Chapter 21
Solutions

21.1 Solutions to Exercises of Chapter 3: Classical Cyclotron

Preliminaries

1. Keywords in zgoubi: by "keyword" it is meant, the name of the optical elements (such as DIPOLE, ELCYLDEF, MULTIPOLE, TOSCA, WIENFILTER, etc.), or I/O procedures (such as FAISCEAU, FAISTORE, IMAGE, etc.), or commands (such as FIT, SYSTEM, etc.), as they appear in a simulation input data file. Keywords are most of the time referred to without any additional explanation in the exercise solutions: details and explanations regarding the use and functioning of keywords are to be found in the users' guide.
2. It is strongly recommended, when setting up the input data files to work out the exercises, to have Zgoubi users' guide at hand. PART B of the guide in particular, details the formatting of the input data which follow any keyword, and their units (a few keywords only, for instance FAISCEAU, MARKER, YMY, do not require additional data). PART A is the "physics content" and details what keywords are doing and how. The users' guide INDEX is a convenient tool to navigate keywords. A complete list may also be found in the "Glossary of Keywords", at the begining of both PART A and PART B of the users' guide, and an overview of what they can be used at is given in "Optical elements versus keywords".

   - A concise notation KEYWORDS[ARGUMENT1, ARGUMENT2, ...] is used in the exercises and solutions: it follows the nomenclature of the Users' Guide, Part B. A couple of examples:
     – OBJET[KOBJ=1] stands for keyword OBJET, and the value of KOBJ=1 retained here;
     – OPTIONS[CONSTY=ON] stands for keyword OPTIONS, and the option retained here, CONSTY, switched ON.
   - The keyword INCLUDE is used in many simulation input data files. The goal is mostly to reduce the length of these files (which would otherwise be prohibitively voluminous, for a book). Just as with the Latex, or Fortran

"include" command, a segment of an optical sequence subject to an INCLUDE in some input data file, may always be replaced by that very sequence segment in plain.

3. Coordinate Systems: two sets of coordinate notations are used in the exercises,

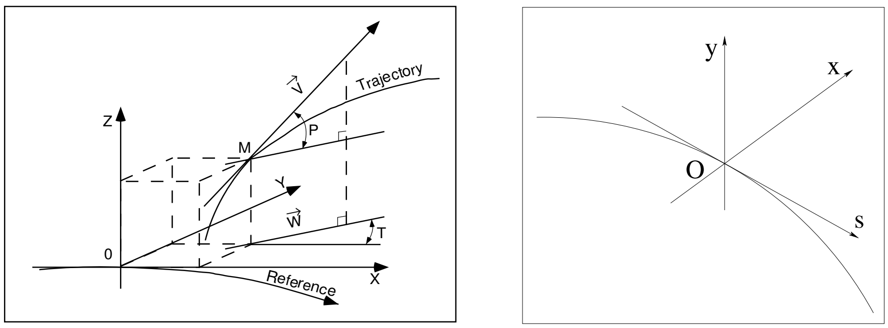

Fig. 21.1 Zgoubi Cartesian frame (O;X,Y,Z), and moving frame (O;s,x,y)

- on the one hand (and, in the Solutions Section mostly), zgoubi's (Y,T,Z,P,X,D) coordinates in the optical element reference frame (O;X,Y,Z), the very frame in which the optical element field $\mathbf{E}(X,Y,Z)$ and/or $\mathbf{B}(X,Y,Z)$ is defined (the origin for X depends on the optical element). Particle coordinates in this frame can be
  - either Cartesian, in which case X, Y, and Z denote the particle position in that frame, T and P the horizontal and vertical trajectory angles,
  - or cylindrical, in which case, given m the projection of particle position M in the Z=0 plane, Y denotes the radial coordinate: $Y = |\mathbf{Om}|$, whereas X denotes the polar angle **OX**-**Om** (as a matter of fact, the nature of the variables named X and Y in the source code does change), T is the horizontal trajectory angle with respect to the normal to **Om**, P is the vertical trajectory angle;

  Note: the sixth coordinate in zgoubi's set above is

$$D = \frac{\text{particle rigidity}}{BORO} = \frac{B\rho}{BORO}$$

  with BORO a reference rigidity, the very first numerical datum to appear in any zgoubi sequence, as part of the definition of initial particle coordinates by OBJET or MCOBJET. BORO may sometimes be denoted $B\rho_{\mathrm{ref}}$, depending upon the context. Note that D-1 identifies with the cordinate $\delta p/p$ below.
- on the other hand (and, in the exercise assignments mostly), the conventional (x,x',y,y',δl,$\delta p/p$) coordinates in the moving frame (O;s,x,y) or close variants.

Comments are introduced wherever deemed necessary (hopefully, often enough) in an effort to lift potential ambiguities regarding coordinate notations.

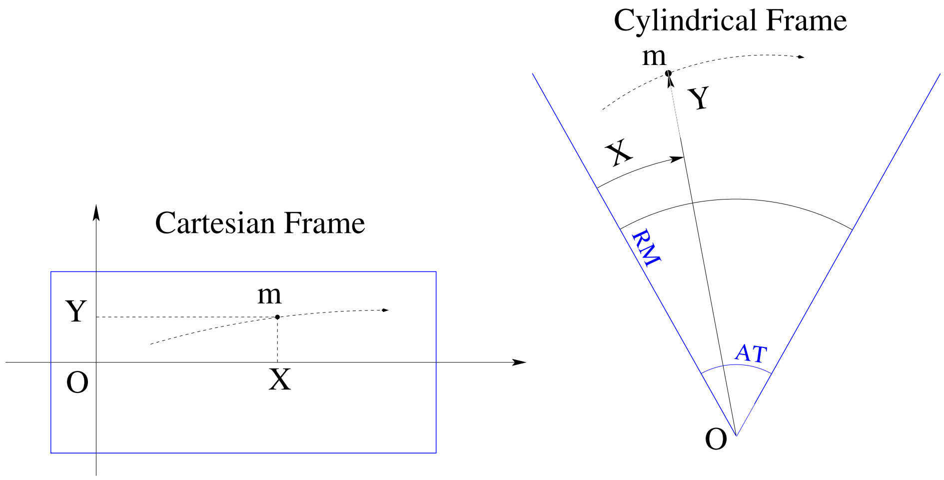

Fig. 21.2 Cartesian and cyclindrical reference frames in zgoubi. Let a particle location M(X,Y,Z) project at m(X,Y) (the dashed line figures the projected trajectory). In the case of an optical element (figured as a rectangular box) defined in Cartesian coordinates (case for instance of *MULTIPOL*, *BEND*, *TOSCA*[MOD $\leq$ 19]), X and Y in zgoubi.plt denote the coordinates taken along the reference frame axes. In the case of an optical element (figured as an angular sector AT with some reference radius RM) (case for instance of *DIPOLE*[*S*][-*M*], *TOSCA*[MOD $\geq$ 20]), X is the polar angle, counted positive clockwise, Y is the radius

4. Plots, in many cases, are obtained using gnuplot. Data are read from the columns in zgoubi.fai (resulting from FAISTORE or FAISCNL) or from zgoubi.plt (resulting from IL=2 option). The nature of the columns in these two files is summarized in their header, and detailed in [1, Sec. 8]. Computation data may also be logged in, and plotted from, additional ancillary files, such as zgoubi.CAVITE.out, zgoubi.MATRIX.out; the nature of the data columns in these files can be found out from their header and in [1].

