Transfer matrix for combined dipole + bent solenoid

Juan C. Gallardo
Center for Accelerator Physics,
Brookhaven National Laboratory, Upton, NY 11973, USA

December 15, 1998

Abstract

We give explicitly the transfer matrix for a combined dipole + bent solenoid. The expression are not original, they were derived by C. Wang and A. Chao [*Transfer matrices of superimposed magnets and rf cavity*, Chun-xi Wang and Alex Chao, SLAC/AP-106, Nov. 1996].

1 Introduction

Bent solenoids are of utmost importance in the emittance exchange scenarios, presently favored [1] as baseline design of a muon collider [2].

The dynamics of muons is complicated and subject to debate. In a previous note [3] a Hamiltonian formulation was presented. However, its implementation in numerical codes has not being successful because of the need of simpler, faster codes that are amenable to optimization algorithm of the many parameters in the beam line.

Solenoidal channels with straight solenoids have shown to be less friendly to standard matching procedures in beam physics. In particular, particles with large amplitude (transverse position) and/or angle (respect to the beam axis) have aberrations which call for more sophisticated matching procedures than the one used so far. This is compounded with the introduction of bent solenoids.

Numerical codes like, MAD, Transport, Synch, Teapot, etc, use 6x6 matrices to represent the magnetic elements and the phase space of the beam is calculated by

$$\vec{z}_{\rm out} = \overset{\leftrightarrow}{\mathcal{M}} \cdot \vec{z}_{\rm in} \tag{1}$$

with $\vec{z}_{\rm in} = (x, p_x, y, p_y, z, \frac{\delta p}{p})$, $z = s - c\beta_o t$, and β_o is the reference energy.

The attraction of this approach is multiple, firstly most beam physicists are familiar with the formulation and second, most important, it allows the use of well understood matching conditions of the Twiss functions.

2 Calculation

The normalized (to the reference momentum p_o) Hamiltonian function is

$$\begin{aligned} H^{\mathrm{dip}}_{\mathrm{tor.sol.}}(x,p_x,y,p_y,z,\delta;s) &\approx \tfrac{1}{2}(hx)^2 - \delta hx \\ &+ \tfrac{(1+hx)}{2(1+\delta)}\left\{(p_x - \tfrac{q}{p_o}A_x)^2 + (p_y - \tfrac{q}{p_o}A_y)^2\right\} \end{aligned} \tag{2}$$

where p_x and p_y are the normalized momentum. The vector potential for a dipole is:

$$A_x = -\frac{B_D}{2h(s)}\frac{h''(s)\,x(y^2-\frac{1}{3}x^2)}{(1+h(s)x)} \tag{3}$$

$$A_y = \frac{B_D}{2h(s)}\frac{h'(s)\,xy}{(1+h(s)x)} \tag{4}$$

$$A_s = -\frac{B_D}{2h(s)}\left\{1+h(s)\,x - h'(s)\,(y^2-x^2)\right\} \tag{5}$$

and for a bent solenoid is:

$$\vec{A} = -\frac{1}{2}B_s\frac{y}{(1+hx)}\vec{e}_x + \frac{B_s}{2h}\ln(1+hx)\vec{e}_y \tag{6}$$

Following the method explained in details in [5] we can write the 6x6 linear transfer matrix with elements:

$$\begin{array}{lll}
R_{11} = a_- + a_+\cos Bs & R_{12} = \frac{\sin Bs}{B} & R_{13} = \frac{B_o}{2B}\sin Bs \\
R_{14} = \frac{B_o}{B^2}(1-\cos Bs) & R_{15} = 0 & R_{16} = \frac{h}{B^2}(1-\cos Bs) \\
R_{21} = s\frac{h^2}{2}a_- - a_+^2 B\sin Bs & R_{22} = R_{11} & R_{23} = -\frac{a_+ B_o}{2}(1-\cos Bs) \\
R_{24} = s\frac{h^2}{B_o}a_- + a_+\frac{B_o}{B}\sin Bs & R_{25} = 0 & R_{26} = \frac{h}{B}(a_+\sin Bs + a_- Bs) \\
R_{31} = -R_{24} & R_{32} = -R_{14} & R_{33} = a_+ + a_-\cos Bs \\
R_{34} = s(\frac{h}{B})^2 + 2\frac{a_-}{B}\sin Bs & R_{35} = 0 & R_{36} = \frac{hB_o}{B^3}(\sin Bs - Bs) \\
R_{41} = -R_{23} & R_{42} = -R_{13} & R_{43} = -\frac{a_-}{2}B\sin Bs \\
R_{44} = R_{33} & R_{45} = 0 & R_{46} = -\frac{hB_o}{2B^2}(1-\cos Bs) \\
R_{51} = -R_{26} & R_{52} = -R_{16} & R_{53} = R_{46} \\
R_{54} = R_{36} & R_{55} = 1 & R_{56} = \frac{h^2}{B^3}(\sin Bs - Bs) \\
R_{61} = 0 & R_{62} = 0 & R_{63} = 0 \\
R_{64} = 0 & R_{65} = 0 & R_{66} = 1
\end{array}$$

where $B = \sqrt{B_o^2 + h^2}$, $B_o = \frac{q}{p_o}B_s$ and $a_\pm = \frac{1}{2}\left(1 \pm (\frac{h}{B})^2\right)$.

