Deriving Second-Order Terms for the Solenoid-Quadrupole Transport Map

Christopher D. Tennant

Department of Physics, Ithaca College, Ithaca, NY 14850

Abstract

Since the equations for the second-order terms for a combination solenoid-quadrupole element have not been (as far as we can tell) published, these need to be derived. Starting with the differential equations of motion for a particle under the influence of a solenoid-quadrupole field, this paper derives the transport map and the associated second-order terms.

Introduction

For the past several years, Wilson Synchrotron Laboratory has been writing its own particle tracking subroutines. The collective code is known as BMAD, an acronym for **B**aby **M**ethodical **A**ccelerator **D**esign. Transfer maps offer a convenient and efficient formalism for computer programs to track particles through the optics of CESR. For long term tracking in CESR it becomes necessary to consider second-order terms of these transport maps. Since the author is not aware of any previously published paper on this derivation, this paper derives the second-order terms, starting with the differential equations of motion for a particle in a solenoid-quadrupole field. The motivation behind this derivation is to incorporate the results into BMAD for increased accuracy in tracking particles through CESR.

The Calculations

I. Elements of Solenoid-Quadrupole Transfer Matrix

The differential equations of motion for a particle under the influence of the combined solenoid-quaudrople field are given by

$$x'' = -k_1 x - k_s y' \tag{1}$$

$$y'' = k_1 y + k_s x' \tag{2}$$

where k_1 is the strength of the quadrupole and k_s is the strength of the solenoid. We make a guess of the solutions to be of the form

$$x = A\sin(\alpha s) + B\cos(\alpha s) + C\sinh(\beta s) + D\cosh(\beta s) \tag{3}$$

$$y = E\sin(\alpha s) + F\cos(\alpha s) + G\sinh(\beta s) + H\cosh(\beta s) \tag{4}$$

where s is the length of the solenoid and α and β are to be determined.

Equations (3) and (4) can be simplified and written using complex notation as

$$x = Re[Je^{i\alpha s}]$$

$$y = Re[Ke^{i\alpha s}].$$

After taking the appropriate derivatives of the previous complex equations, plugging into equations (1) and (2), and doing some algebra we are left to solve the quadratic

$$\alpha^4 - \alpha^2 k_s^2 - k_1^2 = 0$$

which readily gives

$$\alpha^2 = \frac{k_s^2 + \sqrt{k_s^4 + 4k_1^2}}{2}$$

$$\beta^2 = \frac{\sqrt{k_s^4 + 4k_1^2} - k_s^2}{2}.$$

Now that we have determined what α and β are, the constants in equations (3) and (4) need to be determined. This is done by taking the appropriate derivatives of equations (3) and (4) and plugging into equation (1). Gathering all terms to one side and gathering coefficients of like functions, we recognize that the only way the expression can be zero is if all the coefficients are zero. This leads to the following four equations

$$\begin{aligned} A(k_1 - \alpha^2) &= k_s \alpha F \\ B(k_1 - \alpha^2) &= -k_s \alpha E \\ C(k_1 + \beta^2) &= -k_s \beta H \\ D(k_1 + \beta^2) &= -k_s \beta G. \end{aligned} \tag{5}$$

We also evaluate equations (3) and (4) and their first derivatives at zero to yield the additional four equations

$$\begin{aligned} x(0) &= B + D \\ y(0) &= F + H \\ x'(0) &= A\alpha + C\beta \\ y'(0) &= E\alpha + G\beta. \end{aligned} \tag{6}$$

We are now equipped with eight equations (equation sets (5) and (6)) and eight unknown constants. Using the "Solve" capabilities of Mathematica, we are able to quickly solve for the eight constants in terms of α, β, k_1, k_s, and the initial conditions. The elements of the solenoid-quadrupole transfer matrix are found by plugging these constants in equations (3) and (4) giving

$$(x|x_0) = \frac{1}{(\alpha^2+\beta^2)} \left[(\beta^2+k_1)\cos(\alpha s) + (\alpha^2-k_1)\cosh(\beta s)\right]$$

$$(x|x_0') = \frac{1}{(\alpha^2+\beta^2)k_1} \left[\alpha(\beta^2+k_1)\sin(\alpha s) + \beta(k_1-\alpha^2)\sinh(\beta s)\right]$$

$$(x|y_0) = \frac{k_s}{(\alpha^2+\beta^2)k_1} \left[-\beta^2\alpha\sin(\alpha s) + \beta\alpha^2\sinh(\beta s)\right]$$