3.1 Modeling a Cyclotron Dipole: Field Map

(a) A field map of a 180^o sector of a classical cyclotron magnet.

A Fortran program, geneSectorMap.f, given in Tab. 21.1 (it may be transposed to any other language), constructs the required map of a field distribution $B_Z(R,\theta)$. A polar mesh is retained (Fig. 21.3), rather than Cartesian, consistently with cyclotron magnet symmetry. The program can be compiled (*gfortran -o geneSectorMap geneSectorMap.f* will provide the executable, geneSectorMap) and run, as is. The field map is saved under the name geneSectorMap.out, excerpts are given in Tab. 21.2. That name appears under TOSCA in zgoubi input data file for this simulation (Tab. 21.3).

Note the following:

(i) the field map azimuthal extent (set at 180^o in geneSectorMap) can be changed, for instance to simulate a 60 deg sector instead.

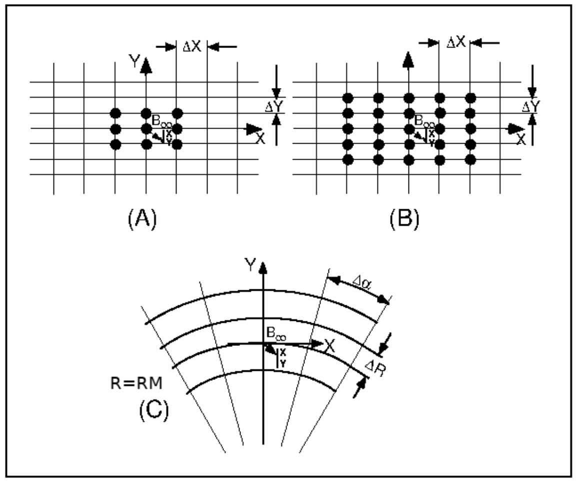

Fig. 21.3 Principle 2-D field map mesh as used by TOSCA, and the (O;X,Y) coordinate system. (A), (B): Cartesian mesh in the (X,Y) plane, case of respectively 9-point and a 25-point interpolation grid; the mesh increments are ΔX and ΔY; (C) : polar mesh and increments $\Delta\alpha$ and ΔR, as used here, and moving frame (O;X,Y) along a reference arc with radius RM. In all three cases the field at the location of the particle is calculated by interpolation from the 9 or 25 nodes closer to the particle.

Table 21.1 A Fortran program which generates a 180^o mid-plane field map. This angle as well as field amplitude can be changed, a field index can be added. This program can be compiled and run, as is. The field map it produces is logged in geneSectorMap.out

```
C geneSectorMap.f program
      implicit double precision (a-h,o-z)
      parameter (pi = 4.d0*atan(1.d0))
C------------ Hypothesis :
C Total angle extent of the field map. Can be changed, e.g., to 360, 0r 60 deg, or else.
      AT = 180.d0  /180.d0*pi
C Radial extent of the field map
      Rmi = 1.d0 ;  Rma = 76.d0                                                  ! cm.
C Take RM=50 cm as an intermediate radius, used to define the mesh:
      RM = 50.d0
C dR ("Delta R" in the main text) is the radial distance between two nodes, a reasonable value is
C (by experience) dR = 0.5 cm
      dR = 0.5d0 ; NR = NINT((Rma - Rmi) /dR)+1    ! CHANGE dR TO CHANGE NUMBER OF R NODES OF THE MESH.
C RdA=RM*dA is the arc length between two nodes along R=RM arc, given angle increment dA (dA is the
C "Delta theta" quantity in the main text).
C Take RdA a few mm, a reasonable value is (by experience):
      RdA = 0.5d0  ! cm                  ! CHANGE RdA TO CHANGE NUMBER OF AZIMUTHAL NODES OF THE MESH.
                                    ! AND CHANGE INTEGRATION STEP SIZE ACCORDINGLY IN zgoubi.dat FILE.
      NX= NINT(RM*AT /RdA) +1 ; RdA= RM*AT / DBLE(NX -1)  ! exact mesh step at RM, corresponding to NX.
      dA = RdA / RM ;A1 = 0.d0 ; A2 = AT                                 ! corresponding delta_angle.
C---------------------------------------------
      BZ = 5.d0 ; BY = 0.d0 ; BX = 0.d0 ; Z = 0.d0                                  ! Field in kG.
      open(unit=2,file='geneSectorMap.out')
      write(2,*) Rmi,dR,dA/pi*180.d0,dZ,
     >'     !  Rmi/cm, dR/cm, dA/deg, dZ/cm'
      write(2,*) '# Field map generated using geneSectorMap.f '
      write(2,fmt='(a)') '# AT/rd,  AT/deg, Rmi/cm, Rma/cm, RM/cm,'
     >//' NR, dR/cm, NX, RdA/cm, dA/rd : '
      write(2,fmt='(a,1p,5(e16.8,1x),2(i3,1x,e16.8,1x),e16.8)')
     >'# ',AT, AT/pi*180.d0,Rmi, Rma, RM, NR, dR, NX, RdA, dA
      write(2,*) '# For TOSCA: ',NX,NR,' 1 22.1 1.  !IZ=1 -> 2D ; '
     >//'MOD=22 -> polar map ; .MOD2=.1 -> one map file'
      write(2,*) '# R*cosA (A:0->360), Z==0, R*sinA, BY, BZ, BX '
      write(2,*) '#      Y                  Z              X '
     >//'            BY               BZ               BX   ix jr'
      write(2,*) '#      cm                 cm             cm '
     >//'            kG               kG               kG '
      do jr = 1, NR
        R = Rmi + dble(jr-1)*dR
        do ix = 1, NX
          A = A1 + dble(ix-1)*dA ; X = R * sin(A) ; Y = R * cos(A)
          write(2,fmt='(1p,6(e16.8),2(1x,i0))') Y,Z,X,BY,BZ,BX,ix,jr
        enddo
      enddo
      stop ' Job complete ! Field map stored in geneSectorMap.out.'
      end
```