It can be verified that taking the limit $h \to 0$ we recover the straight solenoid transfer matrix and taking the limit $B_o \to 0$ we obtain the sector dipole transfer matrix. Figures 1 and 2 show example of the phase space after three bend solenoids transformations following the so-called **Norem configuration**.

In the appendix we reproduce a simple fortran code; the code is available at the URL **http://pubweb.bnl.gov/people/gallardo/bent/**

3 Acknowledgments

This work was supported by the U.S. DoE, under Contract No. DE-AC02-76CH00016. I wish to thank R. Palmer, R. Fernow and J. Norem for useful suggestions.

References

[1] R. B. Palmer, private communication.

[2] C. Ankenbrandt, *et al.*, *Status of Muon Collider Research and Development and Future Plans, in preparation.*

[3] J. C. Gallardo, R. C. Fernow and R. B. Palmer, Muon Dynamics in a Toroidal Sector Magnet, in Proc. 4^{th} Int. Conf. on the Physics Potential & Development of $\mu^+\mu^-$ Colliders *(San Francisco, CA, Dec. 10-12, 1997), AIP Conf. Proc.* **441**, *p. 282 (1998).*

[4] C. Wang and A. Chao, Notes on Lie algebraic analysis of achromats, SLAC/AP-100, Jan. 1995.

[5] C. Wang and A. Chao, Transfer matrices of superimposed magnets and RF cavity, SLAC/AP-106, Nov. 1996.

