3.4 Solutions of Exercises of Chapter 3: Classical Cyclotron

3.1 Modeling a Cyclotron Dipole: Using a Field Map

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

The first option is retained here: a Fortran program, geneSectorMap.f, given in Tab. 3.1. constructs the required map of a field distribution $B_Z(R, \theta)$, to be subsequently read and raytraced through using the keyword TOSCA [16, *lookup* INDEX].

Regarding the second option: using the analytical model DIPOLE together with the keyword OPTIONS[CONSTY=ON] to fabricate a field map, examples can be found for instance in the FFAG chapter exercises (Chap. 10).

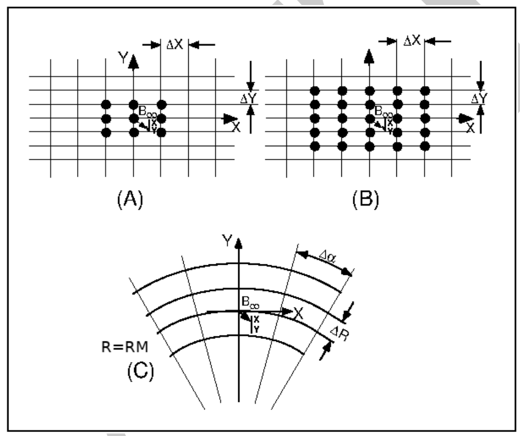

Fig. 3.19 Principle 2-D field map mesh as used by TOSCA, and the (O;X,Y) coordinate system. (A), (B): Cartesian mesh in the (X,Y) plane, case of respectively 9-point and a 25-point interpolation grid; the mesh increments are ΔX and ΔY. (C) : polar mesh and increments ΔR and $\Delta\alpha$ ($\Delta\theta$ in the text), and (O;X,Y) frame moving along a reference arc of radius R_M. The field at particle location is interpolated from its values at the closest 3×3 or 5×5 nodes

A polar mesh is retained (Fig. 3.19), 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 of the expected content are given in Tab. 3.2. That name appears under TOSCA in `zgoubi` input data file for this simulation (Tab. 3.3). Figure 3.20 shows the field over the 180^o azimuthal extent (using a gnuplot script, bottom of Tab. 3.2.

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;

(ii) the field is vertical being the mid-plane field of dipole magnet. The field is taken constant in this exercise, $\forall R$, $\forall\theta$ 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 (in Chap. 4 which addresses Thomas focusing and the isochronous cyclotron, exercises will further resort to $B_Z \equiv B_Z(R, \theta)$, an R- and θ-dependent quantity).

Table 3.1 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), BY=0.d0, BX=0.d0, Z=0.d0)

      open(unit=2,file='geneSectorMap.out')                                    ! Field map storage file.

C------------ Hypotheses :
      AT = 180.d0  /180.d0*pi         ! Angular extent of field map. Can be changed 360, 60 deg, etc.).
      BZ=5.d0                                                                              ! Field (kG).
      Rmi=1.d0; Rma=76.d0; RM=50.d0  ! cm. Radial extent of field map; reference radius to define mesh.
      dR = 0.5d0 ; NR = NINT((Rma - Rmi)/dR)+1   ! R-distance between nodes in mesh. Number of R-nodes.
C                                         RdA=RM*dA is the distance between two nodes along R=RM arc,
      RdA = 0.5d0  !      given angle increment dA (dA is the "Delta theta" quantity in the main text).
      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----------------------------------------------
      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              Z==0,          R*sinA'
     >//'              BY                BZ                BX   ix jr'
      write(2,*) '#      cm                 cm                cm '
     >//'              kG                kG                kG '
      write(2,*) '# '
      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
```

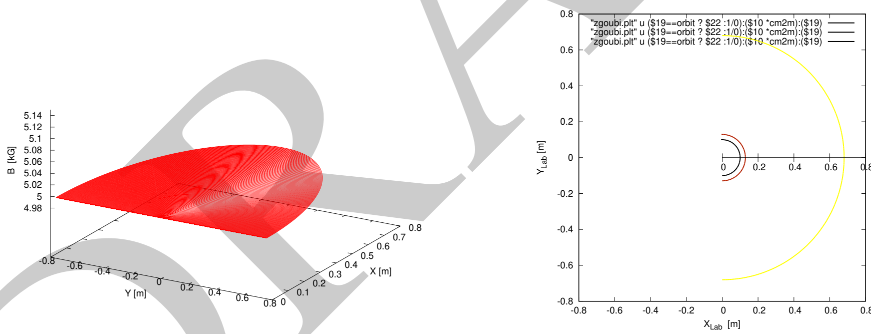

Fig. 3.20 Left: map of a constant magnetic field over a 180 deg sector, 76 cm radial extent. Right: three circular trajectories, at respectively 0.12, 0.2 and 5.52 MeV, computed using that field map

Table 3.2 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` (the following 7 are not used) 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 provide the user with various 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 and radial) node numbers, from (1,1) to (315,151) in the present case

```
  1.00       0.500       0.57324840764331209        0.00           !  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   1.800E+02   1.000E+00   7.600E+01   5.000E+01 151   5.000E-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              Z==0,          R*sinA            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
  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 a graph of $B(X,Y)$, *Fig.* 3.20:

```
# gnuplot_fieldMap.gnu
set key maxcol 1 ; set key t l ; set xtics mirror ; set ytics mirror ; cm2m = 0.01
set xlabel "Y [m]"; set ylabel "X [m]"; set zlabel "B  [kG] \n" rotate by 90; set zrange [:5.15]
splot "geneSectorMap.out" u ($1 *cm2m):($3 *cm2m):($5) w l lc rgb "red" notit; pause 1
```

This field map can be readily tested using the example of Tab. 3.3, which raytraces $E_k = 0.12,\ 0.2$ and 5.52 MeV protons on circular trajectories centered at the center of the field map. Trajectory radii, respectively $R = 10.011,\ 12.924$ and 67.998 cm (Tab. 3.3), have been prior determined from

$$\text{Rigidity}\ \ B\rho = B_0 \times R \quad \text{and} \quad B\rho = p/c = \sqrt{E_k(E_k + 2\,M)}/c \tag{3.34}$$

with $B_0 = 0.5$ T (Tab. 3.1) and $M = 938.272\,\mathrm{MeV/c^2}$ the proton mass.

The optical sequence for this particle raytracing uses the following keywords:

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

(ii) TOSCA, 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 execution listing

(iv) SYSTEM to run a gnuplot script (Tab. 3.3) once raytracing is complete

(v) MARKER, to define two particular "LABEL_1" type labels [16, *lookup* INDEX] (#S_halfDipole and #E_halfDipole), to be used with INCLUDE in subsequent exercises.

Table 3.3 Simulation input data file FieldMapSector.inc: it is set to allow a preliminary test regarding the field map geneSectorMap.out (as produced by the Fortran program geneSectorMap, Tab. 3.1), by computing three circular trajectories centered on the center of the map. This file also defines the INCLUDE segment between the labels (LABEL1 type [16, Sect. 7.7]) #S_halfDipole and #E_halfDipole

```
FieldMapSector.inc
! Uniform field 180 deg sector. FieldMapSector.inc.
'MARKER'   FieldMapSector_S                                          ! Just for edition purposes.
'OBJET'
64.62444403717985                     ! Reference Brho ("BORO" in the users' guide) -> 200keV proton.
2
3 1
10.011362 0. 0. 0. 0. 0.7745802 'a'          ! p[MeV/c]= 15.007, Brho[kG.cm]= 50.057, kin-E[MeV]=0.12.
12.924888 0. 0. 0. 0. 1.        'b'                                                 ! kin-E[MeV]=0.2.
67.997983 0. 0. 0. 0. 5.2610112 'c'          ! p[MeV/c]=101.926, Brho[kG.cm]=339.990, kin-E[MeV]=5.52.
1 1 1
'MARKER'   #S_halfDipole
'TOSCA'
0 2    ! IL=2 to log step-by-step coordinates, spin, etc., to 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.                              ! Integration step size. Small enough for orbits to close accurately.
2                                                                       ! Magnet positionning option.
0. 0. 0. 0.                                                                   ! Magnet positionning.
'MARKER'   #E_halfDipole
'FAISCEAU'
'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 a graph of the orbits, Fig. 3.20:

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

Three circular trajectories in a dee, resulting from the data file of Tab. 3.3 are shown in Fig. 3.20. Inspecting zgoubi.res execution listing one finds the D, Y, T, Z, P, S particle coordinates under FAISCEAU, at OBJET (left) and current (right) after a turn in the cyclotron (unchanged, as the trajectory forms a closed orbit):

```
        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
```

(b) Concentric trajectories in the median plane.

The optical sequence for this exercise is given in Tab. 3.4. Compared to the previous sequence (Tab. 3.3), (i) the TOSCA segment has been replaced by an 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 circular orbit for a particular momentum,
- FAISCEAU, a means to check local particle coordinates,

Table 3.4 Simulation input data file: optical sequence to find cyclotron closed orbits at a series of different momenta. An INCLUDE inserts the #S_halfDipole to #E_halfDipole TOSCA segment of the sequence of Tab. 3.3

```
Uniform field 180 deg. sector. Find orbits.
'MARKER'   FieldMapOrbits_S                                          ! Just for edition purposes.
 'OBJET'
64.62444403717985                       ! Reference Brho ("BORO" in the users' guide) -> 200keV proton.
2
1 1                                                                                  ! Just one ion.
12.9248888074 0.  0. 0. 0.  1. 'm'                ! This initial radius yields BR=64.6244440372 kG.cm.
1
'INCLUDE'                                                                ! A half of the cyclotron dipole.
1
FieldMapSector.inc[#S_halfDipole:#E_halfDipole]
'FAISCEAU'
'INCLUDE'                                                                ! A half of the cyclotron dipole.
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.

'REBELOTE'                                                                         ! Repeat what precedes,
15  0.1  0 1                                                                                  ! 15 times.
1
OBJET 30  10:80      ! Prior to each repeat, first change the value of parameter 30 (i.e., Y) in OBJET.
'SYSTEM'
2
gnuplot <./gnuplot_Zplt.gnu
cp gnuplot_Zplt_XYLab.eps gnuplot_Zplt_XYLab_stage1.eps
'MARKER'   FieldMapOrbits_E                                          ! Just for edition purposes.
'END'
```

A gnuplot script to obtain Fig. 3.21:
Note: removing the test '$51==1 ?' on column 51 in zgoubi.plt, would add on the graph the orbit as it is before each FIT.

```
# gnuplot_Zplt.gnu
set key maxcol 1 ; set key t r ; set xtics ; set ytics  ; set size ratio 1 ; set polar ; unset colorbox
set xlabel "X_{Lab}  [m] \n" ; set ylabel "Y_{Lab}  [m] \n" ; cm2m = 0.01 ; sector1=4 ; sector2=8 ; pi = 4.*atan(1.)
lmnt1 = 4; lmnt2=8  ### column numer in zgoubi.plt,  $42: NOEL;  $51: FITLST;  $49: FIT number
plot for [l=lmnt1/4:lmnt2/4] "zgoubi.plt" u ($42==4*l && $51==1 ? $22 +pi*(l-1):1/0):($10 *cm2m):($49) w p ps .3 lc pal
pause 1
```

- 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 REBELOTE itself defines, prior.

In order to compute and then plot trajectories (Fig. 3.21), `zgoubi` proceeds as follows: orbit 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 that fitting on a series of different values of R; prior to repeating, REBELOTE modifies the initial particle coordinate Y_0 in OBJET. Stepwise particle data through the dipole field are logged in zgoubi.plt, due to IL=2 under TOSCA keyword, at the first pass before FIT, and at the last pass following FIT completion. A key point here: a flag, FITLST, recorded in column 51 in zgoubi.plt [16, Sect.8.3], is set to 1 at the last pass (the last pass follows the completion of the FIT execution and uses updated FIT variable values).

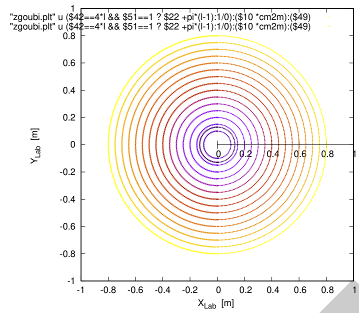

Fig. 3.21 Circular trajectories in the cyclotron mid-plane, centered on the field map center. The outermost orbit is at R=80 cm by hypothesis, thus $BR = B_0 \times R = 0.4$ T m, $E_k = 7.632$ MeV. These stepwise (R, θ) data are read from zgoubi.plt, coordinates (Y,X) in `zgoubi` polar frame nomenclature [16, Sect.8.3]

At the bottom of `zgoubi` input data file, a SYSTEM command produces a graph of ion trajectories, by executing a gnuplot script (bottom of Tab. 3.4). Note the test on FITLST, which allows selecting the last pass following FIT completion. Graphic outcomes are given in Fig. 3.21.

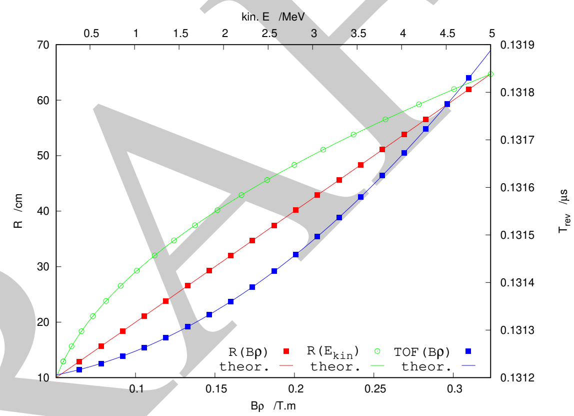

Fig. 3.22 Numerical (markers) and theoretical (solid lines) values of orbit radius, R, and revolution period, T_{rev}, versus kinetic energy (top scale) and rigidity (bottom scale). The mesh density here is $N_\theta \times N_R = 315 \times 151$. The integration step size is $\Delta s = 1$ cm, so ensuring converged results (to $\Delta R/R$ and $\Delta T_{\text{rev}}/T_{\text{rev}} < 10^{-6}$)

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 [16, Sect. 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. 3.22.

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 3.21.

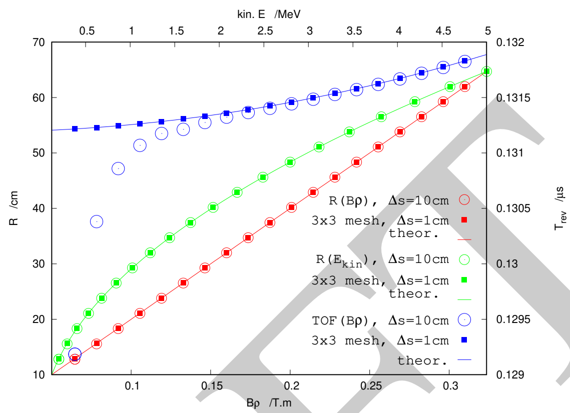

Fig. 3.23 Convergence versus mesh density and step size: a graph of orbit radius R (left axis), and revolution period, $T_{\rm rev}$ (right axis), versus kinetic energy (top scale) and rigidity (bottom scale). Solid markers are for $\Delta s = 1$ cm and $N_\theta \times N_R = 3 \times 3$ node mesh, large empty circles are for $\Delta s = 10$ cm and $N_\theta \times N_R = 106 \times 151$ node mesh. Solid lines are from theory and show convergence in the case 3×3 nodes and $\Delta s = 1$ cm

Table 3.5 Field map of a 60^o constant field sector as read by TOSCA. 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 used by `zgoubi` (the following 7 are not used), namely, the minimum value of the radius in the map, radius increment, azimuthal increment, and vertical increment (null here, as this is a single, mid-plane map)