Table 21.2 First and last few lines of the field map file geneSectorMap.out. The file starts with an 8-line header, the first of which is effectively used by zgoubi and indicates, in that order: the minimum radius of the map mesh Rmi, the radial increment dR, the azimuthal increment dA, the axial increment dZ (null and not used in the present case of a two-dimensional field map), in units of, respectively, cm, cm, degree, cm. The additional 7 lines give indications regarding numerical values used in, or resulting from, the execution of geneSectorMap.f. The first 5 numerical data in line 5 in particular are to be reported in zgoubi input data file under TOSCA keyword. The rest of the file is comprised of 8 columns, the first three give the node coordinates and the next three the field component values at that node, the last two columns are the (azimuthal, radial) node numbering, from (1,1) to (315,151) in the present case

```
   1.   0.5   0.57324840764331209        0.       !  Rmi/cm, dR/cm, dA/deg, dZ/cm
# Field map generated using geneSectorMap.f
# AT/rd,  AT/deg, Rmi/cm, Rma/cm, RM/cm, NR, dR/cm, NX, RdA/cm, dA/rd :
# 3.14159265E+00   1.80E+02   1.E+00   7.60E+01   5.00E+01 151   5.00E-01 315  5.00253607E-01   1.00050721E-02
# For TOSCA:          315         151  1 22.1 1.  !IZ=1 -> 2D ; MOD=22 -> polar map ; .MOD2=.1 -> one map file
# R*cosA (A:0->360), Z==0, R*sinA, BY, BZ, BX
#    Y                  Z                X               BY                BZ                BX     ix jr
#    cm                 cm               cm              kG                kG                kG
  1.00000000E+00  0.00000000E+00  0.00000000E+00  0.00000000E+00  5.00000000E+00  0.00000000E+00 1 1
  9.99949950E-01  0.00000000E+00  1.00049052E-02  0.00000000E+00  5.00000000E+00  0.00000000E+00 2 1
  9.99799804E-01  0.00000000E+00  2.00088090E-02  0.00000000E+00  5.00000000E+00  0.00000000E+00 3 1
  9.99549577E-01  0.00000000E+00  3.00107098E-02  0.00000000E+00  5.00000000E+00  0.00000000E+00 4 1
  9.99199295E-01  0.00000000E+00  4.00096065E-02  0.00000000E+00  5.00000000E+00  0.00000000E+00 5 1
  ................................
  ................................
  ................................
 -7.59391464E+01  0.00000000E+00  3.04073010E+00  0.00000000E+00  5.00000000E+00  0.00000000E+00 311 151
 -7.59657679E+01  0.00000000E+00  2.28081394E+00  0.00000000E+00  5.00000000E+00  0.00000000E+00 312 151
 -7.59847851E+01  0.00000000E+00  1.52066948E+00  0.00000000E+00  5.00000000E+00  0.00000000E+00 313 151
 -7.59961962E+01  0.00000000E+00  7.60372797E-01  0.00000000E+00  5.00000000E+00  0.00000000E+00 314 151
 -7.60000000E+01  0.00000000E+00  9.30731567E-15  0.00000000E+00  5.00000000E+00  0.00000000E+00 315 151
```

A gnuplot script to obtain Fig. 21.4:

```
set key maxcol 1 ; set key t l ; set xtics mirror ; set ytics mirror
set xlabel "X [m]" ; set ylabel "Y [m]"  ; set zrange [:5.15] ; cm2m = 0.01
splot "geneSectorMap.out" u ($1 *cm2m):($3 *cm2m):($5) w l lc rgb "red" notit ; pause 1
```

(ii) the field is purely vertical being the mid-plane field of a mid-plane symmetry dipole magnet. The field is taken constant in this exercise, the same value ∀R, $\forall\theta$ throughout the map mesh, whereas in upcoming exercises, a *focusing index* will be introduced, which will make $B_Z \equiv B_Z(R)$ an R-dependent quantity (and beyond, Thomas focusing and the isochronous cyclotron make $B_Z \equiv B_Z(R,\theta)$ an R- and θ-dependent quantity).

As an indication, the top and bottom parts of the field map file generated by geneSectorMap, in the proper format for TOSCA keyword to be able to read it, are given in Tab. 21.2. Figure 21.4 shows the field over the 180^o azimuthal extent (using a gnuplot script, bottom of Tab. 21.1)

This field map can be readily tested using the example of Tab. 21.3, which raytraces $E_k = 120 \times 10^3$ eV and $E_k = 5.52 \times 10^6$ eV protons on circular trajectories centered at the center of the field map. Trajectory radii, respectively $R = 10.011$ cm and $R = 67.998$ cm (Tab. 21.3), have been prior determined from

$$\text{Rigidity}\ \ B\rho = B_0 \times R \quad \text{and} \quad B\rho = p/c = \sqrt{E_k(E_k + 2\,M)}/c \tag{21.1}$$

with $B_0 = 0.5$ T (Tab. 21.1), $M = 938.272 \times 10^6$ eV/c^2 the proton mass.

The optical sequence for this particle raytracing uses

Table 21.3 Simulation input data file: it is set to allow a preliminary test regarding the field map geneSectorMap.out (as produced by the Fortran program geneSectorMap, Tab. 21.1), by computing two circular trajectories centered on the center of the map. This file also defines the INCLUDE segment between the labels (LABEL1 type [1, Sec. 7.7]) #S_halfDipole and #E_halfDipole, used in subsequent exercises. This input data file is used under the name FieldMapSector.inc in subsequent exercises

```
Uniform field sector. FieldMapSector.inc.
'MARKER'   FieldMapSector_S                                                  ! Just for edition purposes.
'OBJET'
64.62444403717985                      ! Reference Brho ("BORO" in the users' guide) -> 200keV proton.
2
2 1
10.011362 0. 0. 0. 0. 0.7745802 'o'           ! p[MeV/c]= 15.007, Brho[kG.cm]= 50.057, kin-E[MeV]=0.12.
67.997983 0. 0. 0. 0. 5.2610112 'o'           ! p[MeV/c]=101.926, Brho[kG.cm]=339.990, kin-E[MeV]=5.52.
1 1
'MARKER'   #S_halfDipole
'TOSCA'
0 2     ! IL=2 to log step-by-step coordinates, spin, etc., in zgoubi.plt (avoid, if CPU time matters).
1. 1. 1. 1.        ! Normalization coefficients, for B, X, Y and Z coordinate values read from the map.
HEADER_8                                               ! The field map file starts with an 8-line header.
315 151 1 22.1 1.             ! IZ=1 for 2D map; MOD=22 for polar frame; .MOD2=.1 if only one map file.
geneSectorMap.out
0 0 0 0        ! Possible vertical boundaries within the field map, to start/stop stepwise integration.
2
1. ! cm                                                                          ! Integration step size.
2                                                                           ! Magnet positionning option.
0. 0. 0. 0.                                                                        ! Magnet positionning.
'MARKER'   #E_halfDipole
'FAISCEAU'                                             ! Local particle coordinates logged in zgoubi.res.
'SYSTEM'                        ! This SYSTEM command runs gnuplot, for a graph of the two trajectories.
1
gnuplot <./gnuplot_Zplt.gnu
'MARKER'   FieldMapSector_E                                                  ! Just for edition purposes.
'END'
```

A gnuplot script to obtain Fig. 21.4:

```
# gnuplot_Zplt.gnu
set key maxcol 1 ; set key t r ; set xtics ;  set ytics
set xlabel "X_{Lab}  [m]" ; set ylabel "Y_{Lab} [m]"  ; cm2m = 0.01; set size ratio 1  ;  set polar
plot for [i=1:3] "zgoubi.plt" u ($19==i ? $22 :1/0):($10 *cm2m) w l lw 2 lc rgb "black"  notit ; pause 1
```

