.75250E-14	b102= -5.81572E-01	b103= -2.50442E-11	b104= -1.43572E-01
b110= 1.07540E-14	b111= 1.16314E+00	b112= 6.60923E-12	b113= 1.34073E+00	
b120= -2.90786E+00	b121= -4.04935E-11	b122= -2.00897E+00		
b130= -5.56316E-12	b131= 5.74254E-01			
b140= -2.05897E+00				
b200= 7.07107E-01	b201= -6.65913E-16	b202= 5.81572E-01	b203= -1.79913E-12	b204= 1.43572E-01
b210= -1.25140E-14	b211= 1.16314E+00	b212= -7.69554E-12	b213= 1.34073E+00	
b220= 2.90786E+00	b221= -1.28346E-11	b222= 2.00897E+00		
b230= 6.54549E-12	b231= 5.74254E-01			
b240= 2.05897E+00				
b300= 7.07107E-01	b301= 1.44189E-14	b302= 2.90786E+00	b303= -7.27494E-12	b304= 2.05897E+00
b310= -1.17533E-14	b311= -1.16314E+00	b312= 1.78786E-12	b313= -5.74254E-01	
b320= 5.81572E-01	b321= -4.19058E-11	b322= 2.00897E+00		
b330= 5.72632E-12	b331= -1.34073E+00			
b340= 1.43572E-01				
b400= 7.07107E-01	b401= -2.48302E-14	b402= 2.90786E+00	b403= 1.23792E-11	b404= 2.05897E+00
b410= -8.70594E-15	b411= 1.16314E+00	b412= 2.59776E-12	b413= 5.74254E-01	
b420= 5.81572E-01	b421= 4.08394E-11	b422= 2.00897E+00		
b430= 4.38018E-12	b431= 1.34073E+00			
b440= 1.43572E-01				
b500= 1.29068E-17	b501= -2.22144E+00	b502= -1.08931E-14	b503= -3.04900E+00	b504= -4.11158E-13
b510= -2.22144E+00	b511= 9.43231E-14	b512= -1.82706E+00	b513= -3.80141E-11	
b520= 1.11649E-13	b521= -1.82706E+00	b522= -2.78466E-10		
b530= -3.04900E+00	b531= -2.37814E-11			
b540= -4.63317E-11				
b600= 1.20686E-17	b601= 2.22144E+00	b602= -1.61545E-13	b603= 3.04900E+00	b604= 7.83806E-11
b610= -2.22144E+00	b611= -1.18687E-14	b612= -1.82706E+00	b613= -6.93113E-12	
b620= 1.37343E-13	b621= 1.82706E+00	b622= -1.23090E-10		
b630= -3.04900E+00	b631= 1.69625E-12			
b640= -5.51318E-11				

Appendix 1. Fortran code HEL.FTN

```
*******************************************************************
* A.Luccio, BNL
*******************************************************************
c Field of an helix. Multipoles
c Romberg integration
c Linear least squares fit
c
      implicit real*8 (a-h,o-z)
      parameter (pi=3.14159265358979324d0,nv=201,ma=5)
      dimension a(6),b(6)
      dimension vx(nv),vy(nv),vf(nv),vvf(6,nv),cof(ma),ccof(6,ma,nv)
      character xoy,ab
      character*5 ach,bch
      real lambda,kk
      external func
      data ach,bch/'    a','    b'/
      common/a/u,v,k
      common/map/x1,dx,nx,y1,dy,ny,ab
      common/snake/lambda,th1,th2

      call iopen(1)

        sq2     = dsqrt(2.d0)
        kk      = 2*pi/lambda

c write field binary file
        write(21) nx
          x = x1
      do i=1,nx
        write(21) x
          x = x +dx
      enddo
        write(21) ny
          y = y1
      do j=1,ny
        write(21) y
          y = y +dy
      enddo
          y = y1
      do j=1,ny

c Double Loop
          x = x1
      do i=1,nx
          u = kk/sq2*x
          v = kk/sq2*y
        do k=1,6
          call qromb(func,th1,th2,ss)
          a(k) = ss
        enddo
            b(1) = (-a(1)+a(2))/sq2
            b(2) = (a(3)-a(4))/sq2
            b(3) = (a(3)+a(4))/sq2
            b(4) = (a(1)+a(2))/sq2
            b(5) = -a(5)
            b(6) = a(6)
            write(21) b
            write(20,'(f8.4,1p6e14.6)') x,a
            write(22,'(f8.4,1p6e14.6)') x,b
