Beam Physics Note
TRI-BN-20-09
May 21, 2020

TRANSOPTR Routine: FFA

Thomas Planche

TRIUMF

Abstract: In this note I describe the implementation in the envelope code TRANSOPTR of the tracking through an arbitrary static magnetic field with median plane symmetry. I also present a couple of benchmarking examples.

4004 Wesbrook Mall, Vancouver, B.C. Canada V6T 2A3 Tel: 604 222-1047 www.triumf.ca

1 Objective

Let's assume that the reference particle's trajectory lies with the median plane of a fixed (magnetic) field circular accelerator. The objective is to calculate, at every location s along the reference trajectory, the coefficient of the infinitesimal transfer matrix from the vertical component of the magnetic field $B(r, \theta)$, and its partial derivatives.

In practice, I shall get $B(r, \theta)$ and its partial derivative from some numerical interpolation of a 2-dimensional polar map. This should be relatively easy: I just need to find the right bicubic spline FORTRAN library for that, or something like that. For the time being, let's just assume that we know how to evaluate $B(r, \theta)$, and its derivatives.

2 Infinitesimal Matrix

As shown by Rick [1], the quadratic hamiltonian in a static magnet with median plane symmetry is given by:

$$h(x, p_x, y, p_y, z, p_z; s) = \frac{x^2}{2}\frac{1-n}{\rho^2} + \frac{y^2}{2}\frac{n}{\rho^2} + \frac{p_x^2}{2} + \frac{p_y^2}{2} - \frac{p_z x}{\rho} + \frac{p_z^2}{2\gamma^2}\,, \tag{1}$$

where (x, y, s) are the Frenet-Serret coordinates; z is "scaled" time coordinate:

$$z = s - \beta c t\,, \tag{2}$$

where βc is the speed of the reference particle, and t is the time of flight of an arbitrary particle. The momenta canonically conjugated to x, y, z are the "scaled" momenta:

$$\begin{aligned} p_x &= P_x/P\,, \\ p_y &= P_y/P\,, \\ p_z &= \Delta P/P\,, \end{aligned} \tag{3}$$

where P_x and P_y are the usual canonical momenta, and ΔP is the deviation from the reference particle momentum P. The curvature $\rho(s)$ of the reference particle's trajectory is given by:

$$\rho(s) = \frac{P}{qB(s)}\,. \tag{4}$$

Note that this definition of ρ differs from the standard Frenet-Serret definition by a sign. The field index $n(s)$, evaluated around this reference trajectory, is given by:

$$n(s) = -\frac{\rho(s)}{B(s)}\left.\frac{\partial B(s)}{\partial x}\right|_{x=y=0}\,. \tag{5}$$

In practice, it would be more convenient to use θ as independent variable: this way we could ask TRANSOPTR to track for any number of turns. It is much harder to do that if you use s as independent variable (one does not know the orbit length *a priori*). Tracking with θ as independent variable is done by solving:

$$\frac{\mathrm{d}\boldsymbol{\sigma}}{\mathrm{d}\theta} = \frac{\mathrm{d}\boldsymbol{\sigma}}{\mathrm{d}s}\frac{\mathrm{d}s}{\mathrm{d}\theta} = \boldsymbol{F}\boldsymbol{\sigma} + \boldsymbol{\sigma}\boldsymbol{F}^{\mathrm{T}}\,, \tag{6}$$

where, since $\frac{\mathrm{d}s}{\mathrm{d}\theta} = \frac{r}{p_\theta}$, and $\boldsymbol{F}$ is given by:

$$\boldsymbol{F} = \frac{r}{p_\theta} \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{n(s)-1}{\rho(s)^2} & 0 & 0 & 0 & 0 & -\frac{1}{\rho(s)} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{n(s)}{\rho(s)^2} & 0 & 0 & 0 \\ \frac{1}{\rho(s)} & 0 & 0 & 0 & 0 & 1-\beta^2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} . \tag{7}$$