```
   1.0       37.50      30.0   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 :
#   1.04719755E+00   60.   1.   76.   50.   3   37.5   3   26.1799388   0.523598776
# For TOSCA:           3           3  1 22.1 1.  !IZ=1 -> 2D ; MOD=22 -> polar map ; .MOD2=.1 -> one map file
#
#     R*cosA              Z==0,           R*sinA            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
  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.* 3.3, *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 (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.                                  ! Integration step size. Small enough for orbits to close accurately.
2                                                                          ! Magnet positionning option.
0. 0. 0. 0.                                                                      ! Magnet positionning.
```

(d) Numerical convergence: mesh density.

This question concerns the dependence of the numerical convergence of the solution of the differential equation of motion [16, Eq. 1.2.1] upon mesh density.

The program used in (b) to generate a field map (Tab. 3.1) 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. 3.1. The field maps geneSectorMap.out so generated for various (dR, RdA) couples may be saved under different names, and used separately.

Table. 3.5 shows the complete, 9 line, 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. 3.23 which displays both cases, over a $E_k : 0.12 \to 5$ MeV energy span (assuming protons). The reason for the absence of effect of the mesh density is that the field is constant. As a consequence the field derivatives in the Taylor series based numerical integrator are all zero [16, Sect. 1.2]: only B_Z is left in evaluating the Taylor series, however B_Z is constant. Thus R remains unchanged when pushing the ion 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

Figure 3.23 displays two cases of step sizes, $\Delta s \approx 1$ cm and $\Delta s = 10$ cm.

It has been shown (Fig. 3.22) that $\Delta s \approx 1$ cm is small enough that the numerical integration is converged, agreement with theoretical expectation is quite good.

The difference on the value of R, in the case $\Delta s \approx 10$ cm, appears to be weak, only noticeable at the scale of the graph for R values small enough that the number of steps over one revolution goes as low as $2\pi R/\Delta s \approx 2\pi \times 14.5/10 \approx 9$. 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 velocity vector $\mathbf{v}(M_1)$, from an initial location M_0 after a Δs push, are the derivatives of the velocity vector [16, Sect. 1.2] and may take substantial values if $\mathbf{v}(s)$ changes quickly. In such case, taking too large a Δs value makes the high order terms significant and the Taylor series truncation [16, Eq. 1.2.4] is fatal to the accuracy (regardless of a possible additional issue of radius of convergence of the series).

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

Issues faced are the following:

- the increase of $\delta R(\Delta s)/R$ at large Δs has been addressed above;

- a small Δs is liable to cause an increase of $\delta R(\Delta s)/R$, due to computer accuracy: truncation of numerical values at a limited number of digits may cause a Δs push to result in no change in $\mathbf{R}(M_1)$ (position) and $\mathbf{u}(M_1)$ (normed velocity) quantities [16, Eq. 1.2.4].

A detailed answer to the question, including graphs, is left to the reader, the method is the same as in (e).

3.2 Modeling a Cyclotron Dipole: Using an Analytical Field Model

This exercise introduces the analytical modeling of a dipole, using DIPOLE [16, *lookup* INDEX], and compares outcomes to the field map case of 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, namely [16, *lookup* INDEX]

$$B_Z = \mathcal{F}(\theta) B_0 \left[1 + k \left(\frac{R - R_0}{R_0} \right) + k' \left(\frac{R - R_0}{R_0} \right)^2 + k'' \left(\frac{R - R_0}{R_0} \right)^3 \right] \tag{3.35}$$

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. 3.11). $\mathcal{F}(\theta)$ 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

$$\mathcal{F} = \begin{cases} 1 & \text{inside} \\ 0 & \text{outside} \end{cases} \text{ the dipole magnet} \tag{3.36}$$

Setting up the input data list under DIPOLE (Table 3.6) requires close inspection of Fig. 3.24, 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 $ACN - \omega^-$ (exit); wedge angles, etc.

A 60 deg sector is used here for convenience, it is detailed in Table 3.6 (Table 3.7 provides the definition of a 180 deg sector, for possible comparisons with the present three-sector assembly).

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

- $R_0 = 50$ cm, an arbitrary value (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,

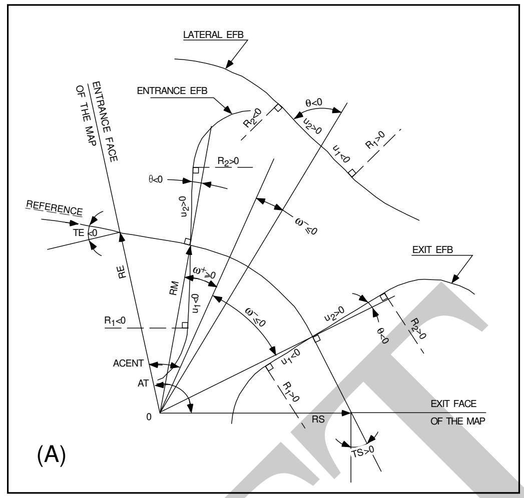

Fig. 3.24 Parameters used to define the geometry of a dipole magnet with index, using DIPOLE. In the text, ACENT is noted ACN [16, Fig. 9]

- radial field index $k = 0$ for the time being (constant field at all (R, θ)),
- a hard-edge field model for $\mathcal{F}$ (Eq. 3.36). In that manner for instance, two consecutive 60 deg sectors form a continuous 120 deg sector.

A graph of $B_Z(R, \theta)$ 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. 3.20.

(b) Concentric trajectories in the median plane.

The optical sequence of Exercise 3.1-b (Tab. 3.4) 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. 3.6 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 version of the Tab. 3.4 file include

- FIT, which finds the circular orbit for a particular momentum,
- FAISTORE to print out particle data 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 REBELOTE itself defines.

For the rest, follow the same procedure as for exercise 3.1-b. The results are the same, Fig. 3.21.

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

The orbit radius R and the revolution time $T_{\rm rev}$ 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 (Fig. 3.22).

A comparison of revolution periods can be made using the simulation file of Table 3.6 which happens to be set for a momentum scan and yields Fig. 3.25, to be compared to Fig. 3.22: 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: a similar outcome to Fig. 3.23 is expected - ignoring mesh density with the present analytical modeling using 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 problem.

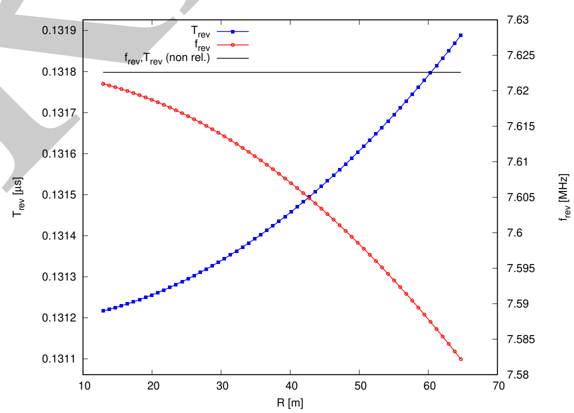

Fig. 3.25 A scan of radius-dependent revolution frequency. An analytical model of a cyclotron dipole is used, featuring uniform field (no radial gradient, at this point)

Table 3.6 Simulation input data file 60degSector.inc: analytical modeling of a dipole magnet, using DIPOLE. That file defines the labels (LABEL1 type [16, Sect. 7.7]) #S_60degSectorUnifB and #E_60degSectorUnifB, for INCLUDEs in subsequent exercises. It also realizes a 60-sample momentum scan of the cyclotron orbits, from 200 keV to 5 MeV, using REBELOTE

Note: this file is available in `zgoubi` sourceforge repository at
https://sourceforge.net/p/zgoubi/code/HEAD/tree/branches/exemples/book/zgoubiMaterial/cyclotron_classical/ProbMdlAnal/

```
60degSector.inc
! 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                                                                        ! Just one ion.
12.9248888074 0. 0. 0. 0.  1. 'm'                      ! Closed orbit coordinates for D=p/p_0=1
1                                          ! => 200keV proton. R=Brho/B=64.624444037[kG.cm]/5[kG].
'PARTICUL'                   ! Optioanl - using PARTICUL is a way to get the time-of-flight computed,
PROTON                                    ! otherwise, by default \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 R0.
30.  5. 0. 0. 0.                 ! Reference azimuthal angle ACN; BM field at R0; 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. Small enough for orbits to close accurately.
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 circular orbit.
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 gnuplot script, gnuplot_TOF.gnu, to obtain Fig. 3.25:

```
# gnuplot_TOF.gnu
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
```

Table 3.7 A 180^o version of a DIPOLE sector, where the foregoing quantities $AT = 60^o$, $ACN = \omega^+ = -\omega^- = 30^o$ have been changed to $AT = 180^o$, $ACN = \omega^+ = -\omega^- = 90^o$ - a file used under the name 180degSector.inc in further exercises

Note: this file is available in `zgoubi` sourceforge repository at
https://sourceforge.net/p/zgoubi/code/HEAD/tree/branches/exemples/book/zgoubiMaterial/cyclotron_classical/ProbMdlAnal/

```
! 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 R0; 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.
0.5                              ! Integration step size. Small enough for orbits to close accurately.
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.
```

3.3 Resonant Acceleration

The field map and TOSCA [16, *lookup* INDEX] model of a 180^o sector is used here (an arbitrary choice, the analytical field modeling DIPOLE would do as well), the configuration is that of Fig. 3.5 with a pair of sectors.

An accelerating gap between the two dees is simulated using CAVITE[IOPT=3], PARTICUL is added in the sequence in order to specify ion species and data, necessary for CAVITE to operate. Acceleration at the gap does not account for the particle arrival time in the IOPT=3 option: whatever the later, CAVITE boost will be the same as longitudinal motion is an unnecessary consideration, here).

The input data file for this simulation is given in Tab. 3.8. It is resorted to INCLUDE, twice in order to create a double-gap sequence, using the field map model of a 180^o sector. The INCLUDE inserts the magnet itself, *i.e.*, the #S_halfDipole to #E_halfDipole TOSCA segment of the sequence of Tab. 3.3. Note: the theoretical field model of Tab. 3.6, segment #S_60degSectorUnifB to #E_60degSectorUnifB (to be INCLUDEd 3 times, twice), could be used instead: exercise 3.2 has shown that both methods, field map and analytical field model, deliver the same results.

Particle data are logged in zgoubi.fai at both occurrences of CAVITE, under the effect of FAISTORE[LABEL=cavity], Tab. 3.8. This is necessary in order to access the evolution of parameters as velocity, time of flight, etc. at each half-turn, given that each half-turn is performed at a different energy

(a) Accelerate a proton.

A proton with initial kinetic energy 20 keV is launched on its closed orbit radius, $R_0 = p/qB = 4.087013$ cm. It accelerates over 25 turns due to the presence to REBELOTE[NPASS=24], placed at the end of the sequence. The energy range, 20 keV to 5 MeV, and the acceleration rate: 0.1 MeV per cavity, 0.2 MeV per turn, determine the number of turns, $NPASS+1 = (5 - 0.02)/0.2 \approx 25$. The accelerated trajectory spirals out in the fixed magnetic field, it is plotted in Fig. 3.26, reading data from zgoubi.plt.

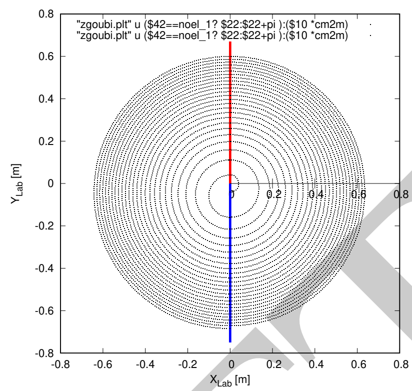

Fig. 3.26 Twenty five turn spiral trajectory of a proton accelerated in a uniform 0.5 T field from 20 keV to 5 MeV at a rate of 200 kV per turn (a 100 kV gap voltage). The vertical thick line materializes the gap, the upper half (red) corresponds to the first occurrence of CAVITE in the sequence (Tab. 3.8), the lower half (blue) corresponds to the second occurrence of CAVITE

(b) Momentum and energy.

Proton momentum p and total energy E as a function of kinetic energy, from raytracing (turn-by-turn particle data are read from zgoubi.fai, filled up due to FAISTORE) are displayed in Fig. 3.27, together with theoretical expectations, namely, $p(E_k) = \sqrt{E_k(E_k + 2M)}$ and $E = E_k + M$.

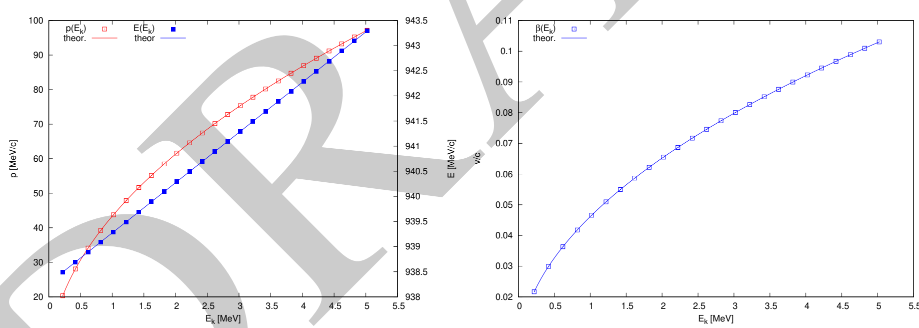

Fig. 3.27 Energy dependence of, left: proton momentum p (left axis) and total energy E (right axis) and of, right: proton normalized velocity $\beta = v/c$. Markers: from raytracing; solid lines: theoretical expectation

(c) Velocity.

Proton normalized velocity $\beta = v/c$ as a function of kinetic energy from raytracing is displayed in Fig. 3.27, together with theoretical expectation, namely, $\beta(E_k) = p/(E_k + M)$.

Table 3.8 Simulation input data file: accelerating a proton in a double-dee cyclotron, from 20 keV to 5 MeV, at a rate of 100 kV per gap, independent of RF phase (longitudinal motion is frozen - see question (e) dealing with CAVITE[IOPT=7] for unfrozen motion). Note that particle data are logged in zgoubi.fai (under the effect of FAISTORE) at both occurrences of CAVITE. The INCLUDE file FieldMapSector.inc is taken from Tab. 3.3

```
Cyclotron, classical. Acceleration:  20 keV -> 6 MeV.
'MARKER'  ProbAccelGap_S                                          ! Just for edition purposes.
'OBJET'
64.62444403717985                      ! Reference Brho ("BORO" in the users' guide) -> 200keV proton.
2
1 1                                                                                ! Just one ion.
4.087013 0. 0. 0. 0. 0.3162126 'o'            ! D=0.3162126 => Brho[kG.cm]= 20.435064, kin-E[keV]= 20.
1
'PARTICUL'                                                ! Usage of CAVITE requires partical data,
PROTON                                        ! otherwise, by default \zgoubi\ only requires rigidity.
'FAISTORE'                                                    ! Store particle data, turn-by-turn.
zgoubi.fai   cavity                  ! Log coordinates at any occurence of LABEL1=cavity, in zgoubi.fai.
1
'INCLUDE'                                                     ! Insert a 180 deg sector field map.
1
FieldMapSector.inc[#S_halfDipole:#E_halfDipole]
'FAISCEAU'                                                   ! Particle coordinates before RF gap.
'CAVITE'  cavity                                                                 ! Accelerating gap.
3                                      ! dW = qVsin(phi), independent of time (phi forced to constant).
0. 0.                                                                                      ! Unused.
100e3  1.57079632679                                           ! Peak voltage 100 kV; RF phase = pi/2.
'INCLUDE'                                                     ! Insert a 180 deg sector field map.
1
FieldMapSector.inc[#S_halfDipole:#E_halfDipole]
'FAISCEAU'                                                   ! Particle coordinates before RF gap.
'CAVITE'  cavity                                                                 ! Accelerating gap.
3                                      ! dW = qVsin(phi), independent of time (phi forced to constant).
0. 0.                                                                                      ! Unused.
100e3  1.57079632679                                           ! Peak voltage 100 kV; RF phase = pi/2.
'REBELOTE'              ! Repeat NPASS=24 times, for a total of 25 turns; K = 99: coordinates at end of
24  0.1 99                             ! previous pass are used as initial coordinates for the next pass.
'FAISCEAU'                                             ! Local particle coordinates logged in zgoubi.res.

'SYSTEM'
2                                                                            ! 2 SYSTEM command follow:
/usr/bin/gnuplot < ./gnuplot_Zplt_XYLab.gnu &                                       ! plot trajectories;
/usr/bin/gnuplot < ./gnuplot_awk_Zfai_dTT.gnu &                         ! dC/C, dbta/bta, dT/T graph.

'MARKER'  ProbAccelGap_E                                          ! Just for edition purposes.
'END'
```