                             vx(i) = x
          do k=1,6
            if (ab.eq.'a') vvf(k,i) = a(k)
            if (ab.eq.'b') vvf(k,i) = b(k)
          enddo
          x = x +dx
        enddo
          do k=1,6
            do l=1,nx
              vf(l) = vvf(k,l)
            enddo
```

```
          call expa(nx,vx,vf,cof)
          do m=1,ma
            ccof(k,m,j) = cof(m)
          enddo
        enddo
      vy(j) = y
      y = y +dy
    enddo

c Linear double fit
c Write coefficients
    do k=1,6
        print *,' '
      do m=1,ma
        do j=1,ny
          vf(j) = ccof(k,m,j)
        enddo
          call expa(ny,vy,vf,cof)
          if (ab.eq.'a') write(*,1000) (ach,k,m-1,l-1,cof(l),l=1,6-m)
          if (ab.eq.'b') write(*,1000) (bch,k,m-1,l-1,cof(l),l=1,6-m)
      enddo
    enddo

 1000 format(5(a5,3i1,'=',1pe13.5))
      call iopen(2)
                                   end
c---------------------------------------------------------------
      function func(z)
      implicit real*8 (a-h,o-z)
      common/a/ u,v,k
        up = u*dcos(z)+v*dsin(z)
        chup = dcosh(up)
        shup = dsinh(up)
        vp = -u*dsin(z)+v*dcos(z)
        chvp = dcosh(vp)
        shvp = dsinh(vp)
      goto (1,2,3,4,5,6),k
 1      func=chup*chvp*dsin(z)
      return
 2      func=shup*shvp*dcos(z)
      return
 3      func=chup*chvp*dcos(z)
      return
 4      func=shup*shvp*dsin(z)
      return
 5      func=shup*chvp
      return
 6      func=chup*shvp
      end
c---------------------------------------------------------------
      subroutine iopen(i)
      parameter (pi=3.14159265358979324d0)
      implicit real*8 (a-h,o-z)
      real lambda
      character*12 infile(2),binfile
      character ab
      namelist/map/x1,x2,dx,y1,y2,dy,ab,infile,binfile
      common/map/x1,dx,nx,y1,dy,ny,ab
      namelist/snake/lambda,th1fr,th2fr
      common/snake/lambda,th1,th2

      goto (100,200),i

 100  open(10,file='hel.dat')
      read(10,snake)
      read(10,map)
      open(20,file=infile(1))
      open(22,file=infile(2))
      open(21,file=binfile,form='unformatted')
        nx  = int(dabs(x2-x1)/dx)+1
```

```
          x1  = x1*1.d-3
          dx  = dx*1.d-3
          ny  = int(dabs(y2-y1)/dy)+1
          y1  = y1*1.d-3
          dy  = dy*1.d-3
          th1 = pi*th1fr
          th2 = pi*th2fr
        return

 200    print *,' '
        write(*,*) infile,binfile,' written'
                       end
c-------------------------------------------------------------
        subroutine expa(nd,va,vb,a)
        implicit real*8 (a-h,o-z)
        parameter (ma=5,nnd=201)
        dimension va(nd),vb(nd),
     +            sig(nnd),a(ma),lista(ma),covar(ma,ma),
     +            vf(ma)
        external funcs
        data sig/nnd*1.d-6/,lista/1,2,3,4,5/
c Linear fit
        mfit = ma
        ncwm = ma
        call lfit(va,vb,sig,nd,a,ma,lista,mfit,covar,ncwm,chisq,funcs)
                            end
c---------------------------------------------------
        subroutine funcs(x,afunc,ma)
        implicit real*8 (a-h,o-z)
        dimension afunc(ma)
c polynomial fit
        do j=1,ma
          afunc(j) = x**(j-1)
        enddo
                          end
c-------------------------------------------------------------