4 Appendix

```
        program bent  ! linear matrix a la Transport
c  Juan C. Gallardo, Nov. 25, 1998.
        implicit double precision (a-h,o-z)
        dimension trans(6,6),xini(6),xfin(6),xx(3),pp(3),coord(5000,6)
cccccccccccccccccccccccccccccccccccccccccccccccccccccc
        character*9 fileunit1(3),fileunit2(3),fileunit3(3)
        common/data/B,bo,h
        data (fileunit1(i),i=1,3)/'x-px1.dat','x-px2.dat','x-px3.dat'/
        data (fileunit2(i),i=1,3)/'y-py1.dat','y-py2.dat','y-py3.dat'/
        data (fileunit3(i),i=1,3)/'z-pz1.dat','z-pz2.dat','z-pz3.dat'/
        parameter(cligth=2.9979d8)
        open(unit=15,file='ini.dat',status='old',iostat=ioc)
        if(ioc.ne.0)goto 9000
        pi=2*asin(1.0d0)
c        xmass=105.65   ! MeV
        xmom=0.180   ! GeV/c reference momentum
c        ener_kin=sqrt(xmom**2+xmass**2)-xmass
c        gamma=ener_kin/xmass+1.
c        beta=sqrt((gamma+1.)*(gamma-1.))/gamma
c       Bo is the solenoid magnetic field
```

```
c       h is curvature of the solenoid and dipole
c       s length of the solenoid
        write(6,*)'input: solenoid curvature in m'
        read*,rho
        h=1.0d0/rho
        write(6,*)'input: solenoidal magnetic field [T]'
        read*,Bs
        Boo=Bs*0.2998/xmom
        B=sqrt(Boo**2+h**2)
        write(6,*)'input: number of particles in ini.dat'
        read*,npar
        xlambda=2.*pi/B
        write(6,*)'Larmor,wavelength', xlambda
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        jpart=0
        do k=1,npar
        read(15,*,end=888)i1,i2,i3,i4,xini(5),w1,(xx(i),i=1,3),
     +  (pp(i),i=1,3)
        jpart=jpart+1
        end do
 888    write(6,*)'number of used particles is:',jpart
        rewind(15)
        do kount=1,3 ! loop over sections of the bent solenoid
        open(unit=25,file=fileunit1(kount),status='unknown',iostat=ios)
        open(unit=26,file=fileunit2(kount),status='unknown',iostat=ios)
        open(unit=27,file=fileunit3(kount),status='unknown',iostat=ios)
        if(ios.ne.0)goto 9001
        write(6,*)' Do you want to change length solenoid: y(1);n(0)'
        read*,ilen
        if(ilen.eq.1)then
        write(6,*)'input: length of the solenoid'
        read*,s
         else
        s=xlambda
        endif
        write(6,*)'effective length',s
         Bo=0.5*Boo
        do k=1,jpart      ! jpart do loop
c read the input beam coordinates
                   if(kount.eq.1)then
c        B=sqrt(Bo**2+h**2)
        read(15,*)i1,i2,i3,i4,xini(5),w1,(xx(i),i=1,3),
     +  (pp(i),i=1,3)
        xini(1)=xx(1)
        xini(3)=xx(2)
        xini(5)=xini(5)*cligth
```

```
        xini(2)=pp(1)
        xini(4)=pp(2)
        xini(6)=sqrt(pp(1)**2+pp(2)**2+pp(3)**2)/xmom-1.0d0
                   else
         do l=1,6
        xini(l)=coord(k,l)
         end do
          if(kount.eq.2)Bo=Boo
                    endif
        B=sqrt(Bo**2+h**2)
        xini(2)=xini(2)/xmom
        xini(4)=xini(4)/xmom
c need some normalization
c dynamics  xfin=trans*xini
        call matrix(s,trans)

        do 15 i=1,6
         temp=0.0d0
        do 25 j=1,6
        temp=temp+trans(i,j)*xini(j)
 25   end do
        xfin(i)=temp
c        if(i.eq.5)write(6,*)xini(5),xfin(5)
 15   end do
        xfin(2)=xfin(2)*xmom
        xfin(4)=xfin(4)*xmom
        write(25,*)xfin(1),xfin(2)
        write(26,*)xfin(3),xfin(4)
        write(27,*)xfin(5),xfin(6)
        do 75 i=1,6
        coord(k,i)=xfin(i)
 75     end do
        end do     ! jpart do loop
        close(25)
        close(26)
        close(27)
        end do ! loop over different sections of the bent solenoid
        goto 9100
 9000   write(6,*)'error in input file'
        goto 9100
 9001   write(6,*)'error in output file'
 9100   stop
        end
        subroutine matrix(s,trans)
        implicit double precision (a-h,o-z)
        dimension trans(6,6)
```

```
      common/data/B,bo,h
c      write(6,*)Bo,Bs,B,h,'Bo,Bs,B,h'
      trans(1,1)=(Bo/B)**2+(h/B)**2*cos(B*s)
      trans(1,2)=sin(B*s)/B
      trans(1,3)=0.0
      trans(1,4)=Bo/B/B*(1.0d0-cos(B*s))
      trans(1,5)=0.0
      trans(1,6)=h/B/B*(1.0d0-cos(B*s))
      trans(2,1)=-h**2/B*sin(B*s)
      trans(2,2)=cos(B*s)
      trans(2,3)=0.0
      trans(2,4)=Bo/B*sin(B*s)
      trans(2,5)=0.0
      trans(2,6)=h/Bo*trans(2,4)
      trans(3,1)=(h/B)**2*Bo/B*(B*s-sin(B*s))
      trans(3,2)=-Bo/B/B*(1.0d0-cos(B*s))
      trans(3,3)=1.0d0
      trans(3,4)=s*(h/B)**2+(Bo/B)**2/B*sin(B*s)
      trans(3,5)=0.0
      trans(3,6)=-Bo*h/B**3*(B*s-sin(B*s))
      trans(4,1)=Bo*(h/B)**2*(1.0d0-cos(B*s))
      trans(4,2)=-Bo/B*sin(B*s)
      trans(4,3)=0.0
      trans(4,4)=(h/B)**2+(Bo/B)**2*cos(B*s)
      trans(4,5)=0.0
      trans(4,6)=-trans(4,1)/h
      trans(5,1)=-s*h*(Bo/B)**2-(h/B)**3*sin(B*s)
      trans(5,2)=-trans(1,6)
      trans(5,3)=0.0
      trans(5,4)=trans(3,6)
      trans(5,5)=1.0d0
      trans(5,6)=h/Bo*trans(5,4)
      trans(6,1)=0.0
      trans(6,2)=0.0
      trans(6,3)=0.0
      trans(6,4)=0.0
      trans(6,5)=0.0
      trans(6,6)=1.0d0
c      write(6,*)h,Bo,B,(trans(5,i),i=1,6)
      return
      end
```