$$(x|y_0') = \frac{k_s}{(\alpha^2+\beta^2)} [-\cos(\alpha s) + \cosh(\beta s)]$$

and

$$(y|x_0) = (x|y_0)$$

$$(y|x_0') = -(x|y_0')$$

$$(y|y_0) = \frac{1}{(\alpha^2+\beta^2)k_1} \left[\beta^2(k_1-\alpha^2)\cos(\alpha s) + \alpha^2(\beta^2+k_1)\cosh(\beta s)\right]$$

$$(y|y_0') = \frac{1}{(\alpha^2+\beta^2)} \left[\frac{(\alpha^2-k_1)}{\alpha}\sin(\alpha s) + \frac{(\beta^2+k_1)}{\beta}\sinh(\beta s)\right].$$

The elements $(x'|x_0)$, $(x'|x_0')$, $(y'|y_0)$, etc. are found by differentiating the above eight matrix elements with respect to the solenoid length, s. The remaining elements of the matrix are

$$(x'|x_0) = \frac{1}{(\alpha^2+\beta^2)} \left[-\alpha(\beta^2+k_1)\sin(\alpha s) + \beta(\alpha^2-k_1)\sinh(\beta s)\right]$$

$$(x'|x_0') = \frac{1}{(\alpha^2+\beta^2)k_1} \left[\alpha^2(\beta^2+k_1)\cos(\alpha s) + \beta^2(k_1-\alpha^2)\cosh(\beta s)\right]$$

$$(x'|y_0) = \frac{k_s k_1}{(\alpha^2+\beta^2)} [-\cos(\alpha s) + \cosh(\beta s)]$$

$$(x'|y_0') = \frac{k_s}{(\alpha^2+\beta^2)} [\alpha\sin(\alpha s) + \beta\sinh(\beta s)]$$

$$(y'|x_0) = (x_0|y_0)$$

$$(y'|x_0') = -(x_0'|y_0')$$

$$(y'|y_0) = \frac{1}{(\alpha^2+\beta^2)k_1} \left[\alpha\beta^2(\alpha^2-k_1)\sin(\alpha s) + \beta\alpha^2(\beta^2+k_1)\sinh(\beta s)\right]$$

$$(y'|y_0') = \frac{1}{(\alpha^2+\beta^2)} \left[(\alpha^2-k_1)\cos(\alpha s) + (\beta^2+k_1)\cosh(\beta s)\right].$$

The transfer matrix is then constructed in the following manner

$$M = \begin{pmatrix} (x|x_0) & (x|x_0') & (x|y_0) & (x|y_0') \\ (x'|x_0) & (x'|x_0') & (x'|y_0) & (x'|y_0') \\ (y|x_0) & (y|x_0') & (y|y_0) & (y|y_0') \\ (y'|x_0) & (y'|x_0') & (y'|y_0) & (y'|y_0') \end{pmatrix}. \qquad (7)$$

II. Transfer Matrix Elements Including Fringe Field Effects

Thus far we have neglected the fringe field effects from the solenoid. To include those effects, we simply matrix multiply the two fringe field matrices with the matrix (7) derived in the previous section. The fringe field matrices are given by

$$f = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & \frac{k_s}{2} & 0 \\ 0 & 0 & 1 & 0 \\ \frac{-k_s}{2} & 0 & 0 & 1 \end{pmatrix}$$

and its inverse. The solenoid-quadrupole transport map, including fringe field effects, is given by

$$\bar{R}_{ij} = f^{-1} \cdot M \cdot f. \qquad (8)$$

The elements of R_{ij} are given explicitly in Appendix A.

III. Second-Order Terms of the Solenoid-Quadrupole Transport Map

To calculate the second-order terms we explicitly include the dependence of k_1 and k_s on the energy $x_6 = z'$.

$$k_1 = \frac{k_1(0)}{1 + x_6} \qquad (9)$$

$$k_s = \frac{k_s(0)}{1 + x_6}.$$

Since we are interested in only the second-order terms, we are allowed to expand equations (9) to first order,

$$k_1 = k_1(0)(1 - x_6) \qquad (10)$$

$$k_s = k_s(0)(1 - x_6).$$

Equation (10) is used in the preceding equations making $\bar{R}(x_6)$ dependent on x_6. There is a slight twist though. The coordinates we have chosen for the analysis are (x, x', y, y', z, z'). However we need to use the canonical coordinates (x, P_x, y, P_y, z, P_z), where to first order

$$P_x = x'(1+x_6)$$

$$P_y = y'(1+x_6)$$

$$P_z = z'.$$

The coordinate transformation, $(x, x', ..., z') \to (x, P_x, ..., P_z)$, is done via N:

$$N = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1+x_6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1+x_6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