        SUBROUTINE QROMB(FUNC,A,B,SS)
        SUBROUTINE TRAPZD(FUNC,A,B,S,N)
        SUBROUTINE POLINT(XA,YA,N,X,Y,DY)
        SUBROUTINE LFIT(X,Y,SIG,NDATA,A,MA,LISTA,MFIT,COVAR,NCVM,CHISQ,
       +FUNCS)
        SUBROUTINE COVSRT(COVAR,NCVM,MA,LISTA,MFIT)
        SUBROUTINE GAUSSJ(A,N,NP,B,M,MP)
c Numerical Recipes
*****************************************************************
* Created: Friday, July 15, 1994   1:47:32 pm (EDT)
* Mod: Monday, August 15, 1994   3:43:41 pm (EDT)
```

Appendix 2. Input file HEL.DAT

```
$snake
 lambda=2.0d0
 th1fr=0.d0
 th2fr=0.5d0
$
$map
 x1=-50.d0
 x2=50.d0
 dx=5.d0
 y1=-50.d0
 y2=50.d0
 dy=5.d0
 ab='b'
 infile(1)='hel.a.map','hel.b.map'
 binfile='hel.bmap'
$
 th1fr=0.25d0
 th2fr=0.75d0
```

Appendix 3. Fortran code TRACK.FTN

```
*****************************************************************
* A.Luccio, BNL
*****************************************************************
      implicit real*8 (a-h,o-z)

      parameter (pi=3.14159265358979324,
     +           em=9.578d7,c=2.998d8,eo=0.9382796)
      parameter (nv=10,mp=256)

      real*4 ran1
      real len,lambda
      dimension v(4),dv(4),aux(16,4),prmt(5),prmto(5)
      dimension ee(6)
      character*80 ffile

      logical spline,verbose

      external  fct,outp

      namelist /field/bo,len,lambda,rth,ffile
      namelist /beam/np,ekin
      namelist /run/spline,prmto,zpr,dzpr,verbose

      data np/1/,verbose/.false./

      common /varb/zpr,dzpr,bx,by,bz,verbose
      common /coef/co,beta,bo,lambda,len,rth
      common /plotvar/np,zo,xo,xpo,alfax,yo,ypo,z,x,xp,y,yp,alfay
      common /center/xoo,xpoo,yoo,ypoo
      common /fint/zold,bxint,byint,bzint

      open (10,file='track.dat')
      read (10,beam)
      read (10,run)
      read (10,field)

      gam  = (ekin+eo)/eo
      beta = dsqrt(1.d0-1.d0/(gam*gam))
      co   = em/(c*gam)

c Open output files
      call fopen(1)
      open(30,file='track.out')

c Read field from binary file. Save values
      call bfld(1,ffile,spline,ee)

c Loop on particles
        sx   = 0
        sxp  = 0
        sxxp = 0
        sx2  = 0
        sy   = 0
        syp  = 0
        syyp = 0
        sy2  = 0
      do 400 jp = 1,np

c       initialize field integrals
          zold  = prmto(1)
          bxint = 0.d0
          byint = 0.d0
          bzint = 0.d0

c       Starting coordinates of a particle
        call popul8(np,jp,xo,xpo,yo,ypo)
        do i = 1,5
          prmt(i) = prmto(i)
        enddo
        do i = 1,4
```

```
          dv(i) = 0.25d0
        enddo

c       Restore field grid search starting point
        call bfld(3,ffile,spline,ee)

          v(1) = xo
          v(2) = xpo
          v(3) = yo
          v(4) = ypo
          zo   = prmt(1)

c       Integrate equations of motion
        call dhpcg(prmt,v,dv,4,ihlf,fct,outp,aux)

        if (ihlf.ne.0) print *,'ihlf = ',ihlf

          x  = v(1)
          xp = v(2)
          y  = v(3)
          yp = v(4)

        if (jp.ne.1) then
          sx   = sx   +x
          sxp  = sxp  +xp
          sxxp = sxxp +x*xp
          sx2  = sx2  +x*x
          sy   = sy   +y
          syp  = syp  +yp
          syyp = syyp +y*yp
          sy2  = sy2  +y*y