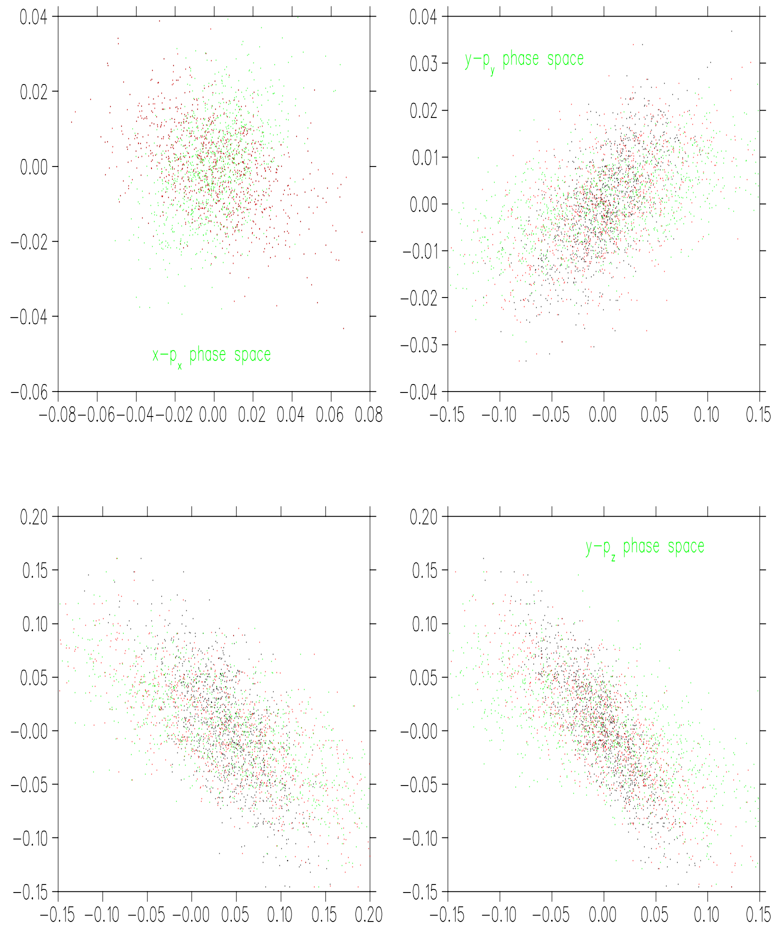

Figure 1: Example of dispersion created by a toroidal sector solenoid plus a vertical dipole. Each panel shows the phase space at the end of the three sections of the bent solenoid in a **Norem configuration** $[\frac{1}{2}B_o, B_o, \frac{1}{2}B_o]$. Black first, red second and green third bent solenoid

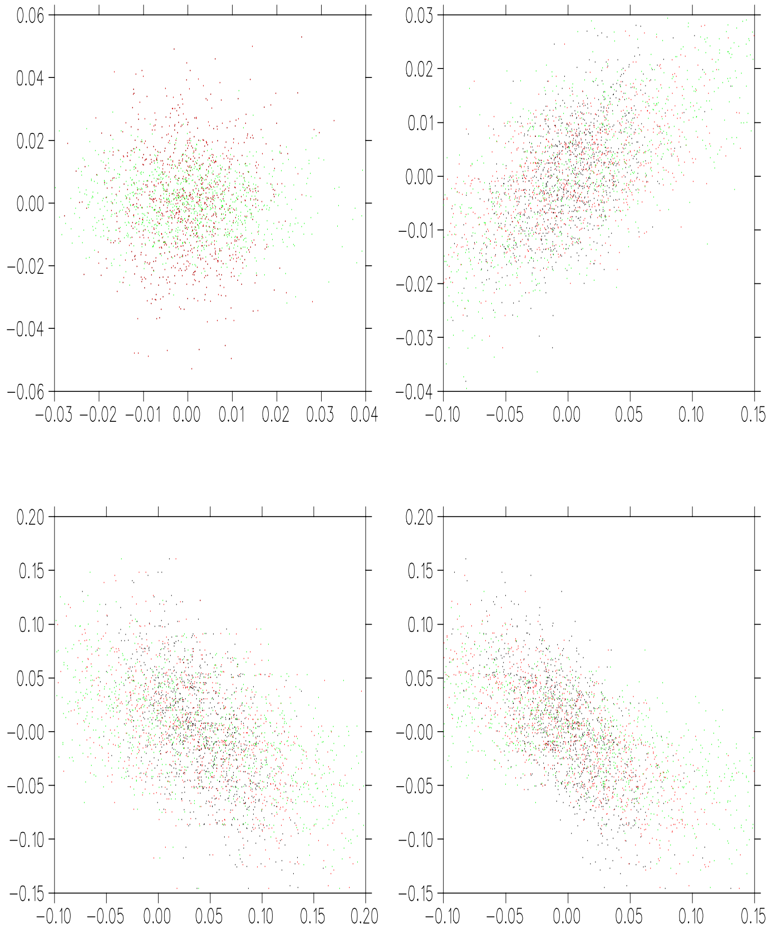

Figure 2: Example of dispersion created by a toroidal sector solenoid plus a vertical dipole. Each panel shows the phase space at the end of the three sections of the bent solenoid in a **Norem configuration** $[2h,h,2h]$. Black first, red second and green third bent solenoid