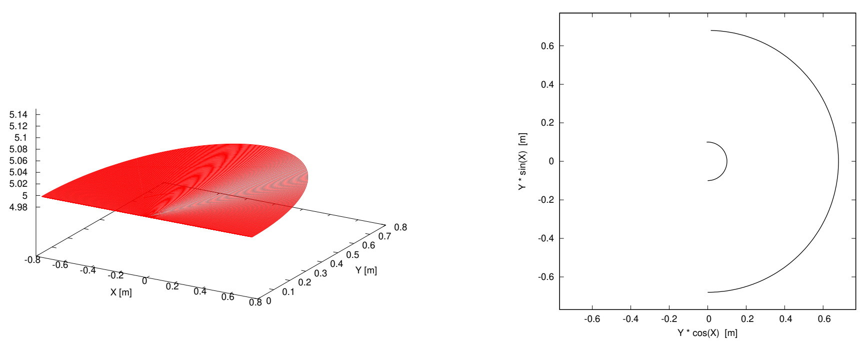

Fig. 21.4 Left: map of a constant magnetic field over a 180 deg sector, 76 cm radial extent. Right: two circular trajectories, at respectively 0.12 and 5.52 MeV, computed using that field map

(i) OBJET to define a (arbitrary) reference rigidity and initial particle coordinates

(ii) TOSCA [1, pp. 169 & 309] to read the field map and raytrace through (and TOSCA's 'IL=2' flag to store step-by-step particle data into zgoubi.plt)

(iii) FAISCEAU to print out particle coordinates in zgoubi.res

(iv) SYSTEM to run a gnuplot script (Tab. 21.26) once raytracing is complete

(v) MARKER, to define two particular "LABEL_1" type labels [1] (#S_halfDipole and #E_halfDipole), to be used with INCLUDE in subsequent exercises

Two circular trajectories in a dee, resulting from the data file of Tab. 21.3 are shown in Fig. 21.4. Inspecting zgoubi.res one finds the D, Y, T, Z, P, S particle coordinates, from FAISCEAU (Tab. 21.3), at OBJET (left) and current (right) after a turn in the cyclotron (they equal as the trajectory is closed):

```
      6  Keyword, label(s) :  FAISCEAU                                                                   IPASS= 1
                                              TRACE DU FAISCEAU
                                            (follows element #      5)
                                                   2 TRAJECTOIRES
                                     OBJET                                                FAISCEAU
          D       Y(cm)     T(mr)     Z(cm)     P(mr)      S(cm)      D-1     Y(cm)    T(mr)    Z(cm)    P(mr)     S(cm)
o  1   0.7746    10.011    0.000     0.000     0.000     0.0000   -0.2254   10.011   -0.000    0.000    0.000   3.145152E+01     1
o  1   5.2610    67.998    0.000     0.000     0.000     0.0000    4.2610   67.998   -0.000    0.000    0.000   2.136220E+02     2
```

Table 21.4 Simulation input data file: stage 1, optical sequence to find closed orbits at a series of different momenta. An INCLUDE inserts the #S_halfDipole to #E_halfDipole TOSCA segment of the sequence of Tab. 21.3

```
Uniform field 180 deg. sector. Find orbits.
'MARKER'   FieldMapStage1_S                                              ! Just for edition purposes.
 'OBJET'
64.62444403717985                       ! Reference Brho ("BORO" in the users' guide) -> 200keV proton.
2
1 1
12.9248888074 0.  0. 0. 0.  1. 'm'                 ! This initial radius yields BR=64.6244440372 kG.cm.
1
'INCLUDE'
1
FieldMapSector.inc[#S_halfDipole:#E_halfDipole]
'FAISCEAU'
 'INCLUDE'
1
FieldMapSector.inc[#S_halfDipole:#E_halfDipole]
'FIT'
1
2  35  0  6.                            ! Vary momentum, to allow fulfilling the following constraint:
1
3.1 1 2 5 0. 1. 0            ! request same radius after a half-turn (i.e., after first 180 deg sector,
!                                                     this ensures centering of orbit on center of map).
'FAISCEAU' CHECK   ! Allows quick check of particle coordinates, in zgoubi.res: final should = initial.
'FAISTORE'
initialRs.fai                                                        ! Log coordinates in initialRs.fai.
1
'REBELOTE'                                                                 ! Repeat what precedes,
15  0.1  0 1                                                                           ! 15 times.
1
OBJET 30  10:80           ! Change the value of parameter 30 (namely, Y) in OBJET (prior to repeating).
'SYSTEM'
2
gnuplot <./gnuplot_Zplt.gnu
cp gnuplot_Zplt_XYLab.eps gnuplot_Zplt_XYLab_stage1.eps
'MARKER'   FieldMapStage1_E                                              ! Just for edition purposes.
'END'
```

(b) Concentric trajectories in the median plane.

The optical sequence for this exercise is given in Tab. 21.4. Compared to the previous sequence (Tab. 21.3), (i) the TOSCA segment has been replaced by an

Table 21.5 Simulation input data file: stage 2, optical sequence to raytrace particles on closed orbits

```
Uniform field sector
'MARKER'   FieldMapStage2_S                                          ! Just for edition purposes.
'OBJET'
64.62444403717985                    ! Reference Brho ("BORO" in the users' guide) -> 200keV proton.
3
1 999 1
1 999 1
1. 1. 1. 1. 1. 1. 1. '*'
0.  0.  0.  0.  0.  0.  0.
0
initialRs.fai
'INCLUDE'                                                        ! Inset a 180 deg sector field map.
1
FieldMapSector.inc[#S_halfDipole:#E_halfDipole]
'FAISCEAU'                                          ! Local particle coordinates logged in zgoubi.res.
'INCLUDE'                                                        ! Inset a 180 deg sector field map.
1
FieldMapSector.inc[#S_halfDipole:#E_halfDipole]
'FAISCEAU'  #End
'SYSTEM'
2
gnuplot < gnuplot_Zplt.gnu
okular gnuplot_Zplt_XYLab.eps &
'MARKER'   FieldMapStage2_E                                          ! Just for edition purposes.
'END'
```

A gnuplot script to obtain Fig. 21.5:
This script applies to both stage 1 and stage 2 parts of the exercise (respectively Tab. 21.4 and Tab. 21.5)

```
# gnuplot_Zplt.gnu
set key maxcol 1 ; set key t r ; set xtics ; set ytics  ; set size ratio 1 ; set polar
set xlabel "X_{Lab}  [m] \n" ; set ylabel "Y_{Lab}  [m] \n" ; cm2m = 0.01 ; sector1 = 5 ; sector2 =9 ; pi = 4.*atan(1.)
plot "zgoubi.plt" u ($42==sector1 ? $22    :1/0):($10 *cm2m) w p ps .3 lc rgb "red"  tit "from 1st 180deg map" ,\
     "zgoubi.plt" u ($42==sector2 ? $22+pi :1/0):($10 *cm2m) w p ps .2 lc rgb "blue" tit "from 2nd 180deg map"
pause 1
```

INCLUDE, for the mere interest of making the input data file for this simulation shorter, and (ii) additional keywords are introduced, including

- FIT, which finds the closed orbit (the closed circle) for a particular momentum,

- FAISTORE to print out particle data, in initialRs.fai here, at the “afterFIT” label 1 location, once FIT is completed,

- REBELOTE, which repeats the execution of the sequence (REBELOTE sends the execution pointer back to the top of the data file) for a new momentum value which it defines itself.

In order to compute and then plot trajectories, proceed in two stages (Fig. 21.5):

- first stage: closed circles for a series of different radii taken in $[10, 80]$ cm are searched, using FIT to find the appropriate momenta. REBELOTE is used to repeat with a series of different values of R (REBELOTE modifies the initial particle coordinate Y_0 in OBJET). Particle coordinates *after the FIT procedure* are logged in initialRs.fai, by FAISTORE. The input data file for this simulation is given in Tab. 21.4,

- second stage: these particles are raytraced using OBJET[KOBJ=3] which reads initial coordinates from initialRs.fai. The input data file for this simulation is given in Tab. 21.5.