Two gnuplot scripts, to obtain respectively Fig. 3.26: *and Fig.* 3.28:
The awk command in gnuplot_awk_Zfai_dTT.gnu takes care of a 1-row shift so to subtract next turn data from currant turn ones.

```
# gnuplot_Zplt_XYLab.gnu
set xtics ;  set ytics ; set xlabel "X_{Lab} [m]" ; set ylabel "Y_{Lab} [m]"
set size ratio 1 ; set polar ; cm2m = 0.01 ; pi = 4.*atan(1.)
set arrow from 0, 0 to 0, 0.67 nohead lc "red" lw 6; set arrow from 0, -0.75 to 0, 0 nohead lc "blue" lw 6
noel_1=6 ; noel_2=11  # 1st CAVITE is element noel_1; 2nd CAVITE is noel_2. Col. $42 in zgoubi.plt is element numb.
plot for  [nl=noel_1:noel_2:5] "zgoubi.plt" u ($42==noel_1? $22:$22+pi ):($10 *cm2m) w p pt 5 ps .2 lc rgb "black"

# gnuplot_awk_Zfai_dTT.gnu
set xtics nomirror; set ytics mirror; set xlabel "E_k [MeV]";
set ylabel "{/Symbol Db}/{/Symbol b},  {/Symbol D}C/C,  {/Symbol D}T_{rev}/T_{rev}"; set logscale y; set yrange [:3]
# zgoubi.fai columns: $25: energy; $14: path length; $23: kinetic E; $29: mass; $15: tim
plot "< awk '/#/ {next;}  { if(prev14>0 && prev25>0) print prev24, ($14 -prev14)/prev14 , prev24}  \
{ prev14 = $14; prev24 = $24; prev25=$25 }' < zgoubi.fai" u 1:2 w p pt 5 lc rgb "black" tit "{/Symbol D}C/C" ,\
"< awk '/#/ {next;}  { if(prev14>0 && prev25>0) print prev24, ( -sqrt(prev25**2-$29**2)/prev25 + \
sqrt($25**2-$29**2)/$25 )/(sqrt(prev25**2-$29**2)/prev25) , prev24}  { prev14 = $14; prev24 = $24; prev25=$25 }' \
< zgoubi.fai"  u 1:2 w p pt 6 ps 1.5 lc rgb "red" tit "d{/Symbol b}/{/Symbol b}" ,\
"< awk '/#/ {next;}  { if(prev14>0 && prev25>0) print prev24, ($14 -prev14)/prev14- ( -sqrt(prev25**2-$29**2)/prev25 \
+ sqrt($25**2-$29**2)/$25 )/(sqrt(prev25**2-$29**2)/prev25) , prev24}  { prev14 = $14; prev24 = $24; prev25=$25 }' \
< zgoubi.fai"  u 1:2 w p pt 8 ps 1.5 lc rgb "blue"  tit "{/Symbol D}T/T=dC/C-d{/Symbol b}/{/Symbol b}" ,\
"< awk '/#/ {next;}  { if(prev14>0 && prev15>0) print prev24, ($15-prev15)/prev15 , prev24}  { prev14 = $14; \
prev24 = $24; prev15=$15 }' < zgoubi.fai"  w l lw 2 lc rgb "blue" tit "theor. {/Symbol D}T/T"
```

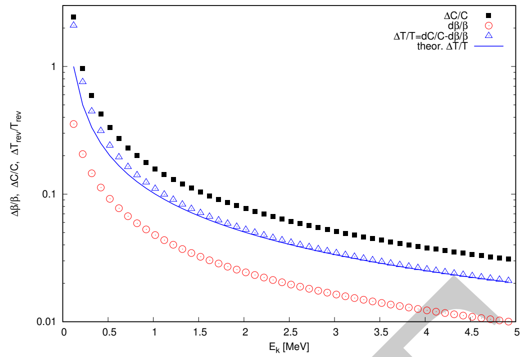

Fig. 3.28 Relative variation of velocity $\Delta\beta/\beta$ (empty circles), circumference $\Delta C/C$ (solid disks) and revolution time $\Delta T/T$ (triangles), as a function of energy, from raytracing. Theoretical expectation for the latter is also displayed (solid line), for comparison

(d) Relative velocity, orbit length and time of flight.

The relative increase in velocity is smaller than the relative increase in orbit length as energy increases (this is what Fig. 3.28 shows). Thus the relative variation of the revolution time, Eq. 3.23, is positive; in other words the revolution time increases with energy, the revolution frequency decreases. Raytracing outcomes are displayed in Fig. 3.28, they are obtained using the gnuplot script given in Tab. 3.8. Note that the path length difference (taken as the difference of homologous quantities in a common line) is always between the two CAVITEs (particle data are logged at the two occurrences of CAVITE), crossed successively, which is half a turn. Same for the difference between homologous velocity data on a common line, it corresponds to two successive crossings of CAVITE, *i.e.*, half a turn. The graph includes the theoretical $\delta T_{\rm rev}/T_{\rm rev}$ (Eq. 3.23) for comparison with raytracing; some difference appears in the low velocity regime, this may be due to the large $\Delta\beta$ step imparted by the 100 kV acceleration at the gaps.

(e) Harmonic h=3 RF.

The input data file for this simulation is given in Tab. 3.9. The RF is on harmonic h=3 of the revolution frequency. It has been tuned to ensure acceleration up to 3 MeV. The accelerating gap between the two dees is simulated using CAVITE[IOPT=7]: by contrast with the previous exercise (where CAVITE[IOPT=3] is used), the RF phase at ion arrival at the gap is now accounted for.

Repeating questions (b-d) is straightforward, changing what needs be changed in Tab. 3.9 input data file.

Table 3.9 Simulation input data file: accelerating a proton in a double-dee cyclotron, from 20 keV to 5 MeV, using harmonic 3 RF. The INCLUDE file is taken from Tab. 3.6

```
Cyclotron, classical. Analytical model of dipole field.
'OBJET'
64.62444403717985                                                          ! 200keV proton.
2
1 1                                                                          ! Just one ion.
12.924888 0. 0. 0. 0.  1.  'm'            ! D=1 => 200keV proton. R=Brho/B=64.624444037[kG.cm]/5[kG].
1
'PARTICUL'                                          ! This is required for spin motion to be computed,
PROTON                                         ! otherwise, by default \zgoubi\ only requires rigidity.
'INCLUDE'
1                                                                ! Include a first 180 deg sector.
./180degSector.inc[#S_180degSectorUnifB:#E_180degSectorUnifB]
'CAVITE'
7
0 22862934.0
285e3 -0.5235987755982988
'INCLUDE'
1                                                               ! Include a second 180 deg sector.
./180degSector.inc[#S_180degSectorUnifB:#E_180degSectorUnifB]
'CAVITE'
7
0 22862934.0                                                                ! RF = 3/T_rev.
285e3 -3.665191429188092                                        ! Peak voltage; synchronous phase.
'REBELOTE'
26 0.4 99                                                               ! 26+1 turn tracking.
 'END'
```

3.4 Spin Dance

The DIPOLE analytical field model of exercise 3.2 (Tab. 3.6) is used here, as opposed to using a field map and TOSCA, as it allows more straightforward changes in the field, if desired.

(a) Spin transport.

Spin transport is obtained by adding SPNTRK. PARTICUL is necessary in order to get the Thomas-BMT equation of motion solved [16, Sect. 2]. This results in the input data file given in Tab. 3.10 (excluding FIT and REBELOTE keywords, introduced for the purpose of the following question (b)).

The use of SPNTRK results in the following outcome (an excerpt from zgoubi.res execution listing):

```
      4  Keyword, label(s) :  SPNTRK
              Spin  tracking  requested.
                        Particle  mass          =    938.2721     MeV/c2
                        Gyromagnetic  factor  G =    1.792847
                        Initial spin conditions type  1 :
                             All particles have spin parallel to  X  AXIS
                        PARAMETRES  DYNAMIQUES  DE  REFERENCE :
                             BORO    =         64.624 kG*cm
                             beta    =    0.02064411
                             gamma   =   1.00021316
                             gamma*G =   1.7932295094
                        POLARISATION  INITIALE  MOYENNE  DU  FAISCEAU  DE        1  PARTICULES :
                             <SX> =     1.000000
                             <SY> =     0.000000
                             <SZ> =     0.000000
                             <S>  =     1.000000
```

Spin coordinates are logged in zgoubi.res execution listing using SPNPRT. Five sample passes around the cyclotron (four iterations by REBELOTE) result in the following outcomes in zgoubi.res, under SPNPRT:

```
     26  Keyword, label(s) :  SPNPRT
                INITIAL                                              FINAL
            SX        SY        SZ        |S|          SX        SY        SZ        |S|        GAMMA
   m  1  1.000000  0.000000  0.000000  1.000000     0.268269  0.963344  0.000000  1.000000      1.0002
   m  1  1.000000  0.000000  0.000000  1.000000     0.268599  0.963252  0.000000  1.000000      1.0002
   m  1  1.000000  0.000000  0.000000  1.000000     0.268949  0.963154  0.000000  1.000000      1.0003
   m  1  1.000000  0.000000  0.000000  1.000000     0.269319  0.963051  0.000000  1.000000      1.0003
   m  1  1.000000  0.000000  0.000000  1.000000     0.269710  0.962942  0.000000  1.000000      1.0003
```

Table 3.10 Simulation input data file: add spin to the cyclotron simulation of Tab. 3.6. The present input file INCLUDEs six copies of the 60 degree sector DIPOLE defined therein

```
Cyclotron, classical. Analytical model of dipole field. Spin transport.
'MARKER'  ProbAddSpin_S                                              ! Just for edition purposes.
'OBJET'
64.62444403717985                    ! Reference Brho ("BORO" in the users' guide) -> 200keV proton.
2
1 1                                                                              ! Just one ion.
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, by default \zgoubi\ only requires rigidity.
'SPNTRK'                                                                     ! Request spin tracking.
1                                                   ! All spins launched longitudinal (parallel to OX axis).
'INCLUDE'
1
6* ./60degSector.inc[#S_60degSectorUnifB:#E_60degSectorUnifB]                  ! 6 * 60 degree sector.
'FAISCEAU'                                                                ! Local particle coordinates.
'FIT'                                    ! Adjust Yo at OBJET so to get final Y = Y0 -> a circular orbit.
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.
'FAISCEAU'                                           ! Allows checking that Y = Y0 and T = T0 = 0, here.
'SPNPRT'                                                          ! Local spin data, logged in zgoubi.res.
'FAISTORE'                                                        ! Log particle data here, to zgoubi.fai,
zgoubi.fai                               ! for further plotting of spin coordinates (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_Zfai_spin.gnu &
'MARKER'  ProbAddSpin_E                                                       ! Just for edition purposes.
 'END'
```

A gnuplot script to obtain Fig. 3.29:
The file zgoubi.1cm is a copy of zgoubi.fai obtained for a $\Delta s = 1\,\text{cm}$ run; zgoubi.fai is for $\Delta s = 0.5\,\text{cm}$.

```
# gnuplot_Zfai_spin.gnu
set xlabel "G{/Symbol g}"; set ylabel "Spin precession angle  {/Symbol q}_{sp} / 2{/Symbol p}"
set y2label "relative difference num./theor"; set logscale y2
set xtics; set ytics nomirror; set y2tics; am = 938.27208; G = 1.79284735; pi = 4.*atan(1.); set key t c spacin 1.5
plot \
"zgoubi.fai" u ($31*$25/$29):(((4.*pi -atan($21/$20)))/(2.*pi)) w lp pt 4 ps .7 tit "{/Symbol q}_{sp}/2{/Symbol p}" ,\
"zgoubi.1cm" u ($31*$25/$29):(abs((4*pi-atan($21/$20))/pi*180-$31*$25/$29*360.)) axes x1y2 w lp pt 8 ps .7 tit "1 cm",\
"zgoubi.fai" u ($31*$25/$29):(abs((4.*pi -atan($21/$20))/pi*180-$31*$25/$29*360.)) axes x1y2 w lp pt 8 ps .7 tit "5 mm"
```

(b) Spin precession.

Proton case is considered, simulation is performed using Tab. 3.10 input data file. Initial spin is parallel to the X axis (longitudinal). The particle is raytraced on the circular closed orbit over one revolution, for a particular momentum. Particle data resulting from a FIT (FIT forces orbit closure, by varying the initial Y_0) are logged in zgoubi.fai, by FAISTORE. The computation is repeated using REBELOTE in the very manner that the energy scan was done in exercise 3.2, over an energy range 12 keV→5 MeV.

Figure 3.29 (obtained using the gnuplot script given in Tab. 3.10) displays the resulting energy dependence of the spin precession, $\theta_{\rm sp}(E)$, together with its difference to theoretical expected $\theta_{\rm sp}(E) = G\frac{E}{M} \times 2\pi = G\gamma \times 2\pi$ (proton gyromagnetic anomaly $G = 1.792847$).

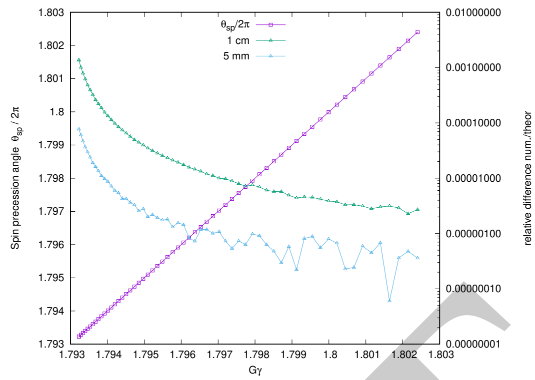

Fig. 3.29 $G\gamma$ dependence of the spin precession angle over a revolution around the cyclotron, in the moving frame (left axis), and relative difference to $G\gamma$ for the two integration step sizes $\Delta s = 0.5$ and 1 cm (right axis), Markers are from raytracing, solid lines are to guide the eye

(c) Spin tune.

Two protons are injected with longitudinal initial spin $\mathbf{S}_i \parallel$ OX axis and respective energies 12 keV and 5.52 MeV, thus the following OBJET (a slight modification to Tab. 3.10 data):

```
'OBJET'
64.62444403717985                       ! Reference Brho ("BORO" in the users' guide) -> 200keV proton.
2
2 1
12.9248888074 0. 0. 0. 0.  1.   'm'       ! D=1 => 200keV proton. R=Brho/B=64.624444037[kG.cm]/5[kG].
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
```

FAISCEAU following FIT (Tab. 3.10) allows to control that momentum and trajectory radius are matched, which means coordinates at OBJET and current coordinates at FAISCEAU are equal. Inspection of zgoubi.res execution listing shows for instance, after 4 turns:

```
                                    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)
      1.0000    12.925     0.000     0.000     0.000      0.0000     0.0000   12.925    0.000    0.000    0.000   3.248379E+02
      5.2610    67.998     0.000     0.000     0.000      0.0000     4.2610   67.998   -0.000    0.000    0.000   1.708976E+03
```

A graphic of the projection of the spin motion on the longitudinal axis, over a few turns, from the ray tracing, is given in Fig. 3.30, together with the longitudinal component as of the parametric equations of motion

$$\begin{cases} S_X = \hat{S}\,\cos(G\gamma\theta) \\ S_Y = \hat{S}\,\sin(G\gamma\theta) \end{cases} \tag{3.37}$$

The motion amplitude is $\hat{S} = \sin\phi$, with ϕ the angle that the spin vector makes with the vertical precession axis. In this simulation **S** is launched parallel to OX, thus $\phi = \pi/2$ and $\hat{S} = 1$.