The map R in $(x, P_x, ..., P_z)$ coordinates is then

$$R(x_6) = N \cdot \bar{R}(x_6) \cdot N^{-1}.$$

The T_{ijk}, where $k = 6$, terms ("The 6-Terms") are

$$T_{ij6} = \left. \frac{\partial R_{ij}(x_6)}{\partial x_6} \right|_{x_6=0}.$$

By integration, the remainder of the second-order terms, T_{ijk}, where $i = 5$ ("The 5-Terms"), are found. The necessary integration takes on the general form

$$\Delta x_5 = -\frac{1}{2} \int_0^{\ell} \left(x_2^2(s) + x_4^2(s) \right) ds \tag{11}$$

where ℓ denotes the solenoid length and

$$x(s) = R' \cdot x(0) \tag{12}$$

$$R' = M \cdot f. \tag{13}$$

Since we are interested in the particle path *within* the solenoid, we only include the first fringe field in defining R'. Using equations (11) and (12) in (10) gives Δx_5 in the form

$$\Delta x_5 \equiv T_{5jk} x_j(0) x_k(0).$$

All of the second-order terms are given in the BMAD code attached to this paper.

Conclusion

With a documented, systematic derivation of the second-order terms for the solenoid-quadrupole transport map, we expect to improve the tracking accuracy of BMAD. Specifically, we should be able to do accurate long-term tracking.

Acknowledgments

I am pleased to acknowledge Dr. David Sagan of Cornell University who proposed this Research Experience for Undergraduates. I would like to especially thank him for his unwavering patience as he guided me through these calculations. I would also like to acknowledge Kern Ormond of Cornell University for taking time away from writing his thesis to help me in using LaTeX. This work was supported by the National Science Foundation REU grant PHY-9731882 and research grant PHY-9809799.

Appendix A: Elements of the Solenoid-Quadrupole Transport Map with Fringe Field Effects

Let t $= \alpha^2 + \beta^2 = \sqrt{k_s^2 + 4k_1^4}$

$$
\begin{aligned}
R_{11} &= \frac{1}{2t}\left[(t+2k_1)\cos(\alpha s) + (t-2k_1)\cosh(\beta s)\right] \\
R_{12} &= \frac{1}{tk_1}\left[\alpha(k_1+\beta^2)\sin(\alpha s) + \beta(k_1-\alpha^2)\sinh(\beta s)\right] \\
R_{13} &= \frac{k_s}{2tk_1}\left[\alpha(k_1-\beta^2)\sin(\alpha s) + \beta(k_1+\alpha^2)\sinh(\beta s)\right] \\
R_{14} &= \frac{k_s}{t}\left[-\cos(\alpha s) + \cosh(\beta s)\right] \\
R_{21} &= \frac{-1}{4t}\left[\frac{k_s^2(-k_1+\alpha^2)+4k_1(k_1+\alpha^2)}{\alpha}\sin(\alpha s) + \frac{k_s^2(k_1+\beta^2)+4k_1(-k_1+\beta^2)}{\beta}\sinh(\beta s)\right] \\
R_{22} &= R_{11} \\
R_{23} &= \frac{k_s^3}{4t}\left[\cos(\alpha s) - \cosh(\beta s)\right] \\
R_{24} &= \frac{k_s}{2t}\left[\frac{(k_1+\alpha^2)}{\alpha}\sin(\alpha s) + \frac{(-k_1+\beta^2)}{\beta}\sinh(\beta s)\right] \\
R_{31} &= -R_{24} \\
R_{32} &= -R_{14} \\
R_{33} &= \frac{1}{2t}\left[(t-2k_1)\cos(\alpha s) + (t+2k_1)\cosh(\beta s)\right] \\
R_{34} &= \frac{1}{t}\left[\frac{(-k_1+\alpha^2)}{\alpha}\sin(\alpha s) + \frac{(k_1+\beta^2)}{\beta}\sinh(\beta s)\right] \\
R_{41} &= -R_{23} \\
R_{42} &= \frac{k_s}{2tk_1}\left[\alpha(-k_1+\beta^2)\sin(\alpha s) - \beta(k_1+\alpha^2)\sinh(\beta s)\right] \\
R_{43} &= \frac{1}{4tk_1}\left[[4k_1(k_1-\beta^2) - k_s^2(k_1+\beta^2)\ \alpha]\sin(\alpha s) + [4k_1(k_1+\alpha^2) - k_s^2(k_1-\alpha^2)]\beta\sinh(\beta s)\right] \\
R_{44} &= R_{33}
\end{aligned}
$$