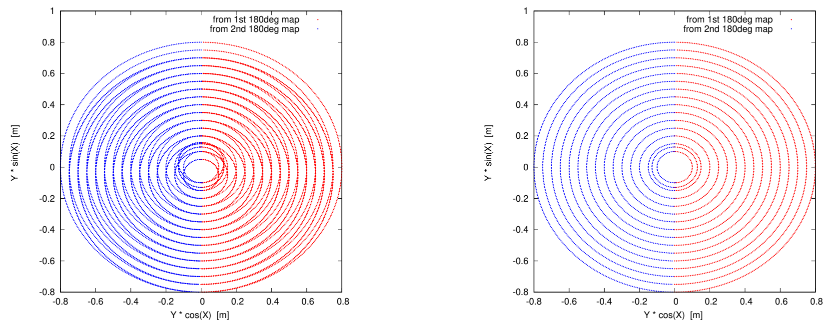

Fig. 21.5 Left, stage 1: circular trajectories go by pair: before FIT (R has been fixed by REBELOTE, but the momentum in OBJET is still that of the previous particle), and after FIT (the proper momentum value has been found by FIT, consistent with R). Right, stage 2: proper circular trajectories, centered on the field map center. The outermost orbit is at R=80 cm by hypothesis, thus $BR = B_0 \times R = 0.4\,\mathrm{T\,m}$, $E_k = 7.632\,\mathrm{MeV}$. These stepwise $(R\theta)$ data are read from zgoubi.plt, coordinates (Y,X) in zgoubi polar frame nomenclature

At the bottom of zgoubi input data file, the SYSTEM command provides plots of particle trajectories, executing a gnuplot script given in Tab. 21.5. Results are given in Fig. 21.5.

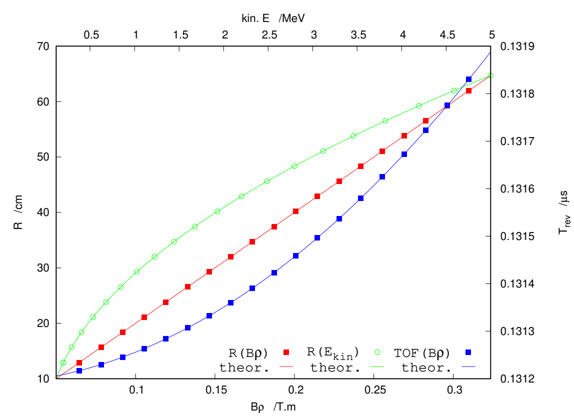

Fig. 21.6 Numerical (markers) and theoretical (solid lines) values of orbit radius, R, and revolution period, T_{rev}, as a function of kinetic energy E_k and rigidity BR. The mesh density here is $N_\theta \times N_R = 315 \times 151$. The integration step size is $\Delta s = 1\,\mathrm{cm}$, so ensuring converged results (to $\Delta R/R$ and $\Delta T_{\mathrm{rev}}/T_{\mathrm{rev}} < 10^{-6}$)

The reason why it is possible to push the raytracing beyond the 76 cm radius field map extent, without loss of accuracy, is that the field is constant. Thus, referring to the polynomial interpolation technique used [1, Sec. 1.4], the extrapolation out of the map will leave the field value unchanged.

(c) Energy and rigidity dependence of orbit radius and time-of-flight.

The orbit radius R and the revolution time T_{rev} as a function of kinetic energy E_k and rigidity BR are obtained by a similar scan to exercise (b). The results are shown in Fig. 21.6.

A slow increase of revolution period with energy can be observed, which is due to the mass increase.

Note that these results are converged for the step size, to high accuracy (see (d)), due to its value taken small enough, namely $\Delta s = 1$ cm. This corresponds for instance to 80 steps to complete a revolution for the 120 keV, $R = 12.9$ cm smaller radius trajectory in Fig 21.5.

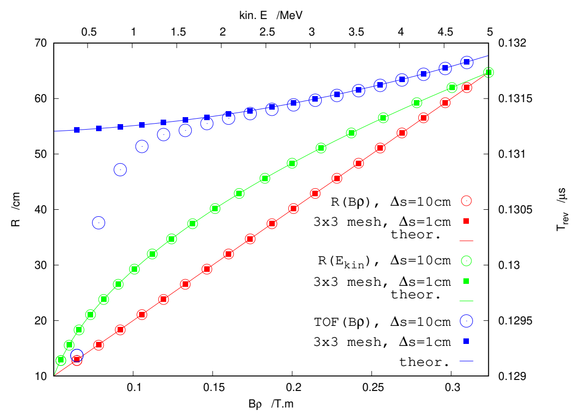

Fig. 21.7 Orbit radius R (left axis), and revolution period, T_{rev} (right axis), as a function of kinetic energy E_k (top scale) and rigidity BR (bottom scale). From numerical integration: small solid markers are for $\Delta s = 1$ cm and 3×3 node mesh, large empty circles for $\Delta s = 10$ cm and 106×151 node mesh. From theory: solid lines. The 3×3 mesh numerical case coincides with theory, thus showing the absence of any effect of mesh density

(d) Numerical convergence: mesh density.

This question concerns the dependence of the numerical convergence of the solution of the differential equation of motion upon mesh density.

The program used in (b) to generate a field map (Tab. 21.1) is modified to construct field maps of $B_Z(R, \theta)$ with various radial and azimuthal mesh densities. Changing these is simply a matter of modifying the quantities dR (radius increment ΔR) and $R\,dA$ (R times the azimuth increment $\Delta\theta$) in the program of Tab. 21.1. The field maps geneSectorMap.out so generated for various (dR, RdA) couples may be saved under different names, and used separately.

Table. 21.6 shows the top and bottom parts of the TOSCA field map, in the case of a 60^o sector covered in $N_\theta \times N_R = \dfrac{60^o}{\Delta\theta} \times \dfrac{75\,\text{cm}}{\Delta R} = \dfrac{360^o}{120^o} \times \dfrac{75\,\text{cm}}{37.5\text{cm}} = 3 \times 3$ nodes. Six sectors are now required to cover the complete cyclotron dipole: zgoubi input data need be changed accordingly, namely stating TOSCA - possibly via an INCLUDE - six times, instead of just twice in the case of a 180 degree sector.