        endif

c       Collect values in output files
        if (np.ne.1) call fopen(2)

 400  continue
c End loop on particles

c Tilt of end ellipse
      if (np.ne.1) then
        alfax = ((np-1)*sxxp-sx*sxp)/((np-1)*sx2-sx*sx)
        alfay = ((np-1)*syyp-sy*syp)/((np-1)*sy2-sy*sy)
        write(*,'(''alfax, alfay = '',2g13.5)') alfax,alfay
      endif

c Write phase-space output files
      if (np.ne.1) call fopen(3)

                                    end
c-----------------------------------------------------------------
      subroutine fct(z,v,dv)

      implicit real*8 (a-h,o-z)
      parameter (pi=3.14159265358979324d0,
     +           gg=1.7928d0,em=9.58d7,c=2.998d8)
      dimension v(4),dv(4),ee(6)
      real lambda,len
      character*80 ffile
      logical spline,verbose
      save zold

      common /varb/zpr,dzpr,bx,by,bz,verbose
      common /coef/co,beta,bo,lambda,len,rth
      common /fint/zold,bxint,byint,bzint

        x   = v(1)
        xp  = v(2)
        y   = v(3)
        yp  = v(4)
```

Appendix 3. ...TRACK.FTN

```
c Find field in a mesh
c note: the ee coeffs are the "b" in "field of a helix"
c
      call bfld(2,ffile,spline,x,y,ee)

c rth=-1 : rigthhanded helix
         skz = rth*dsin(2*pi*z/lambda)
         ckz = dcos(2*pi*z/lambda)
         bx  = bo*(ee(1)*skz +ee(2)*ckz)
         by  = bo*(ee(3)*skz +ee(4)*ckz)
         bz  = bo*(ee(5)*skz +ee(6)*ckz)

         dz    = z -zold
         bxint = bxint +bx*dz
         byint = byint +by*dz
         bzint = bzint +bz*dz
         zold  = z

         omx   = co*bx
         omy   = co*by
         omz   = co*bz

c Diff. Equations of motion
         betaz = beta/dsqrt(1.d0+xp*xp+yp+yp)
         dv(1) = xp
         dv(2) = (xp*yp*omx -(1.d0+xp*xp)*omy +yp*omz)/betaz
         dv(3) = yp
         dv(4) = ((1.d0+yp*yp)*omx -xp*yp*omy -xp*omz)/betaz

                                   end
c-----------------------------------------------------------------
      subroutine bfld(ii,ffile,spline,x,y,ee)

      implicit real*8 (a-h,o-z)
      parameter (pi=3.14159265358979324d0)
      parameter (nnx=101,nny=101)
      dimension xt(nnx),yt(nny),et(6,nnx,nny),ee(6),dedx(6),dedy(6)
      character*80 ffile
      logical spline
      real*4  xmax,xmin,ymax,ymin
      save    nx,ny,xt,yt,et,isave,jsave
      save    b1,b2,b3,b4

      data xmax,xmin,ymax,ymin/-10.,10.,-10.,10./

      goto (100,200,300),ii

c Read binary field file
 100  open(12,file=ffile,form='unformatted')
      read(12) nx
      do i=1,nx
        read(12) xt(i)
        if (xt(i).ge.xmax) xmax = xt(i)
        if (xt(i).le.xmin) xmin = xt(i)
      enddo
        read(12) ny
      do j=1,ny
        read(12) yt(j)
        if (yt(j).ge.ymax) ymax = yt(j)
        if (yt(j).le.ymin) ymin = yt(j)
      enddo
      do j=1,ny
        do i=1,nx
          read(12) (et(k,i,j),k=1,6)
        enddo
      enddo
      isave = 1
      jsave = 1
c
```

```
      return

c      if (spline) call spline (x,y,z,bx,by,bz)

c Bilinear interpolation in (x,y)
 200  do i = isave,nx-1
        x1  = xt(i)
        x2  = xt(i+1)
        dx1 = x-x1
        dx2 = x-x2
        if (dx1*dx2.le.0) then
c            print *,x1,'<',x,'<',x2,'  dx=',dx1,dx2
          ddx = x2-x1