3 Reference Trajectory

To obtain coordinate r, p_θ, and t of the reference particle at every step in θ, we only need to numerically integrate the equations of motion that derive from Hagedoorn's Hamiltonian [4], namely:

$$\frac{\mathrm{d}r}{\mathrm{d}\theta} = r\frac{P_r}{P_\theta} , \tag{8}$$

$$\frac{\mathrm{d}P_r}{\mathrm{d}\theta} = P_\theta + qrB(r,\theta) , \tag{9}$$

$$\frac{\mathrm{d}t}{\mathrm{d}\theta} = \frac{\gamma m r}{P_\theta} , \tag{10}$$

or, using our "scaled" momenta:

$$\frac{\mathrm{d}r}{\mathrm{d}\theta} = r\frac{p_r}{p_\theta} , \tag{11}$$

$$\frac{\mathrm{d}p_r}{\mathrm{d}\theta} = p_\theta + \frac{r}{\rho(r,\theta)} , \tag{12}$$

$$\frac{\mathrm{d}t}{\mathrm{d}\theta} = \frac{r}{\beta c\, p_\theta} , \tag{13}$$

where $p_\theta = \sqrt{1-p_r^2}$.

4 Field Index

The last complication comes from the calculation of the partial derivative in the field index. Let's use the chain rule again:

$$\frac{\partial B}{\partial x} = \frac{\partial B}{\partial \theta}\frac{\partial \theta}{\partial x} + \frac{\partial B}{\partial r}\frac{\partial r}{\partial x} = \frac{\partial B}{\partial \theta}\frac{p_r}{r} - \frac{\partial B}{\partial r} p_\theta . \tag{14}$$

The field index is evaluated by injecting this equation into Eq. (5).

5 Benchmark against BEND

I tracked through a simple analytical field model:

$$B_z(r) = B_0 \left(\frac{r}{r_0}\right)^k , \tag{15}$$

with $r_0 = 1\,\text{m}$, $B_0 = B\rho/1\,\text{m}$, and $k = -0.1$. I chose the particle to be a 50 MeV electron, and I tracked for 360 degree (using the gfortran flag for double precision), and I got the following transfer matrix:

$$\begin{pmatrix} 0.948467 & -33.4015 & 0.00000 & 0.00000 & 0.00000 & 5.72584 \\ 0.300614\times10^{-2} & 0.948467 & 0.00000 & 0.00000 & 0.00000 & -0.334015 \\ 0.00000 & 0.00000 & -0.404216 & 289.242 & 0.00000 & 0.00000 \\ 0.00000 & 0.00000 & -0.289242\times10^{-2} & -0.404216 & 0.00000 & 0.00000 \\ 0.334015 & -5.72584 & 0.00000 & 0.00000 & 1.00000 & -735.180 \\ 0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 & 1.00000 \end{pmatrix}$$

And here is the matrix a get from a call to `call bend(r0,360.,+0.1,'bend')` (mind the sign difference $ak = -k$, due to the definition of the field index):

$$\begin{pmatrix} 0.948467 & -33.4015 & 0.00000 & 0.00000 & 0.00000 & 5.72584 \\ 0.300614\times10^{-2} & 0.948467 & 0.00000 & 0.00000 & 0.00000 & -0.334015 \\ 0.00000 & 0.00000 & -0.404216 & 289.242 & 0.00000 & 0.00000 \\ 0.00000 & 0.00000 & -0.289242\times10^{-2} & -0.404216 & 0.00000 & 0.00000 \\ 0.334015 & -5.72584 & 0.00000 & 0.00000 & 1.00000 & -735.180 \\ 0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 & 1.00000 \end{pmatrix}$$

Note that I obtained the exact same matrix in 'mode' 3 (canned routine) and 4 (numerical integration). Note also that without the '`--doublePrecision`' compiler flag differences appear on the 5^{th} significant digit.