Now, checking the spin precession:

Placing both FAISCEAU and SPNPRT commands right after the first dipole sector allows checking the spin precession and its relationship to particle rotation, for simplicity right after the first pass through that first sector, as follows. FAISCEAU and SPNPRT (Tab. 3.10) yield, respectively:

```
                                    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)
      1.0000    12.925     0.000     0.000     0.000      0.0000     0.0000   12.925    0.000    0.000    0.000   3.248379E+02
      5.2610    67.998     0.000     0.000     0.000      0.0000     4.2610   67.998   -0.000    0.000    0.000   1.708976E+03
```

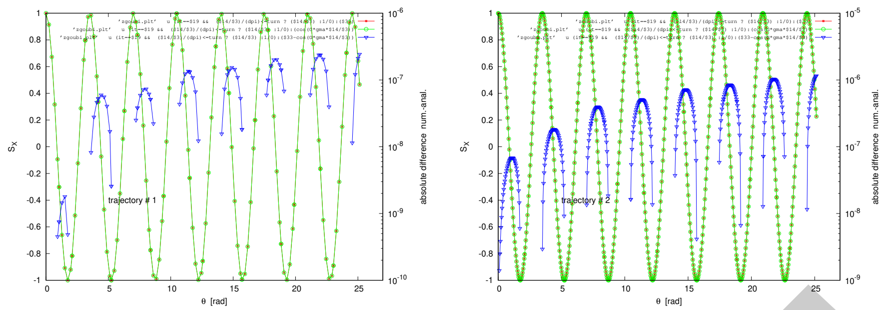

Fig. 3.30 Longitudinal spin component motion (left vertical axis), observed in the moving frame, case of 0.2 MeV energy, R=12.924888 cm (left graph), and of 5.52 MeV energy, R=67.998 cm (right graph). Markers are from ray tracing, the solid line is the theoretical expectation (Eq. 3.37). The right vertical axis (triangle markers; solid line is to guide the eye) shows the absolute difference between both. The oscillation is as expected slightly faster at 5.52 MeV: frequencies are in the ratio $\gamma(5.52\,\mathrm{MeV})/\gamma(0.2\,\mathrm{MeV}) = 1.00566$

```
                 INITIAL                                          FINAL                                      --- angles  ---
            SX        SY        SZ        |S|             SX        SY        SZ        |S|        GAMMA    |Si,Sf|    (Z,Sf)
                                                                                                            (deg.)     (deg.)
 m  1  1.000000  0.000000  0.000000  1.000000        -0.302266 -0.953224  0.000000  1.000000        1.0002 -107.594   90.000
 o  1  1.000000  0.000000  0.000000  1.000000        -0.312396 -0.949952  0.000000  1.000000        1.0059 -108.204   90.000
```

SPNPRT tells that,

- case of the first particle, tagged 'm' above; its energy is 200 keV, $\gamma = 1.00021315$, its spin tune is $\nu_{sp} = G\gamma = 1.793229$

The computed value of the '(S_i, S_f)' angle between initial and final spin vectors is -107.594 (truncated), negative as spin precession has the sign of proton rotation. Theoretical expectation is $G\gamma\alpha = -107.59377$ deg. The resulting spin components are, as above, $S_X = \cos(-107.59377) = -0.302266$ and $S_Y = \sin(-107.59377) = -0.9532235$.

- case of the second particle, tagged 'o'; its energy is 5.52 MeV, $\gamma = 1.00588315$, its spin tune is $\nu_{sp} = G\gamma = 1.803394$

The computed value of '(S_i, S_f)' is -108.204 (truncated). Theoretical expectation is $G\gamma\alpha = -108.20370$ deg.

Now, accounting for particle rotation in order to get spin coordinates in the laboratory frame:

- the FAISCEAU outcome above shows that, after crossing the 60 deg sector the angles of the two particles have the value $T = 0$, which is expected as they are launched with zero incidence, and as DIPOLE uses a polar coordinate system [16] with particle coordinates computed in the moving (rotating) frame. The latter has also undergone a -60 deg rotation, clockwise, which is therefore the implicit rotation of the particles in the laboratory frame. The spin precession in the laboratory frame results, namely,

- case of the first particle: $(1 + G\gamma)\alpha = -167.59377$ deg.

- case of the second particle: $(1 + G\gamma)\alpha = -168.20370$ deg.

(d) Spin dance.

A 200 keV proton is injected with its initial spin vector at 80 degrees from the vertical axis. The input data file for this simulation is given in Tab. 3.11, together with a gnuplot script for the animation. The latter plots three things, concurrently:

- the circular trajectory of the particle in the (X,Y) plane: this is the curve at Z=0 in Fig. 3.31, a set of points $\{(R\cos(-X), R\sin(-X), 0)\}$ resulting from the step by step integration. Note that X is counted positive clockwise in zgoubi.fai (consistently with the definition of DIPOLE parameters, Fig. 9 in [16]), hence "-X" the rotation angle;

- the spin vector: its foot is attached to the particle (the previous set of points), whereas its tip is at $\{(S_X\cos(-X) - S_Y\sin(-X), S_X\sin(-X) + S_Y\cos(-X), S_Z\}$, with S_X, S_Y, S_Z the spin vector components in the moving frame as read from zgoubi.fai. S_Z is constant as the precession axis is parallel to the Z axis. The $\begin{pmatrix}\cos(-X) & -\sin(-X)\\ \sin)-X) & \cos(-X)\end{pmatrix}$ rotation applied to the (S_X, S_Y) vector accounts for the transformation from the moving frame to the laboratory frame;

- the cycloidal shape trajectory of the tip of the spin vector (the previous set of points).

A frozen view of that spin dance, over about 2.5 proton revolutions around the ring, is given in Fig. 3.31.

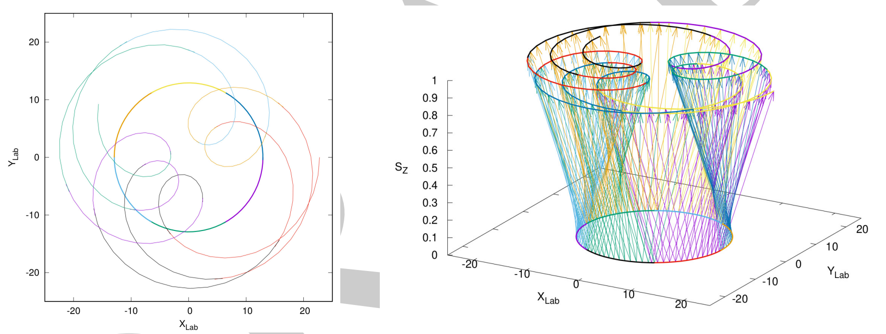

Fig. 3.31 Dance - frozen, here - of the spin of a 200 keV proton over 2.5 turns around the cyclotron. The circle on the left, or bottom closed curve on the right, is the trajectory of the proton. The cycloidal curve represents the motion of the spin vector tip in the moving frame

Table 3.11 Simulation input data file: spin dance, 20 turns around a uniform field cyclotron. The INCLUDE file 60degSector.inc is taken from Tab. 3.6

Note: this animation (input data file & gnuplot script) is available in `zgoubi` sourceforge repository at
https://sourceforge.net/p/zgoubi/code/HEAD/tree/branches/exemples/book/zgoubiMaterial/cyclotron_classical/ProbAddSpin/spinDance/

```
Cyclotron, classical. Spin dance.
'MARKER'  ProbAddSpinDance_S                                            ! Just for edition purposes.
'OBJET'
64.62444403717985                      ! Reference Brho ("BORO" in the users' guide) -> 200keV proton.
2
1 1                                                                                ! Just one ion.
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, by default \zgoubi\ only requires rigidity.
'SPNTRK'                                                                    ! Request spin tracking.
4.1                                                                        ! All spins are initially
0.984807753012  0. 0.173648177667                                         ! at 10 degrees to X axis.
'FAISCEAU'
'INCLUDE'
1
6* ./60degSector.inc[#S_60degSectorUnifB:#E_60degSectorUnifB]                  ! 6 * 60 degree sector.
'REBELOTE'                                                                              ! Multiturn:
19 0.2  99                                                                    19 additional passes.
'SYSTEM'
1
gnuplot < ./gnuplot_Zplt_SDance.gnu
'MARKER'  ProbAddSpinDance_E                                            ! Just for edition purposes.
'END'
```

A gnuplot script to obtain the spin dance in Fig. 3.31. *Note a "mag" factor, aimed at artificially increasing the amplitude of the vector tip oscillation in this graphic*:

```
set xlabel "X_{Lab}"; set ylabel "Y_{Lab}"; set zlabel "S_Z"; set xtics; set ytics; set ztics  #unset ztics
set zrange  [0:]; set xrange [-25:25]; set yrange [-25:25]; set xyplane 0
dip1=7; dip2=22; dd=3 # positining of 1st and last dipoles in zgoubi.dat sequence, and increment
# magnifies apparent spin tilt   speed up graphic           pi/3                    z norm
        mag = 10.           ;     speedUp=1     ;     pi3 = 4.*atan(1.)/3  ;     nz=0.18

# JUST 2D, PROJECTED IN (X,Y) PLANE, FIRST:
set size ratio -1
do for [i=1:239]{ plot \
for [dip=dip1:dip2:dd] "zgoubi.plt" every 1::::speedUp*i u ($19==1 && $42==dip? $10*cos(-$22-pi3*(dip-6.)/3.) :1/0): \
($10*sin(-$22-pi3*(dip-6.)/3.)) w l lw 3 notit ,\
for [dip=dip1:dip2:dd] "zgoubi.plt" every 1::::speedUp*i u ($19==1 && $42==dip? $10*cos(-$22-pi3*(dip-6.)/3.) \
+mag*(cos(-$22-pi3*(dip-6.)/3.)*$33-sin(-$22-pi3*(dip-6.)/3.)*$34) :1/0): \
($10*sin(-$22-pi3*(dip-6.)/3.) +mag*(sin(-$22-pi3*(dip-6.)/3.)*$33+cos(-$22-pi3*(dip-6.)/3.)*$34)) w l notit }
unset size

# 3D, NEXT:
do for [i=1:239]{ splot \
for [dip=dip1:dip2:dd] "zgoubi.plt" every speedUp*i::::speedUp*i u ($19==1&& $42==dip? $10*cos(-$22-pi3*(dip-6)/3):1/0):\
($10*sin(-$22-pi3*(dip-6)/3)):($1*0):(mag*(cos(-$22-pi3*(dip-6)/3)*$33-sin(-$22-pi3*(dip-6)/3)*$34)): \
(mag*(sin(-$22-pi3*(dip-6)/3)*$33+cos(-$22-pi3*(dip-6)/3)*$34)):($35/nz)  w vectors notit ,\
for [dip=dip1:dip2:dd] "zgoubi.plt" every 1::::speedUp*i u ($19==1 && $42==dip? $10*cos(-$22-pi3*(dip-6)/3) :1/0): \
($10*sin(-$22-pi3*(dip-6)/3)):($1*0):($19==1&&$42==dip? $10*cos(-$22-pi3*(dip-6)/3):1/0):($10*sin(-$22-pi3*(dip-6)/3)): \
($1*0) w l lw 3 notit ,\
for [dip=dip1:dip2:dd] "zgoubi.plt" every 1::::speedUp*i u ($19==1 && $42==dip? $10*cos(-$22-pi3*(dip-6)/3)+mag*( \
cos(-$22-pi3*(dip-6)/3)*$33-sin(-$22-pi3*(dip-6)/3)*$34):1/0):($10*sin(-$22-pi3*(dip-6)/3)+mag*(sin(-$22-pi3*(dip-6)/3) \
*$33+cos(-$22-pi3*(dip-6)/3)*$34)):($35/nz):($19==1&&$42==dip? $10*cos(-$22-pi3*(dip-6)/3 +mag*(cos(-$22-pi3*(dip-6)/3) \
*$33 -sin(-$22-pi3*(dip-6)/3)*$34)) :1/0): ($10*sin(-$22-pi3*(dip-6)/3) +mag*(sin(-$22-pi3*(dip-6)/3)*$33+cos(-$22-pi3* \
(dip-6)/3) *$34)):($35/nz) w l lw 3 notit  }
```

(e) Deuteron.

The input data file set up for questions (b-e) can be used *mutatis mutandis*, as follows.

Raytracing a different particle requires changing the reference rigidity, BORO, under OBJET, and changing particle data, under PARTICUL. That reference rigidity is to be determined from the field value in the dipole model (namely, $B_0 = 5\,\mathrm{kG}$).

Particle data for these two particles are (respectively mass (MeV/c^2), charge (C), G factor):

$$deuteron: \quad 1875.612928 \quad 1.602176487 \times 10^{-19} \quad -0.14301$$

$$^3He^{2+}: \quad 2808.391585 \quad 3.204352974 \times 10^{-19} \quad -0.14301$$

3.5 Synchronized Spin Torque

The simulation input data file of exercise 3.4-(d) can be used here, with a few addenda or modifications, as follows:

(i) the initial ion coordinate D (rigidity relative to the reference BORO=64.6244440) under OBJET has to be calculated for the four energies concerned;

(ii) the closed orbit radius at 0.2, 108.412, 118.878 and 160.746 MeV has to be found; calculation is straightforward given that the field considered here is vertical, uniform, namely, B_Z=constant=5 kG, $\forall R$, so that $R = B\rho/B_Z$; otherwise a FIT procedure can be used to find the orbit radius, given the rigidity, as done already in various exercises [16, lookup "closed orbit"], that could help for instance in the presence of a radial index, or field defects;

(iii) initial spins are set vertical for convenience, but this is not mandatory;

(iv) the multiturn tracking is set to a few 10s of turns, in order to allow a few spin precessions;

(v) particle data through DIPOLEs are saved step-by-step all the way in zgoubi.plt by means of IL=2 (the integration step size is 1 cm (Tab. 3.6), thus zgoubi.plt may end up bulky);

(vi) turn-by-turn data are saved in zgoubi.fai by means of FAISTORE;

(vii) SPINR is added at the end of the sequence, to impart on spins the requested X-tilt.

This results in the updated simulation input data file given in Tab. 3.12.

The oscillatory motion of the vertical spin component as the ion orbits around the ring, is displayed in Fig. 3.32. The spin points upward, parallel to the vertical axis at start; SPINR kick is 10 deg in the present case. At $G\gamma = 2$ the spin always finds itself back in the (Y,Z) transverse plane after one proton orbit, this synchronism causes the cumulated spin tilt at SPINR to take the value $N \times 10$ deg (with N the number of orbits). Thus after 18 proton orbits, 36 spin precessions, the spin points downward; it takes 36 orbits, or 226.194 rad, to complete an oscillation. If $G\gamma$ moves away from an integer, the spin tilts with bounded amplitude, within the limits of a cone.

Additional graphs and details are obtained using the simulation file of Tab. 3.13. This file simulates spin motion in three different cases, $G\gamma = 1.79322$, $G\gamma = 2$, integer, yielding an integer number of spin precessions over one proton orbit around the cyclotron, and $G\gamma = 2.5$, half-integer, yielding a half-integer number of spin precessions over one proton orbit. Outcomes are given in Fig. 3.33 which shows the spin motion projected on the (X,Y) plane (horizontal), and on a sphere, step-by-step. The spin kick by SPINR is 20 deg in this case. If $G\gamma = 1.793229$, far from an integer, **S**, initially vertical, remains at a bounded angle to the vertical axis, X-kicked from one circle to another, turn after turn; if $G\gamma = 2$ the spin vector flips by 20 degree in the (Y,Z) plane at SPINR, turn after turn; if $G\gamma = 2.5$, half-integer, the spin vector

Table 3.12 Simulation input data file: superimposition of a turn-by-turn localized 10 deg X-rotation of the spin (using SPINR[$\phi = 0, \mu = 10$]), on top of Thomas-BMT $2\pi G\gamma$ Z-precession. The INCLUDE file 60degSector.inc is taken from Tab. 3.6

```
Cyclotron, classical. Synchronous spin kick.
'MARKER'  ProbAddSpinTorque_S                                          ! Just for edition purposes.
'OBJET'
64.62444403717985                     ! Reference Brho ("BORO" in the users' guide) -> 200keV proton.
2
4 1
12.9248888074 0. 0. 0. 0.  1.     'm'      ! D=1 => 200keV proton. R=Brho/B=64.624444037[kG.cm]/5[kG].
3.0947295453790e2 0. 0. 0. 0.  23.9439548880185 'm'   ! Ggamma=2
3.2492145208941e2 0. 0. 0. 0.  25.1392067607172 'm'   ! Ggamma=2.02
3.8177333586897e2 0. 0. 0. 0.  29.5378429599586 'm'   ! Ggamma=2.1
1 1 1 1
'PARTICUL'                                        ! This is required for spin motion to be computed,
PROTON                                      ! otherwise, by default \zgoubi\ only requires rigidity.
'SPNTRK'                                                                 ! Request spin tracking.
4.1
0. 0. 1.                                                       ! Initial spin vector is defined here.

'FAISTORE'
zgoubi.fai
1

'INCLUDE'
1
6* ./60degSector.inc[#S_60degSectorUnifB:#E_60degSectorUnifB]                 ! 6 * 60 degree sector.
'FAISCEAU'
'SPINR'
1                                                                                ! Spin rotation,
0. 10.                                     1 about the X-axis, by 10 or 20 dgrees as the case may be.

'REBELOTE'                                                              ! Multiturn ray-tracing.
39 0.2  99
'SYSTEM'
1
gnuplot < ./gnuplot_Zplt_spinTilt.gnu
'MARKER'  ProbAddSpinTorque_E                                          ! Just for edition purposes.
'END'
```

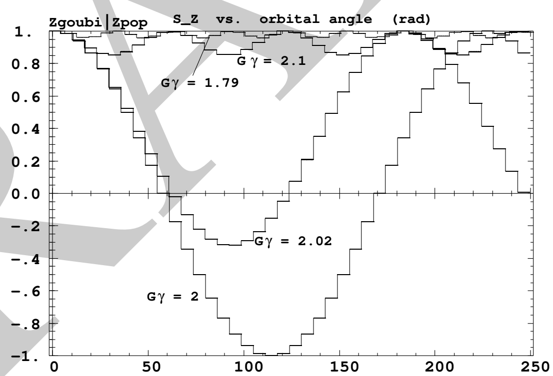

Fig. 3.32 S_Z motion versus orbital angle, while the ion orbits on a circle. S_Z is constant over a turn and then undergoes a discontinuity upon the 10 deg X-tilt, hence the step function. At $G\gamma = 2$ it takes 36 turns, or 226.194 rad, to complete an oscillation. A graph obtained using zpop: menu 7; 1/1 to open zgoubi.plt; 2/[6,23] for S_Z versus θ; 7 to plot

undergoes a half-integer number of precessions over one orbit around the cyclotron, it jumps and alternates between vertical, and the surface of the 20 degree Z-axis cone.