          ddxinv= 1/ddx
          isave = i-1
          if (isave.eq.0) isave = 1
          goto 210
        endif
      enddo
 210  do j = jsave,ny-1
        y1  = yt(j)
        y2  = yt(j+1)
        dy1 = y-y1
        dy2 = y-y2
        if (dy1*dy2.le.0) then
c            print *,y1,'<',y,'<',y2,'  dy=',dy1,dy2
          ddy = y2-y1
          ddyinv= 1/ddy
          jsave = j-1
          if (jsave.eq.0) jsave = 1
          goto 220
        endif
      enddo
 220  continue
c       Field coefficients and its derivatives
          t  = dx1*ddxinv
          ct = 1-t
          u  = dy1*ddyinv
          cu = 1-u
        do k=1,6
          ee1 = et(k,i,j)
          ee2 = et(k,i+1,j)
          ee3 = et(k,i+1,j+1)
          ee4 = et(k,i,j+1)
          ee(k) = (ct*cu*ee1 +t*cu*ee2 +t*u*ee3 +ct*u*ee4)
        enddo

      return

c Restore mesh searching start
 300  isave = 1
      jsave = 1

      end
c-----------------------------------------------------------------
      subroutine outp(z,v,dv,ihlf,ndim,prmt)

      implicit real*8 (a-h,o-z)
      parameter (pi=3.14159265358979324d0)
      parameter (ale90=2.286,ale75=1.905,twoinch=0.0508,
     +           dist=0.660331232)
      dimension v(4),dv(4),prmt(5)
      logical verbose
      save iprint1,iprint2,iprint3,iprint4,iprint5,iprint6

      common /varb/zpr,dzpr,bx,by,bz,verbose

      if (z.ge.zpr) then
        zpr = zpr +dzpr
```

```
          if (verbose) write(*, 1020) z,v,bx,by,bz
                       write(20,1020) z,v,bx,by,bz
        endif

 1020   format(f7.3,4f13.7,1p3e14.5)

                                    end
c-----------------------------------------------------------------------
        subroutine popul8(np,jp,xo,xpo,yo,ypo)

        implicit real*8 (a-h,o-z)
        parameter (pi=3.14159265358979324d0)
        dimension cnt(4)
        real*4  ran1
        logical random,linex
        save iseed,random,sigx,sigy,sigxp,sigyp,thetax,thetay,cnt,
       +     linex,xmin,xmax
        namelist /popul/
       + cnt,iseed,random,emi,betax,betay,chi,alfax,alfay,
       + linex,xmin,xmax

        if (jp.eq.1) read (10,popul)

        if (linex) then
            xo  = xmin
          if (np.ne.1) xo = xmin+2*(jp-1)*xmax/(np-1)
            xpo = cnt(2)
            yo  = cnt(3)
            ypo = cnt(4)
          return
        endif

        if (jp.ne.1) goto 100

c First (or only) point in the center
          xo     = cnt(1)
          xpo    = cnt(2)
          yo     = cnt(3)
          ypo    = cnt(4)
          coup   = 1 +chi*chi
          sigx   = sqrt(emi*betax/coup)
          sigy   = chi*sqrt(emi*betay/coup)
          sigxp  = sqrt(emi/(betax*coup))
          sigyp  = chi*sqrt(emi/(betay*coup))
c Random starting phase on contour
          thetax= 2*ran1(-iseed)-1
          thetay= 2*ran1(-iseed-13)-1
        if (np.eq.1.or.jp.eq.1) return

c Coordinates of the next particles

 100    if (random) then
c         Random extraction
          xo  = cnt(1)+sigx *(2*ran1(-iseed-2*jp)-1)
          xpo = cnt(2)+sigxp*(2*ran1(-iseed-5*jp)-1)
          yo  = cnt(3)+sigy *(2*ran1(-iseed-jp)-1)
          ypo = cnt(4)+sigyp*(2*ran1(-iseed-185*jp)-1)
        else
c         Extract at equal intervals on an ellipse.