The result to be expected: with a mesh reduced to as low as $N_\theta \times N_R = 3 \times 3$, compared to $N_\theta \times N_R = 106 \times 151$, radius and time-of-flight should however remain unchanged. This shows in Fig. 21.7 which displays both cases, over a E_k : $0.12 \rightarrow 5$ MeV energy span (assuming protons). The reason for the absence of effect of the

Table 21.6 Top section (commencing with a header) of the field map of a 60^o constant field sector as read by TOSCA (TOSCA input data list is shown at the bottom of the Table). The field map is complete, with smallest possible $NX \times NR = 3 \times 3 = 9$ number of nodes. The first line of the header is read and used by zgoubi, namely, the minimum value of the radius in the map, radius increment, azimuthal increment, and vertical increment (null here, are this is a 2-dimensional map). The next 7 lines are just comments

```
1.000       37.500     30.00     0.0         !  Rmi/cm, dR/cm, dA/deg, dZ/cm
# Field map generated using geneSectorMap.f
# AT/rd,       AT/deg,  Rmi/cm, Rma/cm,   RM/cm, NR, dR/cm, NX, RdA/cm,       dA/rd :
# 1.04719755   60.       1.      76.       50.    3   37.5   3   26.1799388   0.523598776E-01
# For TOSCA:  3   3  1 22.1 1.  !IZ=1 -> 2D ; MOD=22 -> polar map ; .MOD2=.1 -> one map file
# R*cosA (A:0->360), Z==0, R*sinA, BY, BZ, BX
# cm                 cm    cm      kG  kG  kG
#
  1.00000000E+00  0.00000000E+00  0.00000000E+00  0.00000000E+00  5.00000000E+00  0.00000000E+00 1 1
  8.66025404E-01  0.00000000E+00  5.00000000E-01  0.00000000E+00  5.00000000E+00  0.00000000E+00 2 1
  5.00000000E-01  0.00000000E+00  8.66025404E-01  0.00000000E+00  5.00000000E+00  0.00000000E+00 3 1
  3.85000000E+01  0.00000000E+00  0.00000000E+00  0.00000000E+00  5.00000000E+00  0.00000000E+00 1 2
  3.33419780E+01  0.00000000E+00  1.92500000E+01  0.00000000E+00  5.00000000E+00  0.00000000E+00 2 2
  1.92500000E+01  0.00000000E+00  3.33419780E+01  0.00000000E+00  5.00000000E+00  0.00000000E+00 3 2
  7.60000000E+01  0.00000000E+00  0.00000000E+00  0.00000000E+00  5.00000000E+00  0.00000000E+00 1 3
  6.58179307E+01  0.00000000E+00  3.80000000E+01  0.00000000E+00  5.00000000E+00  0.00000000E+00 2 3
  3.80000000E+01  0.00000000E+00  6.58179307E+01  0.00000000E+00  5.00000000E+00  0.00000000E+00 3 3
  ................................
```

Modified TOSCA keyword data, in the case of a 60^o *sector field map* (*compared to Tab.* 21.3*, the sole data line* "3 3 1 22.1 1." *changes, from* "315 151 1 22.1 1." *in that earlier* 180^o *sector case*):

```
'TOSCA'
0 2    ! IL=2: log step-by-step coordinates, spin, etc., in zgoubi.plt (to avoid if CPU time matters).
1. 1. 1. 1.       ! Normalization coefficients, for B, X, Y and Z coordinate values read from the map.
HEADER_8                                         ! The field map file starts with an 8-line header.
3   3   1 22.1 1.            ! IZ=1 for 2D map; MOD=22 for polar frame; .MOD2=.1 if only one map file.
geneSectorMap.out
0 0 0 0       ! Possible vertical boundaries within the field map, to start/stop stepwise integration.
2
1. ! cm                                                                        ! Integration step size.
2                                                                          ! Magnet positionning option.
0. 0. 0. 0.                                                                    ! Magnet positionning.
```

mesh density is that the field is constant. As a consequence the field derivatives in the Taylor series based numerical integrator are all zero [1, Sec. 1.2]: only B_Z is left in evaluating the Taylor series, however B_Z is constant. Thus R remains unchanged when pushing the particle by a step Δs, and the cumulated path length - the closed orbit length - and revolution time - path length over velocity - end up unchanged. Note: this will no longer be the case when a radial field index is introduced in order to cause vertical focusing, in subsequent exercises.

(e) Numerical convergence: integration step size

This question concerns the dependence of the numerical convergence of the solution of the differential equation of motion upon integration step size.

A 106×151 node mesh is used here (as in Tab. 21.3) which ensures proper convergence of the integration relative to mesh density.

Figure 21.7 displays two cases of step sizes, $\Delta s \approx 1$ cm (as in Fig. 21.6, small enough that the numerical integration is converged) and $\Delta s = 10$ cm. The difference on R between the two values is weak, and only sensed (at the scale of the graph) for smaller R values where the number of steps over one revolution goes as low as $2\pi R/\Delta s \approx 2\pi \times 14.5/10 = 9$ steps. The change in time-of-flight due to the larger step size amounts to a relative 10^{-3}.

Step size is critical in the numerical integration, the reason is that the coefficients of the Taylor series that yield the new position vector $\mathbf{R}(M_1)$ and the new velocity vector $\mathbf{v}(M_1)$, from an initial location M_0 after a Δs push, change when the step size changes [1, Sec. 1.2]: the coefficients of the Taylor series are the derivatives of the velocity vector, which take substantial values, especially at small radius R. Thus, taking too large a ΔS value makes the high order terms significant and the Taylor series truncation is fatal to the accuracy (regardless of possible additional issue of radius of convergence of the series).

(f) Numerical convergence: $\dfrac{\delta R}{R}(\Delta s)$

The increase of $\delta R(\Delta s)/R$ at large Δs has been explained above. The increase of $\delta R(\Delta s)/R$ at very small Δs is due to computer accuracy: truncation of numerical values at a limited number of digits causes one or more Δs steps to have no effect on the change of position and velocity vector of the particle, until these steps sum up to some threshold determined by the truncation.

3.2 Modeling a Cyclotron Dipole: Analytical

This exercise introduces to the analytical modeling of a dipole, using DIPOLE here, and compares to the field map model used to solve exercise 3.1. The exercise is not entirely solved, however all the material needed for that is provided, and indications are given to complete it.

(a) Analytical modeling.

DIPOLE keyword provides an analytical model of the field to simulate a sector dipole with index (in lieu of TOSCA which reads and tracks through a field map, Tab. 21.3). The field model in DIPOLE is [1]

$$B_Z = \mathcal{F}(\theta) B_0 \left[1 + k \left(\frac{R - R_0}{R_0} \right) + k' \left(\frac{R - R_0}{R_0} \right)^2 + k'' \left(\frac{R - R_0}{R_0} \right)^3 \right] \tag{21.2}$$

R_0 is a reference radius, $B_0 = B_Z(R_0)|_{\mathcal{F}\equiv 1}$ is a reference field value, k is the field index and k', k" are homogeneous to its first and second derivative with respect to R (Eq. 3.12). $\mathcal{F}(\theta)$ is an azimuthal form factor, defined by the fringe field model, presumably taking the value 1 in the body of the dipole. In the present case a hard-edge field model is considered, so that

$$\mathcal{F} = \begin{cases} 1 & \text{inside} \\ 0 & \text{outside} \end{cases} \text{ the dipole magnet} \tag{21.3}$$

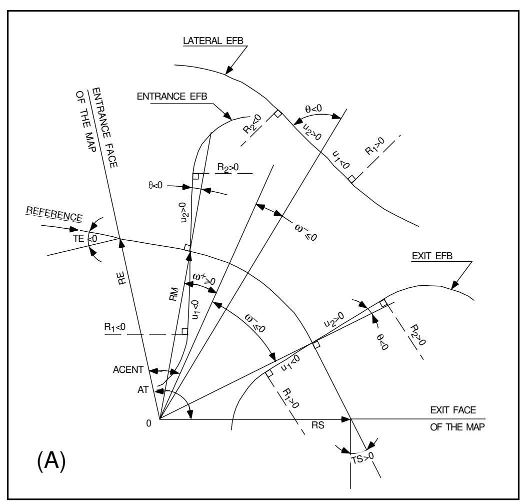

Fig. 21.8 Parameters used to define the geometry of a dipole magnet with index, using DIPOLE [1, pp. 130 & 232]. In the text, ACENT is noted ACN

Setting up the input data list under DIPOLE (Table 21.7) requires close inspection of Fig. 21.8, which details the geometrical parameters such as the full angular opening of the field region that DIPOLE comprises, AT; a reference angle ACN to allow positioning the effective field boundaries at ω^+ and ω^-; field and indices; fringe field regions at $ACN - \omega^+$ (entrance) and $AT - ACN + \omega^-$ (exit); wedge angles, etc.