Table 3.13 Simulation input data file: a similar simulation to 3.12, for different $G\gamma$ values, namely 1.79322, 2 and 2.5. The spin kick at SPINR has been changed to 20 deg. Regarding the use of OBJET[IEX] option: IEX=-9 allows inhibiting the tracking for the particle(s) concerned, all the rest left unchanged; it is necessary here to have at least one particle with IEX=1, for proper operation of the gnuplot scripts. The INCLUDE file 60degSector.inc is taken from Tab. 3.6

```
Cyclotron, classical. Synchronized spin kick in a uniform field
'MARKER'  ProbAddSpinSphere_S                                          ! Just for edition purposes.
'OBJET'
64.62444403717985                       ! Reference Brho ("BORO" in the users' guide) -> 200keV proton.
2
3 1
12.924889  0. 0. 0. 0.   1.          'o'                              ! Ggamma=1.793229 ->  0.200MeV;
309.47295  0. 0. 0. 0.  23.943951797 'i'                              ! Ggamma=2 ->    108.411628MeV;
608.30878  0. 0. 0. 0.  47.064911290 'h'                              ! Ggamma=2.5 ->  370.082556MeV.
1 1 1                           ! For any particle: set to 1 to enable ray-tracing, or to -9 to ignore.
'PARTICUL'                                            ! This is required for spin motion to be computed,
PROTON                                       ! otherwise, by default \zgoubi\ only requires rigidity.
'SPNTRK'                                                                      ! Request spin tracking.
4.1                                                          ! All initial spins taken parallel to Z axis.
0. 0. 1.

'SPNPRT'  PRINT

'INCLUDE'
1
6* ./60degSector.inc[#S_60degSectorUnifB:#E_60degSectorUnifB]                    ! 6 * 60 degree sector.
'FAISCEAU'
'SPINR'
1                                                                                   ! Spin rotation,
0. 20.                                                      1 about the X-axis, by 20 degree here.

'REBELOTE'                                                 ! REBELOTE[K=99] for multiturn ray-tracing,
39 0.2  99                                                                         ! 39+1 turns total.
'SYSTEM'
3
gnuplot <./gnuplot_Zspnprt_spinOscillation.gnu
gnuplot < ./gnuplot_Zplt_spinTilt.gnu
gnuplot <./gnuplot_Zplt_spinTilt_3D.gnu
'END'
'MARKER'  ProbAddSpinSphere_E                                          ! Just for edition purposes.
'END'
```

A gnuplot script to produce spin components versus turn, reading from zgoubi.SPNPRT.Out, Fig. 3.33:

```
# gnuplot_Zspnprt_spinOscillation.gnu
set xlabel "turn"; set ylabel "S_X,   S_Y,   S_Z"; set key b l ; nbtrj=3 # number of trajectories tracked
do for [it=1:nbtrj] {  unset label; set label sprintf("particle %3.5g",it) at 10, 0.8
plot [] [-1:1] \
  'zgoubi.SPNPRT.Out' every nbtrj::(it+2) u ($22):($13) w lp lw .3 pt 4 ps .8 lc rgb "red" ,\
  'zgoubi.SPNPRT.Out' every nbtrj::(it+2) u ($22):($14) w lp lw .3 pt 6 ps .8 lc rgb "blue" ,\
  'zgoubi.SPNPRT.Out' every nbtrj::(it+2) u ($22):($15) w lp lw .3 pt 8 ps .8 lc rgb "black" ; pause .5
set terminal postscript eps blacktext color  enh
set output sprintf('gnuplot_Zspnprt_spinOsc_trj%i.eps',it); replot; set terminal X11; unset output }
```

A gnuplot script to produce 2*D spin motion projection of Fig.* 3.33:

```
# gnuplot_Zplt_spinTilt.gnu
set xlabel "S_X"; set ylabel "S_Y"; set size ratio -1; set xrange [-1:1]; set yrange [-1:1]; set key t l
nbtrj=3 # number of trajectories tracked
do for [it=1:nbtrj] {  unset label; set label sprintf("particle %i",it) at -.9, .8
  plot 'zgoubi.plt' u ($19==it? $33 :1/0):($34) w lp lw .3 ps .2 lc rgb "blue";  pause .5
  set terminal postscript eps blacktext color  enh
  set output sprintf('gnuplot_Zplt_SX-SY_trj%i.eps',it); replot; set terminal X11; unset output }
```

A gnuplot script to produce the projection on a sphere of Figs. 3.33:

```
# gnuplot_Zplt_spinTilt_3D.gnu
set xlabel "X"; set ylabel "Y"; set zlabel "Z"; set xrange [-1:1]; set yrange [-1:1]; set zrange [-1:1]
set xyplane 0; set view equal xyz; set view 49, 339; unset colorbox
set urange [-pi/2:pi/2]; set vrange [0:2*pi]; set parametric; R = 1.   # radius of sphere
nbtrj=3 # number of trajectories tracked
do for [it=1:nbtrj] {  unset label; set label sprintf("  particle %i",it) at -1, .9, 1.
  splot  R*cos(u)*cos(v),R*cos(u)*sin(v),R*sin(u) w l lw .2 lc rgb "cyan" notit ,\
  'zgoubi.plt'   u ($19==it? $33 :1/0):($34):($35) w lp lw .2 ps .4 lc palette ;  pause .5
  set terminal postscript eps blacktext color  enh
  set output sprintf('gnuplot_Zplt_S3D_trj%i.eps',it); replot; set terminal X11; unset output }
```

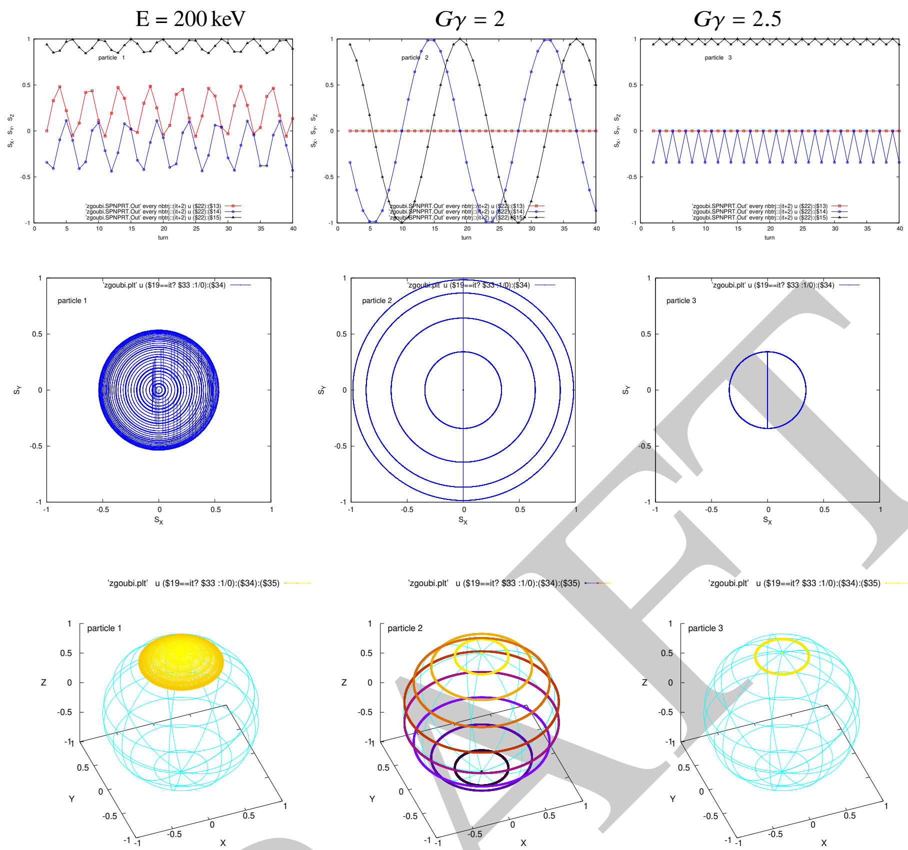

Fig. 3.33 Top row: spin coordinates versus turn; middle row: projection in the median plane (the segment between two consecutive circles materializes the location of the X-kick by SPINR); bottom row: projection on a sphere. $G\gamma = 1.793229$: far from an integer, **S** remains within a cone of reduced aperture. $G\gamma = 2$: the spin vector oscillates between up and down orientations, by 20 deg steps; it takes 180/20=9 orbits for the X-precession at SPINR to flip the spin; $G\gamma = 2.5$: the spin vector finds itself back in the (Y,Z) plane at the location of SPINR, after one orbit and a half-integer number of precessions; it alternates between vertical and 20 deg from vertical, after each orbit around the cyclotron

3.6 Weak Focusing

(a) Add a field index.

To the first order in R, in the median plane (Z=0) and noting $R = R_0 + dR$, $B_Z(R_0) = B_0$, $B_Z(R) = B$, the field writes (Sect. 3.2.2) $B(R) = B_0 + dR\ \frac{\partial B}{\partial R}\big|_{R_0}$. With $k = \frac{R_0}{B_0}\frac{\partial B}{\partial R}$ (Eq. 3.11) this yields

$$B(R) = B_0 + \frac{B_0}{R_0}\,k\,dR \qquad (3.38)$$

Assume the earlier 200 keV conditions as a reference, thus take

$R_0 = 12.9248888$ cm as the 200 keV radius, whereas $B_0 = B(R_0) = 5$ kG.

Take $k = -0.03$, a slow decrease of the field with R - proper to ensure appropriate vertical focusing with marginal impact on the radial extent of the cyclotron. For instance, with that index value the 5 MeV orbit is at a radius of 75.75467 cm (see OBJET in Tab. 3.3) (giving $B = 0.3235$ T along the orbit), whereas if k=0 then $R = 75.75467$ cm is the 6.8463 MeV orbit radius ($B = 0.3788$ T).

The field map is generated using a similar Fortran program to that of exercise 3.1 (see Tab. 3.1), *mutatis mutandis*, namely, introducing a reference radius R_0 and field index k. The resulting program is given in Tab. 3.14, it can be compiled and executed, as is, excerpts of the field data file so obtained are given in Tab. 3.15, a graph $B_Z(R, \theta)$ is given in Fig. 3.34. The orbit radius is assessed for three different energies, and appears to be in accord with theoretical expectation (Fig. 3.34-right). Comparison with Fig. 3.20-right shows the effect of the negative index on the radial distribution of the orbits, including a radius about 20% greater in the 5 MeV range. The input data file to find these trajectories is given in Tab. 3.16:

- the file defines an INCLUDE segment, #S_60degSectorIndx to #E_60degSectorIndx, used in subsequent exercises;

- the file is set to allow a preliminary test regarding the field map geneSectorMapIndex.out (as produced by the program given in Tab. 3.14), by computing three circular trajectories centered on the center of the map, at respectively 20 keV, 200 keV (the reference energy for the definition of the gradient index k) and 5 MeV (a large radius);

- note that once the FIT procedure is completed, `zgoubi` continues in sequence, so raytracing the 3 ions through the field map with, this time, IL set to 2 under TOSCA for stepwise particle data to be logged in zgoubi.plt.

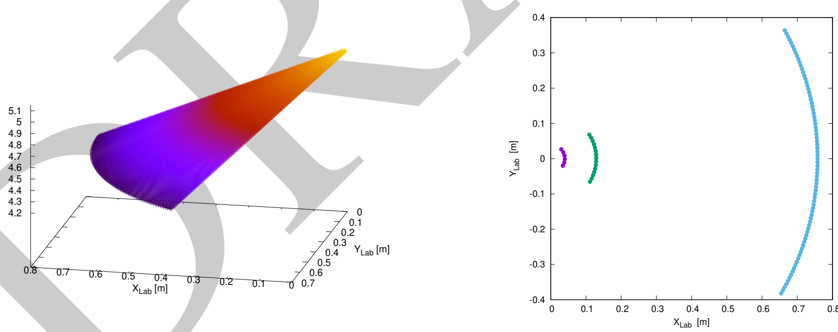

Fig. 3.34 Left: field map of a 60 deg magnetic sector with radial index, 76 cm radial extent. The field decreases from the center of the ring (at $(X_{\rm Lab}, Y_{\rm Lab}) = (0, 0)$). Right: three circular arc of trajectories over a sextant, at respectively from left to right: 0.02 MeV, 0.2 MeV (energy on the reference radius) and 5 MeV

Table 3.14 A Fortran program which generates a 60^o mid-plane field map with non-zero transverse field k. The field map it produces is logged in geneSectorMapIndex.out

```
C geneSectorMapIndex.f program
      implicit double precision (a-h,o-z)
      parameter (pi=4.d0*atan(1.d0), BY=0.d0, BX=0.d0, Z=0.d0)

      open(unit=2,file='geneSectorMapIndex.out')                              ! Field map storage file.

C------------ Hypotheses :
      AT = 60.d0  /180.d0*pi           ! Angular extent of field map. Can be changed 360, 60 deg, etc.).
      B0 = 5.d0 ;R0 = 12.9248888074d0             ! field at R0 (kG); 200keV radius (cm), B(R0)=B0=5kG.
      ak = -0.03d0                                                               ! Field index, defined at R0.
      Rmi=1.d0; Rma=76.d0; RM=50.d0  ! cm. Radial extent of field map; reference radius to define mesh.
      dR = 0.5d0 ; NR = NINT((Rma - Rmi)/dR)+1   ! R-distance between nodes in mesh. Number of R-nodes.
C                                            RdA=RM*dA is the distance between two nodes along R=RM arc,
      RdA = 0.5d0  !      given angle increment dA (dA is the "Delta theta" quantity in the main text).
      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-----------------------------------------------
      write(2,*) Rmi,dR,dA/pi*180.d0,dZ,
     >'     !  Rmi/cm, dR/cm, dA/deg, dZ/cm'
      write(2,*) '# Field map generated using geneSectorMapIndex.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,*) '# '
      write(2,*) '#     R*cosA              Z==0,         R*sinA'
     >//'            BY               BZ               BX   ix jr'
      write(2,*) '#      cm                 cm               cm '
     >//'            kG               kG               kG '
      do jr = 1, NR
        R = Rmi + dble(jr-1)*dR
        BZ = B0 + B0/R0 * ak * (R - R0)
        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 geneSectorMapIndex.out.'
      end
```