          xo  = cnt(1)+sigx *cos(thetax)
          xpo = cnt(2)+sigxp*sin(thetax)
          thetax= thetax+2*pi/(np-1)
          yo  = cnt(3)+sigy *cos(thetay)
          ypo = cnt(4)+sigyp*sin(thetay)
          thetay= thetay+2*pi/(np-1)
        endif

        end
c-----------------------------------------------------------------
        subroutine fopen(ii)
```

```
      implicit real*8 (a-h,o-z)
      integer  fid,fidend
      dimension vo(4,516),vf(4,516)
      character*12  title,fout(20)
      character*80  record(20)
      save  j,vo,vf,fout,fidend
      namelist/labels/title,fout
      common /plotvar/np,zo,xo,xpo,alfax,yo,ypo,z,x,xp,y,yp,alfay
      data title,fout/21*'            '/

      goto (100,200,300), ii

c Open output files
 100  read(10,labels)
      do i=1,20
        if(fout(i)(1:1).eq.' ') goto 110
      enddo
 110    if (i.eq.1) return
        fidend=18+i
      do fid = 20,fidend
        open (fid,file=fout(fid-19))
        write(fid,'(''!'',2x,a)') title
      enddo
              j = 1
      return

c Collect phase space coordinates
 200  vo(1,j) = xo
      vo(2,j) = xpo
      vo(3,j) = yo
      vo(4,j) = ypo
      vf(1,j) = x
      vf(2,j) = xp
      vf(3,j) = y
      vf(4,j) = yp
            j = j+1
      return

c Write to output files
 300   write(20,'(''L5   z x y bx by bz'')')
       write(21,'(''L1 initial x-phase space. zo,xo,xpo='',3g15.6)')
     +                           zo,(vo(i,1),i=1,2)
      do j=2,np
        write(21,'(1p8g15.6)')  (vo(i,j),i=1,2)
      enddo
        write(22,'(''L1 final x-phase space. zf,xf,xpf='',3g15.6)')
     +                            z,(vf(i,1),i=1,2)
      do j=2,np
        write(22,'(1p8g15.6)')  (vf(i,j),i=1,2)
      enddo
        write(23,'(''L1 alfax = '',g15.6)') alfax
          xoo = 4.e-3
        write(23,'(2g13.5)') -xoo,-alfax*xoo
        write(23,'(2g13.5)')  xoo, alfax*xoo

        write(24,'(''L1 initial y-phase space. zo,yo,ypo='',3g15.6)')
     +                            zo,(vo(i,1),i=3,4)
      do j=2,np
        write(24,'(1p8g15.6)')  (vo(i,j),i=3,4)
      enddo
        write(25,'(''L1 final y-phase space. zf,yf,ypf='',3g15.6)')
     +                            z,(vf(i,1),i=3,4)
      do j=2,np
        write(25,'(1p8g15.6)')  (vf(i,j),i=3,4)
      enddo
        write(26,'(''L1 alfay = '',g15.6)') alfay
          yoo = 4.e-3
        write(26,'(2g13.5)') -yoo,-alfay*yoo
        write(26,'(2g13.5)')  yoo, alfay*yoo
```

```
      do fid = 20,fidend
        print *,'file ',fout(fid-19),' written'
      enddo
                                  end
c-------------------------------------------------------------
      subroutine spline (x,y,z,bx,by,bz)
c-------------------------------------------------------------
      FUNCTION RAN1(IDUM)
c Numerical Recipes
c-------------------------------------------------------------
      subroutine dhpcg(prmt,y,dery,ndim,ihlf,fct,outp,aux)
c     Hamming Predictor-Corrector integration routine
*******************************************************************
* Created: Thursday, June 16, 1994   4:39:11 pm (EDT)
* Mod: Friday, August 12, 1994   7:18:59 pm (EDT)
```

Appendix 4. Input file TRACK.DAT

```
$beam
 ekin=29.2619
 np=1
$
$run
 spline=.f.
 prmto=0.d0, 2.40d0, 1.d-4, 1.d-6, 0
 zpr=0.
 dzpr=0.01
 verbose=.true.
$
$field
 ffile='hel.bmap'
 bo=-4.d0
 len=2.4d0
 lambda=2.4d0
 rth=-1.d0
$
$labels
 title='helical snake'
 fout(1)='traj','phsp1','phsp2','phsp3','phsp4','phsp5','phsp6'
$
$popul
 cnt(1)=  4*0.d0,-.01d0,3*0.d0
 emi=1.d-6
 betax=20.d0
 betay=20.d0
 chi=1.d0
 alfax=0
 alfay=0
 iseed=132406
 random=.false.
 linex=.false.
 xmin=-0.04
 xmax= 0.04
$
```