A 60 deg sector is used here rather than a 180 deg one, it is detailed in Table 21.7 (which however also provides the definition of a 180 deg sector, for possible use and comparisons with a three-60 deg sector assembly).

In setting up DIPOLE data the following values have been accounted for:

- $R_0 = 50$ cm, an arbitrary value (yet, consistent with other exercises), more or less half the dipole extent

- $B_0 = B_Z(R_0) = 5$ kG, as in the previous exercise. Note in passing, $R_0 = 50$ cm thus corresponds to $BR = 0.25$ T m, $E_k = 2.988575$ MeV proton kinetic energy,

- $k = 0$ for the time being (constant field at all (R, θ)),

- a hard-edge field model for $\mathcal{F}$ (Eq. 21.3). In that manner for instance, two 60 deg sectors connect without any field discontinuity to form a 120 deg sector.

A graph of $\mathrm{B_Z(R,\theta)}$ can be produced by computing constant radius orbits, for a series of energies ranging in $0.12 - 5.52$ MeV for instance. DIPOLE[IL=2] causes logging of step by step particle data in zgoubi.plt, including particle position and magnetic field vector; these data can be read and plotted, to yield similar results to Fig. 21.4.

(b) Concentric trajectories in the median plane.

The optical sequence of the previous exercise, Tab. 21.4, can be used, by just changing the INCLUDE to account for a 180^o DIPOLE (instead of TOSCA), namely

```
'INCLUDE'
1
3* 60degSector.inc[#S_60degSectorUnifB:#E_60degSectorUnifB]
```

wherein 60degSector.inc is the name of the data file of Tab. 21.7 and

[#S_60degSectorUnifB:#E_60degSectorUnifB]

is the DIPOLE segment as defined in the latter. Note that the segment represents a 60^o DIPOLE, thus it is included 3 times.

The additional keywords in that modified Tab. 21.4file include

- FIT, which finds the closed orbit (the closed circle) for a particular momentum,

- FAISTORE to print out particle data, in initialRs.fai here, at the “afterFIT” label 1 location, once FIT is completed,

- REBELOTE, which repeats the execution of the sequence (REBELOTE sends the execution pointer back to the top of the data file) for a new momentum value which it defines itself.

For the rest, follow the two-stage procedure of exercise 3.1-b. The results are the same, Fig. 21.5.

(c) Energy and rigidity dependence of orbit radius and time-of-flight.

The orbit radius R and the revolution time T_{rev} as a function of kinetic energy E_k and rigidity BR are obtained by a similar scan to exercise (b). The procedure is the same as in exercise 3.1-c. Results are expected to be the same as well (as in Fig. 21.6).

A comparison of revolution periods can be made using the simulation file of Table 21.7 which happens to be set for a momentum scan and yields Fig. 21.9, to be compared to Fig. 21.6: DIPOLE and TOSCA produce the same results as long as both methods are converged, from the integration step size stand point (small enough), and regarding TOSCA from field map mesh density stand point in addition (dense enough).

(d) Numerical convergence: integration step size; $\frac{\delta R}{R}(\Delta s)$.

This question concerns the dependence of the numerical convergence of the solution of the differential equation of motion upon integration step size.

Follow the procedure of exercise 3.1-e, to obtain a similar outcomes to Fig. 21.7 (ignoring mesh density cases in that graph, in the present case of the analytical modeling with DIPOLE).

The $\frac{\delta R}{R}$ dependence upon the integration step size Δs is commented in exercise 3.1-e and holds regardless of the field modeling method (field map or analytical model).

(e) Pros and cons.

Using a field map is a convenient way to account for complicated one-, two- or three-dimensional field distributions.

However, using an analytical field model rather, ensures greater accuracy of the integration method.

CPU-time wise, one or the other method may be faster, depending on the length of the optical sequence to be raytraced, on the number of particles to be raytraced, the number of turns (iterations by REBELOTE).

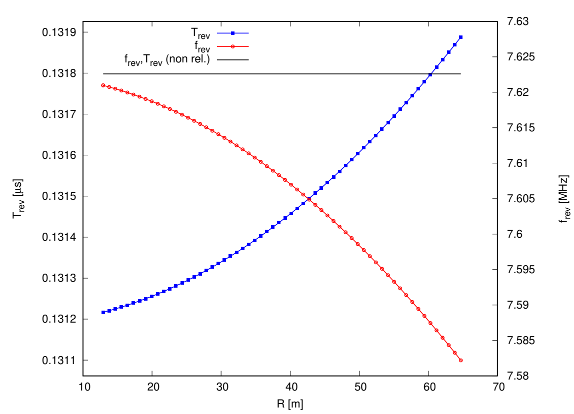

Fig. 21.9 A scan of radius-dependent revolution frequency. An analytical model of a cyclotron dipole is used, featuring uniform field (no radial gradient, at this point)