6 Spiral Sector Scaling FFAG

As an intermediate 'debug' step toward my objective of tracking though a field map, I implemented the routine `FFA_SYMON` to track through the analytical field model of a spiral FFAG [7] with a simple cosine azimuthal field variation:

$$B_z(r) = B_0 \left(\frac{r}{r_0}\right)^k \left[1 + f\cos\left(N\theta - N\log\left(\frac{r}{r_0}\right)\tan(\zeta)\right)\right] . \tag{16}$$

I decided to test the code by calculating the tune for the case of the small electron ring we had worked on with Aurelia [5]. I tracked through 1 sector of the 5-sector lattice using

```
nsec = 5 !number of sectors
r0 = 30.0 !cm
b0 = -BRHO/r0*100.0 !T, since BRHO in stored in TRANSOPTR in T.m
ak = -0.1
f = 0.5
zeta = 65.0 !deg
call FFA_SYMON(ri,rpi,0.0,360./nsec,r0,b0,ak,nsec,f,zeta)
```

Now, to calculate a tune, I first had to place the reference particle right on the closed orbit. So I implemented a routine to do that, based on CYCLOPS algorithm [3], called `CLOSED_ORBIT`.

Let me write a few words on how to use the routine CLOSED_ORBIT. Let's look at the following sy.f example:

```
SUBROUTINE TSYSTEM
EXTERNAL ONE_SECTOR
r=30.
rp=0.
call CLOSED_ORBIT(r,rp,ONE_SECTOR)
return
end

SUBROUTINE ONE_SECTOR(ri,rpi)
COMMON/MOM/P,BRHO,pMASS,ENERGK,GSQ,ENERGKi,charge,current
nsec=5
r0=30.0 !cm
b0=-BRHO/r0*100.0 !T
ak=-0.1
f=0.5
zeta=65.0 !deg
call FFA_SYMON(ri,rpi,0.0,360./nsec,r0,b0,ak,nsec,f,zeta)
RETURN
END
```

After calling `CLOSED_ORBIT(r,rp,ONE_SECTOR)`, the parameters `r` and `rp` are right on the closed orbit (within t required accuracy controlled by the RK EPS parameters, specified in data.dat on line 2, column 4). Note that the 3$^{\rm rd}$ argument of the `CLOSED_ORBIT` is a subroutine: it defined the "lattice" though which the tracking is done. It is a wapping around one of the "FFA" routines, either `FFA_SYMON` or `POLAR_MAP`.

Once I had found the closed orbit, I tracked over 1 sector (i.e. 1/5 of a turn) and I got the following transfer matrix:

$$\begin{pmatrix} -1.05439 & 34.8717 & 0.00000 & 0.00000 & 0.00000 & 27.4213 \\ -0.794976\times10^{-1} & 1.68079 & 0.00000 & 0.00000 & 0.00000 & 1.87369 \\ 0.00000 & 0.00000 & 1.18408 & 25.9283 & 0.00000 & 0.00000 \\ 0.00000 & 0.00000 & -0.901261\times10^{-1} & -1.12899 & 0.00000 & 0.00000 \\ -0.204324 & -19.2492 & 0.00000 & 0.00000 & 1.00000 & -11.3227 \\ 0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.00000 & 1.00000 \end{pmatrix}$$

This leads to tunes of $\nu_r = 0.9965$ and $\nu_z = 1.228$, in full agreement with what is found in Aurelia's paper [5].

7 Tracking through TRIUMF CYclotron Field Map

I wrote the routine `POLAR_MAP` to load a 2-dimension polar field map and track through it. To evaluate the field and its partial derivative anywhere within the map, I use a bicubic spline interpolation from the fortran library FITPACK [2]. Note that the effect of space charge is taken into account in the same way than in all the other "SC" routines in TRANSOPTR (via a call to the routine `SC_KICK`).