Appendix B: BMAD Code

```
subroutine sol_quad_calc (ks, k1, s_len, t)

 use bmad_struct

 implicit none

 real ks, ks2, k1, s, c, snh, csh, s_len
 real t(6,27), q_, r_, a_, b_
 real darg1, alpha, alpha2, beta, beta2, f, q, r, a, b
 real df, dalpha2, dalpha, dbeta2, dbeta, darg
 real dC, dCsh, dS, dSnh, dq, dr, da, db
 real ks3, fp, fm, dfm, dfp, df_f, ug
 real s1, s2, snh1, snh2, dsnh1, dsnh2, ds1, ds2
 real coef1, coef2, dcoef1, dcoef2, ks4

 real IS2, IC2, ISnh2, ICsh2, ISCsh, ICSnh
 real ISC, ICCsh, ISSnh, ISnhCsh
 real v1, v2, v3, v4, s11, s22, s3, ct, vp, vm

! Calc

 ks2 = ks**2
 ks3 = ks2 * ks
 ks4 = ks2*ks2
 f = sqrt(ks4 + 4*k1**2)
 ug = 1 / (4*f)
 alpha2 = (f + ks2) / 2; alpha = sqrt(alpha2)
 beta2 = (f - ks2) / 2; beta = sqrt(beta2)
 S = sin(alpha*s_len)
 C = cos(alpha*s_len)
 Snh = sinh(beta*s_len)
 Csh = cosh(beta*s_len)
 q = f + 2*k1 - ks2
 r = f - 2*k1 + ks2
 a = f + 2*k1 + ks2
 b = f - 2*k1 - ks2
 fp = f + 2*k1
 fm = f - 2*k1

 df = -2 * (ks4 + 2*k1**2) / f
 dalpha2 = df/2 - ks2
 dalpha = (df/2 - ks2)/(2*alpha)
```

```
dbeta2 = ks2 + df/2
dbeta = (ks2 + df/2)/(2*beta)
darg = s_len*dalpha
darg1 = s_len*dbeta
dC = -darg*S
dCsh = darg1*Snh
dS = darg*C
dSnh = darg1*Csh
dq = -2*k1 + 2*ks2 + df
dr = 2*k1 - 2*ks2 + df
da = -2*k1 - 2*ks2 + df
db = 2*k1 + 2*ks2 + df
dfp = df - 2*k1
dfm = df + 2*k1
df_f = -df/f

S1 = S * alpha
S2 = S / alpha
dS1 = dS * alpha + S * dalpha
dS2 = dS / alpha - S * dalpha / alpha2

Snh1 = Snh * beta
Snh2 = Snh / beta
dSnh1 = dSnh * beta + Snh * dbeta
dSnh2 = dSnh / beta - Snh * dbeta / beta2

coef1 = ks2*r + 4*k1*a
coef2 = ks2*q + 4*k1*b

dcoef1 = -2*ks2*r + ks2*dr - 4*k1*a + 4*k1*da
dcoef2 = -2*ks2*q + ks2*dq - 4*k1*b + 4*k1*db

!

t = 0
call mat_unit(t(1:6,1:6), 6, 6)

t(1,1) = 2*ug * (fp*C + fm*Csh)
t(1,2) = (2*ug/k1) * (q*S1 - r*Snh1)
t(1,3) = (ks*ug/k1) * (-b*S1 + a*Snh1)
t(1,4) = 4*ug*ks * (-C + Csh)

t(2,1) = -(ug/2) * (coef1*S2 + coef2*Snh2)
t(2,2) = t(1,1)
t(2,3) = ug*ks3 * (C - Csh)
```

```
 t(2,4) = ug*ks * (a*S2 + b*Snh2)

 t(3,1) = -t(2,4)
 t(3,2) = -t(1,4)
 t(3,3) = 2*ug * (fm*C + fp*Csh)
 t(3,4) = 2*ug * (r*S2 + q*Snh2)

 t(4,1) = -t(2,3)
 t(4,2) = -t(1,3)