Table 3.15 First and last few lines of the field map file geneSectorMapIndex.out. The file starts with an 8-line header, the first one of which is effectively used by `zgoubi`, the following 7 are just comments

```
  1.      0.5      0.57142857142857140       0.          !  Rmi/cm, dR/cm, dA/deg, dZ/cm
# Field map generated using geneSectorMapIndex.f
# AT/rd,  AT/deg, Rmi/cm, Rma/cm, R0/cm, NR, dR/cm, NX, RdA/cm, dA/rd :
#   1.04719755E+00   6.0E+01   1.0E+00   7.60E+01   1.29248888E+01 151   5.0E-01 106  4.98665501E-01  9.97331001E-03
# For TOSCA:         106         151  1 22.1 1.  !IZ=1 -> 2D ; MOD=22 -> polar map ; .MOD2=.1 -> one map file
#
#     R*cosA             Z==0,         R*sinA           BY               BZ               BX   ix jr
#      cm                 cm             cm             kG               kG               kG
  1.00000000E+00  0.00000000E+00  0.00000000E+00  0.00000000E+00  5.13839448E+00  0.00000000E+00 1 1
  9.99950267E-01  0.00000000E+00  9.97314468E-03  0.00000000E+00  5.13839448E+00  0.00000000E+00 2 1
  9.99801073E-01  0.00000000E+00  1.99452974E-02  0.00000000E+00  5.13839448E+00  0.00000000E+00 3 1
  9.99552432E-01  0.00000000E+00  2.99154662E-02  0.00000000E+00  5.13839448E+00  0.00000000E+00 4 1
  9.99204370E-01  0.00000000E+00  3.98826594E-02  0.00000000E+00  5.13839448E+00  0.00000000E+00 5 1
 ................................
  4.05947602E+01  0.00000000E+00  6.42500229E+01  0.00000000E+00  4.26798081E+00  0.00000000E+00 102 151
  3.99519665E+01  0.00000000E+00  6.46516850E+01  0.00000000E+00  4.26798081E+00  0.00000000E+00 103 151
  3.93051990E+01  0.00000000E+00  6.50469164E+01  0.00000000E+00  4.26798081E+00  0.00000000E+00 104 151
  3.86545219E+01  0.00000000E+00  6.54356779E+01  0.00000000E+00  4.26798081E+00  0.00000000E+00 105 151
  3.80000000E+01  0.00000000E+00  6.58179307E+01  0.00000000E+00  4.26798081E+00  0.00000000E+00 106 151
```

A gnuplot script to obtain Fig. 3.34:

```
# PLOT THE FIELD MAP:
set xtics mirror ; set ytics mirror ; set xlabel "X_{Lab} [m]" ; set ylabel "Y_{Lab} [m]" ; cm2m = 0.01
set zrange [:5.15] ; set view 66, 192 ; unset colorbox
splot "geneSectorMapIndex.out" u ($1 *cm2m):($3 *cm2m):($5) w p lc palette notit ; pause 1

# PLOT THREE TRAJECTORIES
set xtics ;  set ytics ; set xlabel "X_{Lab}  [m]" ; set ylabel "Y_{Lab}  [m]" ; cm2m = 0.01 ; set size ratio 1
plot for [trj=1:3] \
  "zgoubi.plt" u ($19==trj ? $10*cm2m*cos($22) :1/0):($10*cm2m*sin($22)) w p pt 7 ps .6 notit ; pause 1
```

Table 3.16 Simulation input data file FieldMapSectorIndex.inc: a file to test trajectories for a field map with radial index. This file also defines the INCLUDE segment between the LABEL_1s #S_60degSectorIndx and #E_60degSectorIndx

```
FieldMapSectorIndex.inc
! Uniform field sector with index. INCLUDE file FieldMapSectorIndex.inc
'MARKER'   FieldMapSectorIdx_S                                         ! Just for edition purposes.
'OBJET'
64.62444403717985                     ! Reference Brho ("BORO" in the users' guide) -> 200keV proton.
2
3 1
4.003593 0. 0. 0. 0. 0.3162126 'o'          ! p[MeV/c]= 6.126277, Brho[kG.cm]=20.435, kin-E[MeV]=0.02.
12.92488 0. 0. 0. 0. 1.        'o'    ! Reference ; p[MeV/c]=193.739, Brho[kG.cm]=BORO, kin-E[MeV]=0.2.
75.75467 0. 0. 0. 0. 5.0063900 'o'              ! p[MeV/c]=969.934, Brho[kG.cm]=323.535, kin-E[MeV]=5.
1 1 1
'MARKER'   #S_60degSectorIndx
'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.
106 151 1 22.1 1.            ! IZ=1 for 2D map; MOD=22 for polar frame; .MOD2=.1 if only one map file.
geneSectorMapIndex.out
0 0 0 0       ! Possible vertical boundaries within the field map, to start/stop stepwise integration.
2
1.0                               ! Integration step size. Small enough for orbits to close accurately.
2                                                                       ! Magnet positionning option.
0. 0. 0. 0.                                                                    ! Magnet positionning.
'MARKER'   #E_60degSectorIndx
 'FIT2'                                    ! This matching procedure finds the closed orbit radius.
3   nofinal
2 30 0 [2.,10.]        ! Variable : Y_0, trajectory 1
2 40 0 [10.,15.]       ! Variable : Y_0, trajectory 2
2 50 0 [50.,80.]       ! Variable : Y_0, trajectory 3
3  1e-20  9999         ! Penalty; max numb of calls to function
3.1 1 2 #End 0. 1. 0    ! Constraint :  Y_final=Y_0, trajectory 1
3.1 2 2 #End 0. 1. 0    ! Constraint :  Y_final=Y_0, trajectory 2
3.1 3 2 #End 0. 1. 0    ! Constraint :  Y_final=Y_0, trajectory 3

!   Carry on with coordinates as found, yet with IL=2 under TOSCA so to log trajectories in zgoubi.plt.
'TOSCA'
0 2       ! IL=2: 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.
106 151 1 22.1 1.            ! IZ=1 for 2D map; MOD=22 for polar frame; .MOD2=.1 if only one map file.
geneSectorMapIndex.out
0 0 0 0       ! Possible vertical boundaries within the field map, to start/stop stepwise integration.
2
1.0                               ! Integration step size. Small enough for orbits to close accurately.
2                                                                       ! Magnet positionning option.
0. 0. 0. 0.                                                                    ! Magnet positionning.
'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'   FieldMapSectorIdx_E                                         ! Just for edition purposes.
'END'
```

(b) R-dependence of orbit rigidity.

The method is similar to exercise 3.1-(b) (see Tab. 3.4.): FIT finds the closed orbit radius R for a given ion rigidity, and REBELOTE is used to repeat for a series of different momenta, 20 here. The input data file for this exercise is given in Tab. 3.17, it includes a 21 ion 1-turn raytracing, in sequence with the previous 21-orbit finding.

Raytracing outcomes for $k = -0.03$, $R_0 = R(E = 200\,\text{keV}) = 12.924888\,\text{cm}$, $B_0 = B(R_0) = 0.5\,\text{T}$ are given in Fig. 3.35, together with theoretical expectation (with B(R) from Eq. 3.7)

$$\text{Rigidity} \quad BR(R) = B_0 \left(1 + \frac{R - R_0}{R_0} k\right) R \tag{3.39}$$

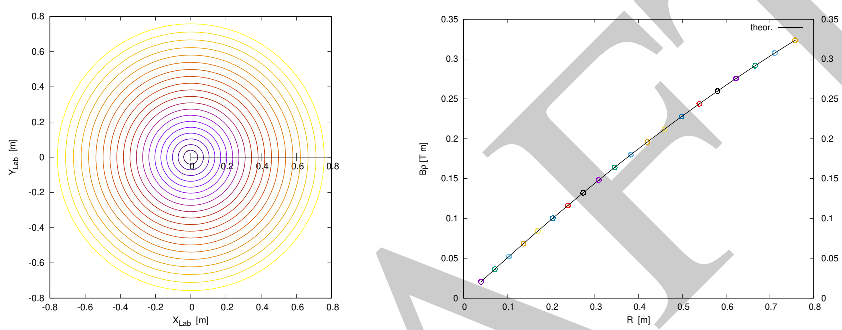

Fig. 3.35 Case of field index k=-0.03. Left: closed orbits at a series of different rigidities. Right: comparison of $B\rho(R)$ from raytracing outcomes (markers) and from theory (solid line, Eq. 3.39)

(c) Paraxial motion.

A proton with energy 1 MeV is considered, here. DIPOLE [16, *lookup* INDEX] is used rather than a field map, so to allow to freely change the k index value (using TOSCA instead would require computing a new field map when changing k).

The input data for a 60 deg sector are given in Tab. 3.18, essentially a copy of the uniform dipole field case of Tab. 3.6 in which the index value $k = -0.03$ has been added (line 3 under DIPOLE). The input data sequence for multiturn trajectory computation around the cyclotron is given in Tab. 3.19: in a first stage, orbit finding is performed by FIT, for 1 MeV energy; in a subsequent second stage, 4 protons with their initial horizontal coordinates taken on the closed orbit, and differing by their initial vertical take-off angle, are tracked over 120 sectors, *i.e.*, 20 turns around the ring.

Fig. 3.36 displays the vertical sine motion. Stronger index (k closer to -1) results in stronger vertical focusing, hence more oscillations as expected from Eq. 3.18 and smaller motion amplitude as expected from Eq. 3.17. The latter can be written

Table 3.17 Simulation input data file: scan orbits for momentum dependence. Two problems are stacked, executed in sequence: in a first stage FIT finds a closed orbit, whose coordinates are logged in initialRs.fai file when FIT is completed, following what REBELOTE repeats for an additional 20 momenta; in a second stage OBJET grabs the 21-set of ion coordinates from initialRs.fai and these ions are raytraced over 6 sectors, *i.e.*, one full turn. The INCLUDE file FieldMapSectorIndex.inc is taken from Tab. 3.16

```
Uniform field sector with index. Scan orbits.
'MARKER'   scanSectorIdx_S                                                 ! Just for edition purposes.
'OBJET'
64.62444403717985                       ! Reference Brho ("BORO" in the users' guide) -> 200keV proton.
2
1 1                                                                                     ! Just one ion.
4.0039    0. 0. 0. 0. 0.3162126 'o'          ! p[MeV/c]= 6.126277, Brho[kG.cm]=20.435, kin-E[MeV]=0.02.
1
'FAISCEAU'                                             ! Local particle coordinates logged in zgoubi.res.
'INCLUDE'
1
./FieldMapSectorIndex.inc[#S_60degSectorIndx:#E_60degSectorIndx]
 'FIT'                                       ! This matching procedure finds the closed orbit radius.
1   nofinal
2 30 0 [3.,80.]         ! Variable : Y_0
1  1e-15  99            ! Penalty; max numb of calls to function
3.1 1 2 #End 0. 1. 0    ! Constraint :  Y_final=Y_0
'FAISTORE'
initialRs.fai                                                      ! Log coordinates in initialRs.fai.
1
'REBELOTE'              ! A do-loop. Repeat the above, after changing particle rigidity to a new value.
20 0.2  0 1            ! 20 diffrnt rigidities; I/O options; coordinates as from OBJET; changes follow:
1                                 ! Parameter 35 to be changed, in OBJET: relative momentum, namely,
OBJET 35  0.3162126:5.0063900                                ! for energy scan from 0.02 MeV to 5 MeV.

'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
'FAISCEAU'                                             ! Local particle coordinates logged in zgoubi.res.
'INCLUDE'
1
6* ./FieldMapSectorIndex.inc[#S_60degSectorIndx:#E_60degSectorIndx]                 ! INCLUDE 6 times.
'SYSTEM'
2
gnuplot <./gnuplot_Zplt_orbits.gnu                                 ! Plot orbits around the cyclotron.
gnuplot <./gnuplot_Zplt_scanBrho.gnu                                                  ! Plot R(Brho).
'MARKER'   scanSectorIdx_E                                                 ! Just for edition purposes.
'END'
```

A gnuplot script to obtain orbits, Fig. 3.35:

```
set xtics ;  set ytics ; set xlabel "X_{Lab}  [m]" ; set ylabel "Y_{Lab}  [m]" ; cm2m = 0.01; set polar; set size ratio 1
unset colorbox; pi = 4.*atan(1.); TOSCA1=12; dT=3 # number of 2nd TOSCA & increment in zgoubi.plt listing
plot for [trj=2:21] \
  "zgoubi.plt" u ($19==trj ? $22+($42-TOSCA1)/dT*pi/3 :1/0):($10*cm2m ):($19) w l lw 2 lc palette notit ; pause 1
```

A gnuplot script to obtain $B\rho(R)$, *Fig.* 3.35:

```
set xtics ;  set ytics nomirror ; set y2tics; set xlabel "R  [m]" ; set ylabel "B{/Symbol r} [T m]"
B0=0.5; R0=12.924888e-2; k=-0.03; Brho(x)=  B0* (1.+  (x-R0)/R0* k )*x ; kGcm2Tm=1e-3;  cm2m = 0.01
plot for [trj=2:21] \
  "zgoubi.plt" u ($19==trj? $10*cm2m :1/0):($40*(1.+$2)*kGcm2Tm) w p pt 6 ps 1.2 notit ,\
  Brho(x) axes x1y2 w l lw 2 lc rgb "black" tit "theor." ; pause 1
```

Table 3.18 Simulation input data file sectorWithIndex.inc: definition of a dipole with index, case of analytical field modeling, namely here k=-0.03 and reference radius $R_0 = 50$ cm. Definition of the [#S_60degSectorWIdx:#E_60degSectorWIdx] segment

```
# sectorWithIndex.inc
'MARKER'    #S_60degSectorWIdx                                ! 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.
30. 5. -0.03 0. 0.  ! Reference azimuthal angle ACN; BM field at R0; indices, N (=k=-0.03) at R0=50cm.
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
0.   0.                                                                                     ! EFB 2.
4  .1455   2.2670  -.6395  1.1558  0. 0.  0.
-30. 0.  1.E6  -1.E6  1.E6  1.E6
0. 0.                                                                             ! EFB 3 (unused).
0  0.      0.      0.      0.      0. 0.  0.
0. 0.  1.E6  -1.E6  1.E6  1.E6 0.
4   10.
0.5                          ! Integration step size. Small enough for orbits to close accurately.
2  0. 0. 0. 0.                                                  ! Magnet positionning RE, TE, RS, TS.
'MARKER'    #E_60degSectorWIdx                                ! Label should not exceed 20 characters.
'END'
```

$$Z(s) = P_0 \frac{R_0}{\sqrt{-k}} \sin \frac{\sqrt{-k}}{R_0}(s - s_0) \quad \text{and} \quad \hat{Z} = P_0 \frac{R_0}{\sqrt{-k}} \tag{3.40}$$

Note that this vertical oscillation results in a modulation of the field along the trajectory (see question (d) of this exercise) which results in a radial oscillation, a second order Y-Z coupling effect (extremely weak), displayed in Fig. 3.37.

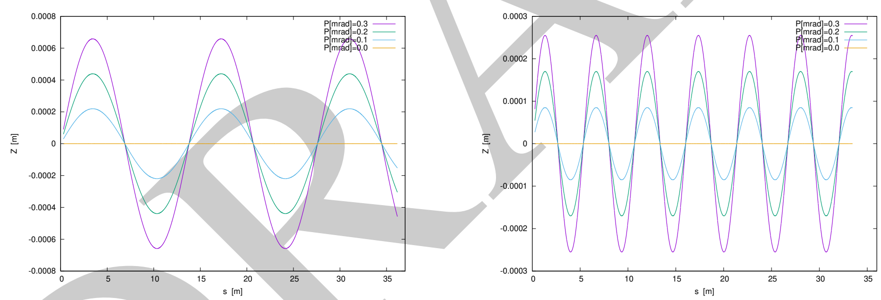

Fig. 3.36 Vertical sine motion over a few turns around the cyclotron, at 1 MeV. Vertical take-off angles are $P_0 = 0,\ 0.1,\ 0.2,\ 0.3$ mrad. Left: k=-0.03, $\nu_Z = \sqrt{0.03} \approx 0.173$ oscillations per turn; right: for k=-0.2, $\nu_Z = \sqrt{0.2} \approx 0.447$ oscillations per turn

(d) Magnetic field.

The magnetic field experienced by 1 MeV protons with four different take-off angles P_0 (Fig. 3.36), along their respective trajectories, case of an index value $k = -0.03$, is displayed in Fig. 3.38. It is essentially constant as expected.