I tested `POLAR_MAP` with the mid-plane field map of the TRIUMF 500 MeV cyclotron field map (angular step size: 1 deg., radial step size: 3 inch). To calculate tunes, and especially to get the integer part of the tune right, I implemented the robust algorithm proposed by Meade [6]. I ran TRANSOPTR in a loop from 1 to 500 MeV, in steps of 1 MeV, updating the common block `MOM` for every energy. If you want to see what the input file look like, take a look in the transopt r example/FFA on gitlab.triumf.ca. The resulting tune diagram as compared with CYCLOPS output is shown in Fig. 1. I do not understand the slight

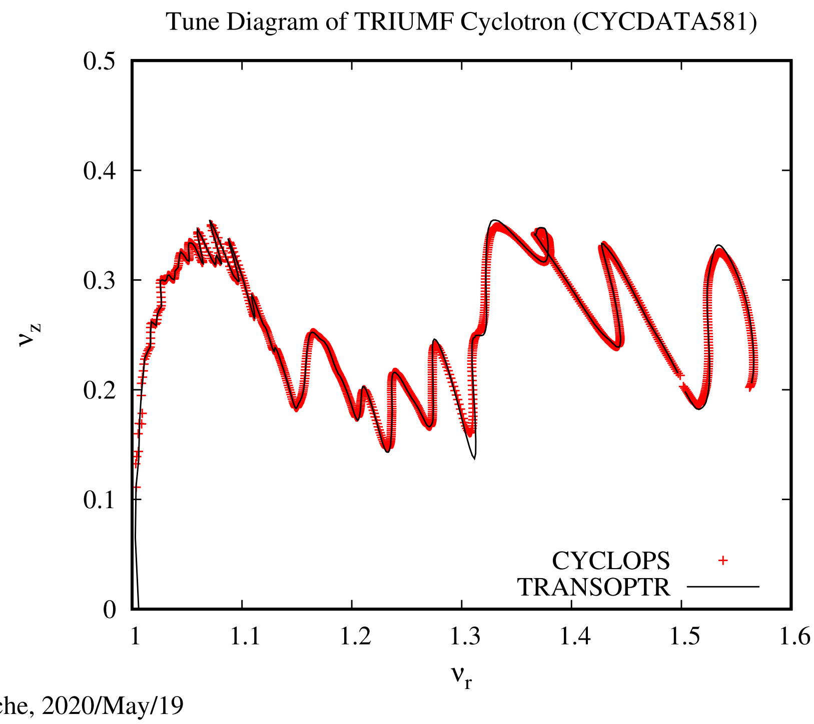

Figure 1: TRIUMF cyclotron tune diagram calculated by CYCLOPS, and TRANSOPTR..

difference between the two curves: is it coming from the fact that CYCLOPS precision is limited due to the constant step size of 1 deg.? Note also that the run time of TRANSOPTR is 2 to 3 orders of magnitude larger than CYCLOPS. Most of the computation times seems to be spent evaluating the bicubic spline. This could be improved by using a simpler and faster interpolation scheme, such as one using a kernel function. I am not interested in speed at the moment, I won't implement it.

References

[1] R. Baartman. Linearized Equations of Motion in Magnet with Median Plane Symmetry. Technical Report TRI-DN-05-06, TRIUMF, 2005.

[2] Paul Dierckx. *Curve and surface fitting with splines*. Oxford University Press, 1995.

[3] MM Gordon. Orbit properties of the isochronous cyclotron ring with radial sectors. *Annals of Physics*, 50(3):571–597, 1968.

[4] HL Hagedoorn and NF Verster. Orbits in an AVF cyclotron. *Nuclear Instruments and Methods*, 18:201–228, 1962.

[5] Aurelia Laxdal, Richard Baartman, Iouri Bylinskii, A Gottberg, FW Jones, P Kunz, T Planche, A Sen, S Ganesh, and L Lopera. Recirculating electron beam photo-converter

for rare isotope production. In *21st Int. Conf. on Cyclotrons and Their Applications (Cyclotrons' 16), Zurich, Switzerland, September 11-16, 2016*, pages 383–386. JACOW, Geneva, Switzerland, 2017.

[6] Colleen Meade. Changes to the main magnet code. Technical Report TRI-DN-70-54-Addendum4, TRIUMF, 1971.

[7] K. R. Symon, D. W. Kerst, L. W. Jones, L. J. Laslett, and K. M. Terwilliger. Fixed-field alternating-gradient particle accelerators. *Phys. Rev.*, 103:1837–1859, Sep 1956.