 t(4,3) = (ug/(2*k1)) * (-coef2*S1 + coef1*Snh1)
 t(4,4) = t(3,3)

 t(1,x16$) = t(1,1)*df_f + 2*ug*(fp*dC + C*dfp + fm*dCsh + Csh*dfm)
 t(1,x26$) = t(1,2)*df_f + (2*ug/k1) * (dq*S1 + q*dS1 - dr*Snh1 - r*dSnh1)
 t(1,x36$) = t(1,3)*df_f + (ks*ug/k1)*(-db*S1 - b*dS1 + da*Snh1 + a*dSnh1)
 t(1,x46$) = t(1,4)*(df_f - 2) + 4*ks*ug*(-dC + dCsh)

 t(2,x16$) = t(2,1)*(df_f + 1) - &
 (ug/2)*(dcoef1*S2 + coef1*dS2 + dcoef2*Snh2 + coef2*dSnh2)
 t(2,x26$) = t(1,x16$)
 t(2,x36$) = t(2,3)*(df_f - 2) + ks3*ug*(dC - dCsh)
 t(2,x46$) = t(2,4)*(df_f - 1) + ug*ks*(da*S2 + a*dS2 + db*Snh2 + b*dSnh2)

 t(3,x16$) = -t(2,x46$)
 t(3,x26$) = -t(1,x46$)
 t(3,x36$) = t(3,3)*df_f + 2*ug*(fm*dC + C*dfm + fp*dCsh + Csh*dfp)
 t(3,x46$) = t(3,4)*(df_f - 1) + 2*ug*(dr*S2 + r*dS2 + dq*Snh2 + q*dSnh2)

 t(4,x16$) = -t(2,x36$)
 t(4,x26$) = -t(1,x36$)
 t(4,x36$) = t(4,3)*(df_f + 2) + &
 (ug/(2*k1))*(-dcoef2*S1 - coef2*dS1 + dcoef1*Snh1 + coef1*dSnh1)
 t(4,x46$) = t(3,x36$)

! 5 terms

ct = 8*ug**2

IS2 = alpha*(alpha*s_len - S*C)/2
IC2 = (alpha*s_len + S*C)/(2*alpha)

ISnh2 = beta*(-beta*s_len + Snh*Csh)/2
ICsh2 = (beta*s_len + Snh*Csh)/(2*beta)

ISCsh = (alpha - alpha*C*Csh + beta*S*Snh)/f
```

```
ICSnh = (-beta + beta*C*Csh + alpha*S*Snh)/f

ISC = (S**2)/(2*alpha)
ICCsh = (alpha*Csh*S + beta*C*Snh)/f
ISSnh = alpha*beta*(beta*Csh*S - alpha*C*Snh)/f
ISnhCsh = (Snh**2)/(2*beta)

V1 = alpha * (a*ISC + b*ISCsh)
V2 = beta * (a*ICSnh + b*ISnhCsh)
V3 = alpha * (r*ISC + q*ISCsh)
V4 = beta * (r*ICSnh + q*ISnhCsh)

Vp = f * (IS2 - ISnh2) + (a*r*IC2 + b*r*ICCsh + a*q*ICCsh + b*q*ICsh2)/4
Vm = 2*k1*(IS2 + 2*ISSnh + ISnh2)

S11 = r**2*IC2 + 2*q*r*ICCsh + q**2*ICsh2
S22 = a**2*IC2 + 2*b*a*ICCsh + b**2*ICsh2
S3 = 4*ks2 * (IS2 + 2*ISSnh + ISnh2)

t(5,x11$) = -(ct/4)*(fp**2*IS2 - 2*ks4*ISSnh + fm**2*ISnh2 + ks2*S22/4)
t(5,x12$) = (ct/2)*(q*V1 - r*V2)
t(5,x13$) = ((ct*ks)/4)*(-fm*V1 + fp*V2 + fp*V3 - fm*V4)
t(5,x14$) = ct*ks*(vp + vm)

t(5,x22$) = -(ct/4)*(S22 + S3)
t(5,x23$) = -ct*ks*(vp - vm)
t(5,x24$) = (ct*ks)*(V3 - V1 + V4 - V2)

t(5,x33$) = -(ct/4)*(fm**2*IS2 - 2*ks4*ISSnh + fp**2*ISnh2 + ks2*S11/4)
t(5,x34$) = (ct/2)*(b*V3 - a*V4)
t(5,x44$) = -(ct/4)*(S3 + S11)

end subroutine
```