Table 3.19 Simulation input data file: scan orbits for momentum dependence; the file actually stacks two simulations, executed in sequence; the second simulation uses data produced by the first one, as follows. The first part of the file finds the closed orbits, they depend on the vertical excursion and are not exactly zero, due to the field index; closed orbit coordinates so found are logged in initialRs.fai when FIT is completed. The second part of the file starts at the second occurrence of OBJET which reads initial particle coordinates from initialRs.fai and tracks these particles through a sequence of 120 sector dipoles, *i.e.*, 20 turns. The [#S_60degSectorWIdx:#E_60degSectorWIdx] segment of Tab. 3.18 is INCLUDEd, here

```
Uniform field sector with index. Scan orbits.
'MARKER'   1MeVVMotion_S                                                    ! Just for edition purposes.
!                                              First stage: find closed orbit at 1 MeV, for some k value.
'OBJET'
64.62444403717985                      ! Reference Brho ("BORO" in the users' guide) -> 200keV proton.
1.1
1   1   1   4    1   1
0.  1.  0.  0.1  0.  1.
30.107900 0.   0.  0.   0. 2.2365445724 'm'             ! 1 MeV proton -> Brho/Brho_ref = 2.2365445724.
'INCLUDE'
1
./sectorWithIndex.inc[#S_60degSectorWIdx:#E_60degSectorWIdx]             ! DIPOLE case R0=50cm k=-0.03.
 'FIT'                                         ! This matching procedure finds the closed orbit radius.
1   nofinal
2  40  0  .9                                             ! Variable : Y_0. Variation can be up to 90%.
1  1e-15  99                                                ! Penalty; max numb of calls to function.
3.1 1 2 #End 0. 1. 0                                                    ! Constraint :  Y_final=Y_0.
'FAISTORE'
initialRs.fai                                                      ! Log coordinates in initialRs.fai.
1
!                                               Second stage: raytrace the four particles over 20 turns.
'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
'FAISCEAU'                                         ! Local particle coordinates logged in zgoubi.res.
'INCLUDE'
1
120 * sectorWithIndex.inc[#S_60degSectorWIdx:#E_60degSectorWIdx]      ! INCLUDE 120 sectors (20 turns).
'FAISCEAU'                                         ! Local particle coordinates logged in zgoubi.res.
'SYSTEM'
2
gnuplot <./gnuplot_Zplt_1MeVVMotion.gnu
gnuplot <./gnuplot_Zplt_1MeVBField.gnu
'MARKER'   1MeVVMotion_E                                                     ! Just for edition purposes.
'END'
```

A gnuplot script to obtain Figs. 3.36, 3.37:

```
# gnuplot_Zplt_1MeVVMotion.gnu
set xtics ;  set ytics ; set xlabel "s  [m]" ; set ylabel "Z  [m]" ; cm2m = 0.01; unset colorbox ; set xrange [:36]
plot for [trj=4:1:-1] \
  "zgoubi.plt" u ($19==trj && $42>10? $14*cm2m :1/0):($12*cm2m ):($19) w l lw 2 tit "P[mrad]=0.".(trj-1) ; pause 1
```

A gnuplot script to obtain Fig. 3.38:

```
# gnuplot_Zplt_1MeVBField.gnu
set xtics;  set ytics; set xlabel "s  [m]"; set ylabel "Y  [m]"; cm2m = 0.01; unset colorbox
plot for [trj=4:1:-1] \
  "zgoubi.plt" u ($19==trj && $42>10? $14*cm2m :1/0):($10*cm2m ):($19) w l lw 2 tit "P[mrad]=0.".(trj-1) ; pause 1
```

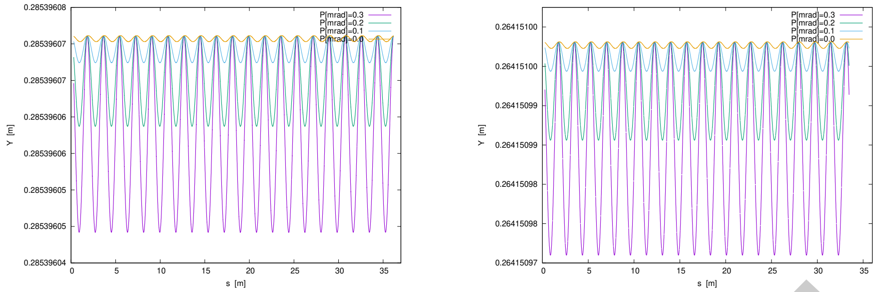

Fig. 3.37 Horizontal motion at 1 MeV, 20 turns around the cyclotron, for vertical take-off angles $P_0 = 0,\ 0.1,\ 0.2,\ 0.3$ mrad. Left: k=-0.03, $\nu_R = \sqrt{1 + 0.03} \approx 1.015$ oscillations per turn; right: for k=-0.2, $\nu_R = \sqrt{1 + 0.2} \approx 1.095$ oscillations per turn

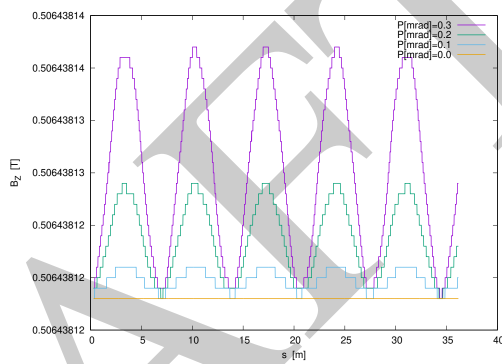

Fig. 3.38 Magnetic field experienced by 1 MeV protons with four different take-off angles P_0 (Fig. 3.36), along their respective trajectories. Case $k = -0.03$. The stepwise structure of these $B_Z(s)$ curves is due to the fact that field variations are at the limit of computer truncation related accuracy

3.7 Loss of Isochronism

In order to scan $T_{\rm rev}(R)$ for different k values, DIPOLE [16, *lookup* INDEX] is used here, as it allows to easily vary k and subsequently find the closed orbit using FIT. The method of exercise 3.6 is employed to obtain a scan. The input data file of Tab. 3.17 is a good starting point to do this exercise, changing the INCLUDE to account for DIPOLE instead of a field map modeling using TOSCA: the proper INCLUDE formatting can be reproduced from Tab. 3.19. IL under DIPOLE may be set at IL=0 as zgoubi.plt is not used here. Introduce FAISTORE to store local particle data after FIT (that includes time of flight, the quantity of interest here, which requires PARTICUL[PROTON] following OBJET).

The new input data file so built for this simulation, is given in Tab. 3.20.

This input data file is run for four different k values, namely, under DIPOLE (*cf.* Tab. 3.18), the line "30. 5. -0.03 0. 0." is successively changed to $\begin{cases} 30.\ 5.\ 0\ 0.\ 0. \\ 30.\ 5.\ -0.5\ 0.\ 0. \\ 30.\ 5.\ -0.95\ 0.\ 0. \end{cases}$. The corresponding zgoubi.fai files are saved under dedicated copies for plotting, see "gnuplot script gnuplot_Zfai_scanTrev.gnu" at the bottom of Tab. 3.20.

The results of these $T_{\rm rev}$ scans are displayed in Fig. 3.39. In the case $k = 0$ the loss of isochronism is only due to the relativistic change of the mass, a non-zero k

Table 3.20 Simulation input data file: scan revolution time. The [#S_60degSectorWIdx:#E_60degSectorWIdx] segment of Tab. 3.18 is INCLUDEd, here

```
Uniform field sector with index. Scan orbits.
'MARKER'   isoChroLoss_S                                                    ! Just for edition purposes.
'OBJET'
64.62444403717985                       ! Reference Brho ("BORO" in the users' guide) -> 200keV proton.
2
1 1                                                                                       ! Just one ion.
4.0039    0. 0. 0. 0. 0.3162126 'o'          ! p[MeV/c]= 6.126277, Brho[kG.cm]=20.435, kin-E[MeV]=0.02.
1
'PARTICUL'                                            ! Necessary as time of flight computation is needed,
PROTON                                             ! otherwise, by default \zgoubi\ only requires rigidity.
'INCLUDE'
1
./sectorWithIndex.inc[#S_60degSectorWIdx:#E_60degSectorWIdx]             ! DIPOLE case R0=50cm k=-0.03.
'FIT2'                                              ! This matching procedure finds the closed orbit radius.
1   nofinal
2 30 0 [0.5,80.]         ! Variable : Y_0
1  1e-15  99            ! Penalty; max numb of calls to function
3.1 1 2 #End 0. 1. 0    ! Constraint :  Y_final=Y_0
'FAISCEAU'                                               ! Local particle coordinates logged in zgoubi.res.
'FAISTORE'
zgoubi.fai
1
'REBELOTE'              ! A do-loop. Repeat the above, after changing particle rigidity to a new value.
20 0.2  0 1            ! 20 diffrnt rigidities; I/O options; coordinates as from OBJET; changes follow:
1                                 ! Parameter 35 to be changed, in OBJET: relative momentum, namely,
OBJET 35  0.3162126:5.00639    ! Acceleration to 5MeV. Commented here, for use in subsequent exercises.
! OBJET 35  0.3162126:2.2365445724    ! Substitute to previous, for energy scan from 0.02 MeV to 1 MeV.
'SYSTEM'
1
gnuplot <./gnuplot_Zfai_scanTrev.gnu                                                ! Plot revolution time.
'MARKER'   isoChroLoss_E                                                    ! Just for edition purposes.
'END'
```

A gnuplot script to obtain Fig. 3.39:

```
# gnuplot_Zfai_scanTrev.gnu
set xtics ;  set ytics nomirror ; set y2tics; set xlabel "R  [m]" ; set ylabel "T_{rev} [{/Symbol m}s]"
cm2m = 0.01; nSec=6; set y2label "T_{rev}  at  k=0[{/Symbol m}s]" ; set key c r
plot "zgoubi_k0.fai"    u ($10 *cm2m):($15 * nSec) w lp pt 4 ps 1.2 lc rgb "black" tit "k=0" ,\
     "zgoubi_k0.fai"    u ($10 *cm2m):($15 * nSec) axes x1y2 w lp pt 4 ps 1.2  lc rgb "black" notit ,\
     "zgoubi_k0.03.fai" u ($10 *cm2m):($15 * nSec) w lp  pt 7 ps 1.2 tit "k=-03" ,\
     "zgoubi_k0.5.fai"  u ($10 *cm2m):($15 * nSec) w lp  pt 8 ps 1.2 tit "k=-5" ,\
     "zgoubi_k0.95.fai" u ($10 *cm2m):($15 * nSec) w lp  pt 9 ps 1.2 tit "k=-95" ; pause 1
```

augments the effect. The loss of isochronism is the cause of the $\approx 20\,\mathrm{MeV}$ proton energy limit of the classical cyclotron.

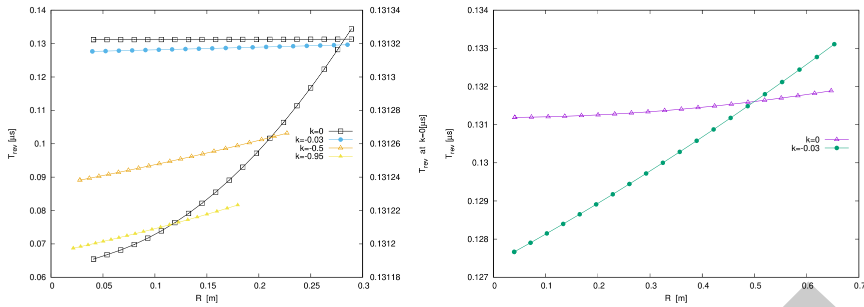

Fig. 3.39 A scan of the revolution time, from 0.02 to 1 MeV, and its dependence on the field index k. The right vertical axis only concerns the case $k = 0$ where the change in revolution time is weak and only due to the mass increase (in $T_{\rm rev} = 2\pi\gamma m_0/qB$). The right graph shows, up to 5 MeV, the relatively important contribution of the focusing index, even a weak k=-0.03, compared to the effect of the mass increase (k=0 curve). Markers are from raytracing, solid lines are from theory

3.8 Ion Trajectories

A `zgoubi` data file is set up for computation of particle trajectories, taking a field value on reference radius of $B_0(R_0) = 0.5$ T, and reference energy 200 keV (proton). These hypotheses determine the reference radius value. DIPOLE [16, *lookup* INDEX] is used (Tab. 3.21), for its greater flexibility in changing magnet parameters, field and radial field index amongst other, compared to using TOSCA and a field map.

(a) Transverse motion.

It first has to be checked that there is consistency between initial orbital radius Y_0 in OBJET at 200 keV proton energy and the value of the reference radius R_0 in DIPOLE (Eq. 3.35). Its theoretical value is $R_0 = BORO/5[kG] = 12.924889$ cm, a closed orbit finding using FIT can be performed, or it can be referred to the solutions of earlier exercises, to check agreement with raytracing outcomes.

(b) Wave numbers at 1 and 5 MeV.

These considerations result in the input data file given in Tab. 3.22, to compute multiturn trajectories. ; note that $R_0 = 12.924889$ cm therein, whereas a value of $R_0 = 50$ cm may be taken instead in other exercises. Field index derivatives k', k'', ... are taken null in the present exercise.

Three particles with paraxial radial and axial motions are raytraced over a few turns. Their starting radius is the closed orbit radius for the respective energies, while a 0.1 mrad take-off angle is imparted to each particle both vertically and horizontally.

The value of the focusing index k_E at an energy E can be expressed in terms of DIPOLE data which are, the index value k at R_0 (Eq. 3.11), reference radius R_0, and field $B_0 = B_Z(R_0)$, namely,

$$k_E = \frac{R_E}{B_E}\frac{\partial B}{\partial R} = \frac{R_0 + \Delta R}{B_0 + \Delta B}\frac{\partial B}{\partial R} \approx k\,\frac{1 + \Delta R/R_0}{1 + k\Delta R/R_0} \approx k\left[1 + (1-k)\frac{\Delta R}{R_0}\right]$$

Table 3.21 Input data file 60DegSectorR200.inc: it defines DIPOLE as a sequence segment comprised between the "LABEL_1" type labels [16, Sect. 7.7] #S_60DegSectorR200 and #E_60DegSectorR200. DIPOLE here, has an index $k = -0.03$, reference radius $R_0 \equiv R_0(E_k = 200\,\text{keV}) = 12.924888$ cm and $B_0 = B(R_0) = 0.5$ T. Note that (i) this file can be run on its own: it has been designed to provide the transport MATRIX of that DIPOLE; (ii) for the purpose of some of the exercises, IL=2 under DIPOLE, optional, causes the printout of particle data in zgoubi.plt, at each integration step (this is at the expense of CPU time, and memory volume)

```
60DegSectorR200.inc
'OBJET'
64.62444403717985                                                                              ! 200keV proton.
5
0.01 0.001 0.01 0.001 0. 0.0001
12.9248888074 0.  0.  0. 0. 1.                                                   ! 200keV. R=Brho/B=*/.5.
'DIPOLE'   #S_60DegSectorR200                                  ! Analytical modeling of a dipole magnet.
2     ! IL=2, purpose: log stepwise particle data in zgoubi.plt. Avoid if unused: I/Os take CPU time.
60. 12.924888                                                    ! Sector angle AT; reference radius R0.
30.  5. -0.03  0. 0.            ! Reference azimuthal angle ACN; BM field at R0; 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
0.   0.                                                                                     ! EFB 2.
4  .1455   2.2670  -.6395  1.1558  0. 0.  0.
-30. 0.  1.E6  -1.E6  1.E6  1.E6
0. 0.                                                                              ! EFB 3 (unused).
0  0.      0.      0.      0.      0. 0.  0.
0. 0.  1.E6  -1.E6  1.E6  1.E6 0.
4   10.
0.5                            ! Integration step size. Small enough for orbits to close accurately.
2  0. 0. 0. 0.                                                     ! Magnet positionning RE, TE, RS, TS.
'FAISCEAU'  #E_60DegSectorR200
'MATRIX'
1 0
'END'
```

Table 3.22 Simulation input data file: raytrace a few turns around the cyclotron, three particles with different momenta, and 0.1 mrad horizontal and vertical take-off angles. The INCLUDE segment is taken from Tab. 3.21

```
'MARKER'   ProbProjTraj_S
 'OBJET'
64.62444403717985                   ! Reference Brho ("BORO" in the users' guide) -> 200keV proton.
2
3 1
12.924888 0.1 0. 0.1 0. 1. 'o'          ! A particle with kin-E=0.2 MeV and 0.1 mrad take-off angles.
30.107898 0.1 0. 0.1 0. 2.2365445 'm'         ! p[MeV/c]=433.306, Brho[kG.cm]=144.535, kin-E[MeV]=1.
75.754671 0.1 0. 0.1 0. 5.0063900 'o'         ! p[MeV/c]=969.934, Brho[kG.cm]=323.535, kin-E[MeV]=5.
1 1 1
'INCLUDE'
1
6* 60DegSectorR200.inc[#S_60DegSectorR200:#E_60DegSectorR200]     ! 6 sectors for an overall 360 deg.
'REBELOTE'
 9 0.1 99                                                          ! There will be a total of 9+1=10 tunrs.
'SYSTEM'
1
gnuplot < ./gnuplot_Zplt_traj.gnu                           ! Plot the projected Y(s) and Z(s) motions.
'MARKER'   ProbProjTraj_E
'END'
```