Table 21.7 Simulation input data file: analytical modeling of a dipole magnet. That file defines the labels (LABEL1 type [1, Sec. 7.7]) #S_60degSectorUnifB and #E_60degSectorUnifB, for use in subsequent exercises. It also realizes a 60-sample momentum scan of the cyclotron orbits, from 200 keV to 5 MeV, using REBELOTE. It is used under the name 60degSector.inc in subsequent exercises

```
Cyclotron, classical. Analytical model of dipole field. File name: 60degSector.inc
'MARKER'  ProbMdlAnal_S                                              ! Just for edition purposes.
'OBJET'
64.62444403717985                                                             ! 200keV proton.
2
1 1
12.9248888074 0. 0. 0. 0.  1.  'm'        ! D=1 => 200keV proton. R=Brho/B=64.624444037[kG.cm]/5[kG].
1
'PARTICUL'                                                ! This is required to get the time-of-flight,
PROTON                                                      ! otherwise zgoubi only requires rigidity.
'FAISCEAU'                                                             ! Local particle coordinates.
'MARKER'    #S_60degSectorUnifB                               ! Label should not exceed 20 characters.
'DIPOLE'                                                     ! Analytical modeling of a dipole magnet.
2                 ! IL=2, only purpose is to logged trajectories in zgoubi.plt, for further plotting.
60. 50.                                                     ! Sector angle AT; reference radius RM.
30.  5. 0. 0. 0.                ! Reference azimuthal angle ACN; BM field at RM; indices, N, N', N''.
0.  0.                                                                      ! EFB 1 is  hard-edge,
4  .1455   2.2670  -.6395  1.1558  0. 0.  0.            ! hard-edge only possible with sector magnet.
30. 0.  1.E6  -1.E6  1.E6  1.E6                     ! Entrance face placed at omega+=30 deg from ACN.
0.   0.                                                                                   ! EFB 2.
4  .1455   2.2670  -.6395  1.1558  0. 0.  0.
-30. 0.  1.E6  -1.E6  1.E6  1.E6                        ! Exit face placed at omega-=-30 deg from ACN.
0. 0.                                                                          ! EFB 3 (unused).
0  0.      0.      0.      0.      0. 0.  0.
0. 0.  1.E6  -1.E6  1.E6  1.E6 0.
2 10    ! '2' is for 2nd degree interpolation. Could also be '25' (5*5 points grid) or 4 (4th degree).
1.                        ! Integration step size. The smaller, the more accurately the orbits close.
2  0. 0. 0. 0.                 ! Magnet positionning RE, TE, RS, TS. Could be instead non-zero, e.g.,
!                               2 RE=50. 0. RS=50. 0., as long as Yo is amended accordingly in OBJET.
'MARKER'    #E_60degSectorUnifB                               ! Label should not exceed 20 characters.
'FAISCEAU'                                                             ! Local particle coordinates.
'FIT'                                   ! Adjust Yo at OBJET so to get final Y = Y0 -> a closed circle.
1   nofinal
2 30 0 [12.,65.]                                                                     ! Variable : Yo.
1  2e-12  199             ! constraint; default penalty would be 1e-10; maximu 199 calls to function.
3.1 1 2 #End 0. 1. 0                                                     ! Constraint:  Y_final=Yo.
 'FAISTORE'                                               ! Log particle data here, to zgoubi.fai,
zgoubi.fai                                              ! for further plotting (by gnuplot, below).
1
'REBELOTE'                                                             ! Momentum scan, 60 samples.
60 0.2  0       1 60 different rigidities; log to video ; take initial coordinates as found in OBJET.
1                                                           ! Change parameter(s) as stated next lines.
OBJET 35  1:5.0063899693            ! Change relative rigity (35) in OBJET; range (0.2 MeV to 5 MeV).
'SYSTEM'
1                                                                        ! 2 SYSTEM commands follow.
/usr/bin/gnuplot < ./gnuplot_TOF.gnu &                             ! Launch plot by ./gnuplot_TOF.gnu.
'MARKER'  ProbMdlAnal_E                                                  ! Just for edition purposes.
 'END'
```

A 180^o *version of a DIPOLE sector, where the foregoing quantities* $AT = 60^o$, $ACN = \omega^+ = -\omega^- = 30^o$ *have been changed to* $AT = 180^o$, $ACN = \omega^+ = -\omega^- = 90^o$ *- the only modification - a file used under the name* 180*degSector.inc in further exercises*:

```
! 180degSector.inc
'MARKER'    #S_180degSectorUnifB                              ! Label should not exceed 20 characters.
 'DIPOLE'                                                    ! Analytical modeling of a dipole magnet.
2
180. 50.                                                 ! Sector angle 180deg; reference radius 50cm.
90.  5. 0. 0. 0.                    ! Reference azimuthal angle; Bo field at RM; indices, N, N', N''.
0.  0.                                                                      ! EFB 1 is  hard-edge,
4  .1455   2.2670  -.6395  1.1558  0. 0.  0.            ! hard-edge only possible with sector magnet.
90. 0.  1.E6  -1.E6  1.E6  1.E6
0.   0.                                                                                   ! EFB 2.
4  .1455   2.2670  -.6395  1.1558  0. 0.  0.
-90. 0.  1.E6  -1.E6  1.E6  1.E6
0. 0.                                                                                     ! EFB 3.
0  0.      0.      0.      0.      0. 0.  0.
0. 0.  1.E6  -1.E6  1.E6  1.E6 0.
2   10.
.5                                   ! Integration step size. The smaller, the better the orbits close.
2  0. 0. 0. 0.                 ! Magnet positionning RE, TE, RS, TS. Could be isntead non-zero, e.g.,
!                               2 RE=50. 0. RS=50. 0., as long as Yo is amended accordingly in OBJET.
'MARKER'    #E_180degSectorUnifB                              ! Label should not exceed 20 characters.
```

A gnuplot script to obtain Fig. 21.9:

```
set xlabel "R [m]"; set ylabel "T_{rev} [{/Symbol m}s]"; set y2label "f_{rev} [MHz]"
set xtics mirror; set ytics nomirror; set y2tics nomirror; set key  t l ;  set key spacin 1.2
nSector=6; Hz2MHz=1e-6; M=938.272e6; c=2.99792458e8; B=0.5; freqNonRel(x)= Hz2MHz* c**2*B/M/(2.*pi)
set y2range [7.58:7.63] ; set yrange[1/7.63:1/7.58]
plot \
"zgoubi.fai" u 10:($15 *nSector) axes x1y1 w lp pt 5 ps .6 lw 2 linecolor rgb "blue" tit "T_{rev}" ,\
"zgoubi.fai" u 10:(1/($15*nSector)) axes x1y2 w lp pt 6 ps .6 lw 2 linecol rgb "red" tit "f_{rev}" ,\
freqNonRel(x)  axes x1y2 w l lw 2. linecolor rgb "black"  tit "f_{rev},T_{rev} (non rel.)"  ; pause 1
```

3.3 Geometrical Horizontal Focusing

(a) Vertical motion.

In the absence of any focusing of the vertical motion, the vertical component of particle velocity is constant. In conjunction with the horizontal circular motion, the particle therefore spirals away from the bend plane, with constant spiral pitch.

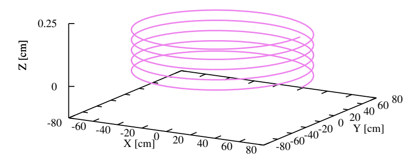

Fig. 21.10 Vertical spiral motion at constant velocity, in a constant vertical field. From both ray tracing and theory (Eq. 21.4), superimposed

The input data file of Tab. 21.8 (designed for question (b)) can be used to compute the spiral motion, by adding a vertical angle to any one of the three particles in OBJET. The result is given in Fig. 21.10.

The parametric equations of the theoretical motion, which is superimposed in Fig. 21.10, can be obtained by assuming a non-zero initial vertical velocity component in solving Eq. 3.4. They can also be obtained from the parametric Eqs. 3.15, 3.16 for $k \to 0$, namely,

$$\begin{aligned} R(s) &= R_0 \quad \text{as } x_0 = 0 \text{ and } x_0' = 0, \qquad \text{a circular motion, and} \\ y(s) &= y_0 \cos\frac{\sqrt{-k}}{R_0}(s - s_0) + y_0' \frac{R_0}{\sqrt{-k}} \sin\frac{\sqrt{-k}}{R_0}(s - s_0) \\ &\xrightarrow{k\to 0} y_0 + y_0' \times (s - s_0), \quad \text{uniform motion in the y direction} \end{aligned} \tag{21.4}$$

(b) Horizontal focusing.

Horizontal geometrical focusing stems from the non-zero curvature of the horizontal component of the motion.

The effect is shown using the input data file of Tab. 21.8, which allows producing the three horizontal trajectories displayed in Fig. 21.11.