A gnuplot script to obtain Fig. 3.41:

```
# gnuplot_Zplt_traj.gnu
set xtics nomirror; set x2tics; set ytics; set xlabel 's  /C_E '; set ylabel 'Y  [cm]'
set palette defined ( 1 "red", 2 "blue", 3 "black" ) ; unset colorbox
array R[3]; R[1]=0.12924888; R[2]=0.301078986; R[3]=0.75754671; pi = 4.*atan(1.); cm2m = 0.01
sector1=3    # number (NOEL) of 1st DIPOLE in \zgoubi,res (col. 42 in zgoubi.plt)
# in zgoubi.plt, col. 19: particle number; col. 42: keyword number; col. 14: distance; col. 10: Y ; col. 12: YZ
plot for [sector=1:6] for [trj=1:3] 'zgoubi.plt' u ($19==trj && $42==sector1+2*(sector-1)? $14*cm2m/(2.*pi*R[$19]) :1/0) \
:($10*cm2m-R[trj]):($19) w p ps .2 lc palette notit ; pause 1

set ylabel 'Z  [cm]'; plot for [sector=1:6] for [trj=1:3] 'zgoubi.plt' u ($19==trj && $42==sector1+2*(sector-1)? $14*cm2m \
/(2.*pi*R[$19]) :1/0):($12):($19) w p ps .2 lc palette notit ; pause 1
```

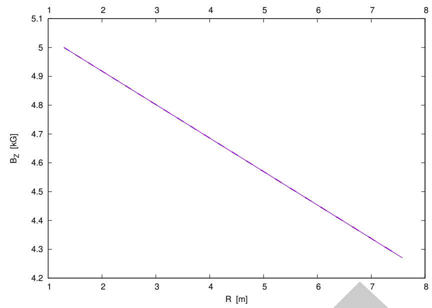

Fig. 3.40 In DIPOLE field model (Eq. 3.35), $\frac{\partial B}{\partial R}$ is constant: this graph shows the linear decrease of the field $B_Z(R)$ (Eq. 3.38), obtained from the raytracing of particles circulating in the median plane on orbits spanning a 0.2 to 5 MeV energy range

with ΔR assumed small, $\partial B/\partial R = kB_E/R_E$ an energy independent quantity, and the index E denoting a quantity taken at the reference energy. The latter property is illustrated in Fig. 3.40, produced using the input data file of Tab. 3.23.

Table 3.23 Simulation input data file for a magnetic field scan. The INCLUDE segment is taken from Tab. 3.21

```
Field and derivative dB/dR, as a finction of R
'MARKER'   ProbProjTrajB_S
'OBJET'
64.62444403717985                      ! Reference Brho ("BORO" in the users' guide) -> 200keV proton.
2
1 1                                                                                    ! Just one ion.
12.924888 0.1 0. 0.1 0. 1. 'o'           ! A particle with kin-E=0.2 MeV and 0.1 mrad take-off angles.
1
'INCLUDE'
1               ! IL=2 is necessary under DIPOLE, for step-by-step log of particle data in zgoubi.plt.
60DegSectorR200.inc[#S_60DegSectorR200:#E_60DegSectorR200]                    ! One sector is enough.
'FIT'
1
2 30 0 [12,80]                   ! Vary particle's Y0 at OBJET, to have it match its D (=Brho/BORO).
1  1e-20
3.1 1 2 #End 0. 1. 0                                                         ! Consrain Y_final=Y0.
'REBELOTE'
25 0.1 0 1                                    ! Scan parameter 35 (relative rigidity, D) in OBJET.
1
OBJET 35 1:5.00639                  ! Scan relative rigidity D from 1 (200 keV) to 5.0063900 (5 MeV).
'SYSTEM'
1
gnuplot < ./gnuplot_Zplt_field.gnu                               ! Plot B(R), as read fron zgoubi.plt.
'MARKER'   ProbProjTrajB_E
'END'
```

A gnuplot script to obtain Fig. 3.40:

```
# gnuplot_Zplt_field.gnu
set xtics nomirror; set x2tics; set ytics; set xlabel 's  /C_E '; set ylabel 'Y  [cm]'
set palette defined ( 1 "red", 2 "blue", 3 "black" ) ; unset colorbox
array R[3]; R[1]=0.12924888; R[2]=0.301078986; R[3]=0.75754671; pi = 4.*atan(1.); cm2m = 0.01
sector1=3    # number (NOEL) of 1st DIPOLE in \zgoubi,res (col. 42 in zgoubi.plt)
# in zgoubi.plt, col. 19: particle number; col. 42: keyword number; col. 14: distance; col. 10: Y ; col. 12: YZ
plot  for [i=1:6] for [trj=1:3]
  'zgoubi.plt' u ($19==trj && $42==sector1 +2*(i-1) ? $14*cm2m /(2.*pi*R[$19]) :1/0) \
  :($10*cm2m-R[trj]):($19) w p ps .2 lc palette notit ; pause 1

set ylabel 'Z  [cm]' ;
plot  for [i=1:6] for [trj=1:3]
  'zgoubi.plt' u ($19==trj && $42==sector1 +2*(i-1) ? $14*cm2m \
  /(2.*pi*R[$19]) :1/0):($12):($19) w p ps .2 lc palette notit ; pause 1
```

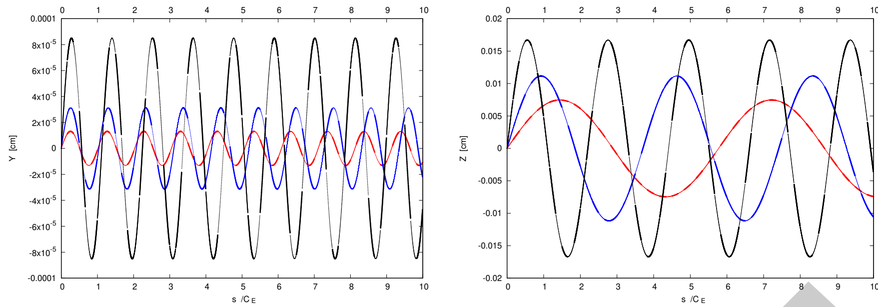

Fig. 3.41 Radial (left) and axial (right) paraxial motion around respectively the 200 keV (smallest amplitude), 1 MeV (intermediate) and 5 MeV (greatest amplitude) closed orbit (the latter is circular, in the median plane, with radius respectively $R_{200\,keV} = 12.924888$ cm, $R_{1\,MeV} = 30.107898$ cm and $R_{5\,MeV} = 75.754671$ cm). The horizontal axis in this graph is s/C_E: path length over closed orbit circumference at energy E, the vertical axis is the motion excursion

The resulting radial and axial motions over 10 turns are displayed in Fig. 3.41, which also illustrates, for paraxial motion at some reference energy, the energy dependence of the focusing strength (or wave number) and of the motion amplitude.

Table 3.24 Wave numbers, from numerical raytracing (columns denoted "ray-tr."), from theory, and from discrete Fourier transform ('DFT' cols.) from a multi-turn tracking

E (MeV)	k_E	ray-tr.	$\nu_R = \sqrt{1+k}$	DFT	ray-tr.	$\nu_Z = \sqrt{-k}$	DFT
0.2	-0.03	0.98520	0.9849	0.98513	0.17320	0.1732	0.17321
1	-0.07279	0.96187	0.96292	0.96291	0.26980	0.26979	0.26981
5	-0.20586	0.89083	0.89115	0.89115	0.45326	0.45371	0.45371

An estimate of the wave numbers can be obtained as the inverse of the number of turns per oscillation, namely,

$$\nu_R = \left.\frac{C_E}{\Delta s_M}\right|_E \quad \text{and} \quad \nu_Z = \left.\frac{C_E}{\Delta s_M}\right|_E$$

with Δs_M the measured distance between two consecutive maxima in the sinusoid of concern in Fig. 3.41, C_E the closed orbit length for the energy of concern. Both quantities are obtained from motion records in zgoubi.plt. This yields the values of Tab. 3.24, where they are compared with the theoretical expectations, namely (Eq. 3.18), $\nu_R = \sqrt{1+k}$ and $\nu_Z = \sqrt{-k}$.

The maximum amplitude of the oscillation is obtained from zgoubi.plt records as well, this yields the results of Tab. 3.25. For comparison, the theoretical values are (Eqs. 3.16, 3.17 with respectively $x_0 = 0,\ x'_0 = T_0$ and $y_0 = 0,\ y'_0 = P_0$) $\hat{Y} = T_0 \frac{R_E}{\sqrt{1+k}}$ and $\hat{Z} = P_0 \frac{R_E}{\sqrt{-k}}$. wherein R_E denotes the closed orbit radius at energy E (for the record: $R_E \equiv R_0$ at energy $E = 200$ keV, in the foregoing).

Table 3.25 Maximum amplitude of the oscillation, from raytracing (columns denoted "ray-tr.") and from theory. R_E is the closed orbit radius for the energy of concern, $T_0 = P_0 = 0.1$ mrad is the trajectory angle at the origin, positions at the origin are zero

		$\hat{Y}$		$\hat{Z}$	
E (MeV)	k	ray-tr.	$T_0 \frac{R_E}{\sqrt{1+k}}$	ray-tr.	$P_0 \frac{R_E}{\sqrt{-k}}$
(MeV)		($\times 10^{-5}$)		($\times 10^{-5}$)	
0.2	-0.03	1.3123	1.3125	7.4622	7.4624
1	-0.072787	3.1270	3.1267	1.1160	1.1160
5	-0.20586	8.5010	8.5008	1.6697	1.6697

(c) Comparison with theory.

Figure 3.42 shows the difference between numerical and theoretical vertical motion excursion, using an *ad hoc* gnuplot script. An integration step size $\Delta s = 2$ cm is used in the numerical integration.

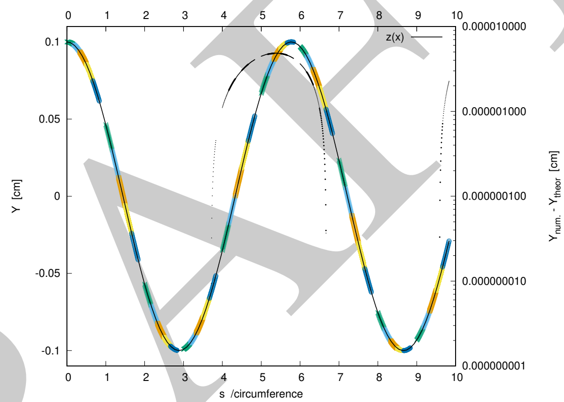

Fig. 3.42 Vertical excursion of a 1 MeV trajectory over 20 turns (left vertical axis), and difference with theoretical expectation as per Eq. 3.17 (right vertical axis). The plot shows two sinusoidal curves: a segmented one, thicker, from numerical integration, and a thinner one, superimposed, from Eq. 3.17

(d) A scan of energy dependence of wave numbers.

A scan of the wave numbers over 200 keV−5 MeV energy range, computing tunes with MATRIX, is performed using the input data file given in Tab. 3.26 (essentially a copy of the input data file of Tab. 3.23, with an INCLUDE accounting for 6 DIPOLEs [16, *lookup* INDEX]).

OBJET[KOBJ=5] generates 13 particles with paraxial horizontal, vertical and longitudinal sampling, proper to allow the computation of the first order transport coefficients and wave numbers by MATRIX. REBELOTE repeats MATRIX computation for a series of different particle rigidities. It is preceded by FIT which finds the closed orbit. MATRIX includes a PRINT command, which causes the transport coefficients (and various other outcomes of MATRIX computation) to be logged in zgoubi.MATRIX.out. This allows producing the graphic in Fig. 3.43 - using the gnuplot script given at the bottom of Tab. 3.26.

Table 3.26 Simulation input data file: for this wave number scan, the INCLUDE segment is taken from Tab. 3.21

```
Field and derivative dB/dR, as a finction of R
'MARKER'   ProbMATRIX_S
'OBJET'
64.62444403717985                      ! Reference Brho ("BORO" in the users' guide) -> 200keV proton.
5                                                  ! Define 13 particles for MATRIX computation.
.001 .01 .001 .01 .001 .00001                                ! Sampling of the initial coordinates.
12.924888 0. 0. 0. 0. 1.              ! Reference: p[MeV/c]=193.739, Brho[kG.cm]=BORO, kin-E[MeV]=0.2.
'INCLUDE'
1               ! IL=2 is necessary under DIPOLE, for step-by-step log of particle data in zgoubi.plt.
6* 60DegSectorR200.inc[#S_60DegSectorR200:#E_60DegSectorR200]                 ! Six 60 degree sectors.
'FIT'
1
2 30 0 [12,80]                      ! Vary particle's Y0 at OBJET, to have it match its D (=Brho/BORO).
1  1e-10
3.1 1 2 #End 0. 1. 0                                                          ! Consrain Y_final=Y0.
'MATRIX'
1 11  PRINT           ! PRINT: log computation outcome data to zgoubi.MATRIX.out, for further plotting.
'REBELOTE'
25 0.1 0 1                                         ! Scan parameter 35 (particle 1's D) in OBJT.
1
OBJET 35 1:5.00639
'SYSTEM'
1
gnuplot < ./gnuplot_MATRIX_Qxy.gnu
'MARKER'   ProbMATRIX_E
'END'
```

A gnuplot script to obtain Fig. 3.43:

```
# gnuplot_MATRIX_Qxy.gnu
set xlab "kin. E [MeV]";set ylab "{/Symbol n}_x,   ({/Symbol n}_x^2+{/Symbol n}_y^2)^{1/2}";set y2label "{/Symbol n}_y"
set key t l maxrow 1; set xtics; set ytics nomirror; set y2tics nomirror
BORO = 64.62444403717985; am = 938.27203e6; c = 2.99792458e8; BrhoRef = BORO *1e-3; eV2MeV = 1e-6
plot  "zgoubi.MATRIX.out" u ((sqrt(($47*BrhoRef*c)**2 + am*am)-am)*eV2MeV):($56) w lp pt 5 lt 1 lw .5 lc rgb "red" \
  tit "{/Symbol n}_x " , \
  "zgoubi.MATRIX.out" u ((sqrt(($47*BrhoRef*c)**2 + am*am)-am)*eV2MeV):($57) axes x1y2 w lp \
  pt 6 lt 3 lw .5 lc rgb "blue" tit "{/Symbol n}_y " ,\
  "zgoubi.MATRIX.out" u ((sqrt(($47*BrhoRef*c)**2 + am*am)-am)*eV2MeV):(sqrt($56**2+$57**2)) \
  w lp pt 7 lt 1 lw .5 lc rgb "black" t "    ({/Symbol n}_x^2+{/Symbol n}_y^2)^{1/2}";pause 1
```

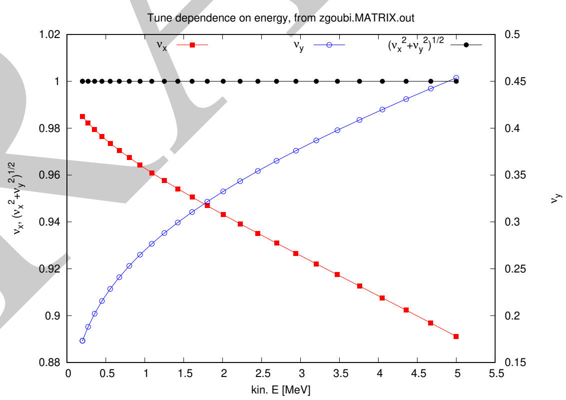

Fig. 3.43 A scan of the energy dependence of the horizontal and vertical wave numbers. Markers are from raytracing, solid lines are from theory (Eq. 3.18). The figure also shows that the raytracing yields $\nu_R^2 + \nu_y^2 = 1,\ \forall E$, as expected