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).

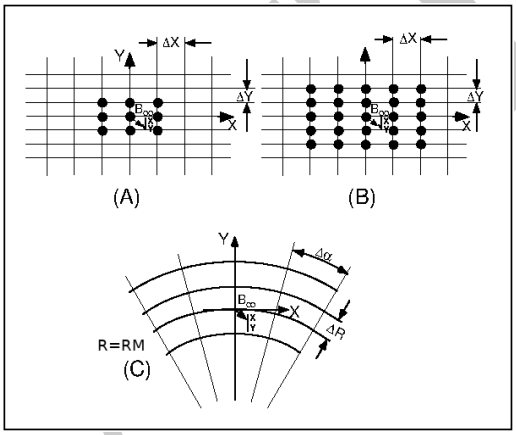

Fig. 3.19 Principle 2-D field map mesh as used by TOSCA, and the (O;X,Y) coordinate system. (A), (B): Cartesian mesh in the (X,Y) plane, case of respectively 9-point and a 25-point interpolation grid; the mesh increments are ΔX and ΔY. (C) : polar mesh and increments ΔR and $\Delta\alpha$ ($\Delta\theta$ 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
```

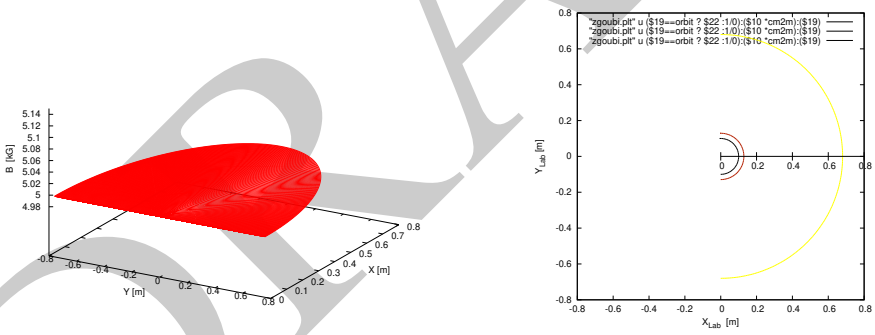

Fig. 3.20 Left: map of a constant magnetic field over a 180 deg sector, 76 cm radial extent. Right: three circular trajectories, at respectively 0.12, 0.2 and 5.52 MeV, computed using that field map

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 $M = 938.272\,\text{MeV/c}^2$ 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.$\dot{8}$.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).

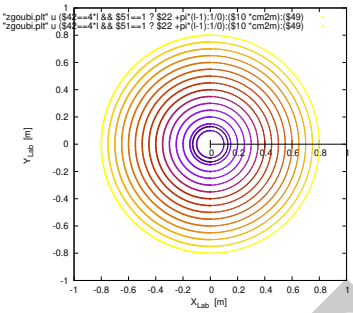

Fig. 3.21 Circular trajectories in the cyclotron mid-plane, centered on the field map center. The outermost orbit is at R=80 cm by hypothesis, thus $BR = B_0 \times R = 0.4\,\mathrm{T\,m}$, $E_k = 7.632\,\mathrm{MeV}$. These stepwise (R, θ) data are read from zgoubi.plt, coordinates (Y,X) in `zgoubi` polar frame nomenclature [16, Sect.8̇.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.

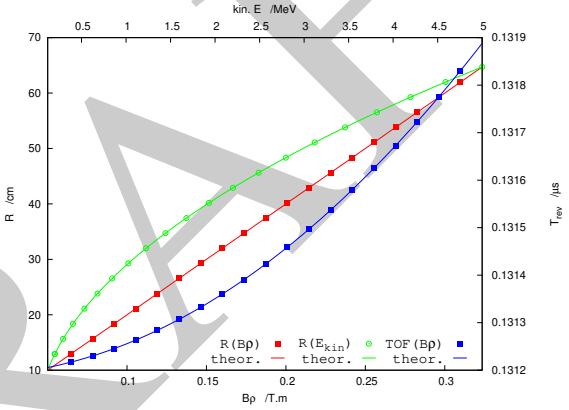

Fig. 3.22 Numerical (markers) and theoretical (solid lines) values of orbit radius, R, and revolution period, T_{rev}, versus kinetic energy (top scale) and rigidity (bottom scale). The mesh density here is $N_\theta \times N_R = 315 \times 151$. The integration step size is $\Delta s = 1\,\mathrm{cm}$,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

The reason why it is possible to push the raytracing beyond the 76 cm radius field map extent, without loss of accuracy, is that the field is constant. Thus, referring to the polynomial interpolation technique used [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.

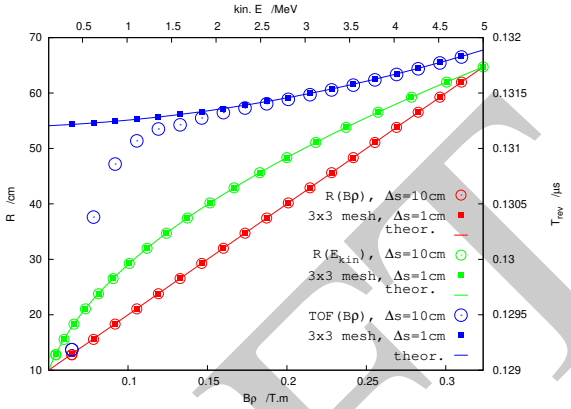

Fig. 3.23 Convergence versus mesh density and step size: a graph of orbit radius R (left axis), and revolution period, $T_{\rm rev}$ (right axis), versus kinetic energy (top scale) and rigidity (bottom scale). Solid markers are for $\Delta s = 1$ cm and $N_\theta \times N_R = 3 \times 3$ node mesh, large 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 $E_k : 0.12 \rightarrow 5$ 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

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

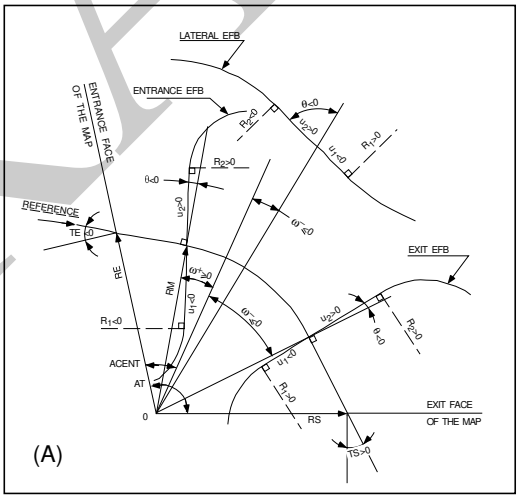

Fig. 3.24 Parameters used to define the geometry of a dipole magnet with index, using DIPOLE. In the text, ACENT is noted ACN [16, Fig. 9]

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,

- 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.

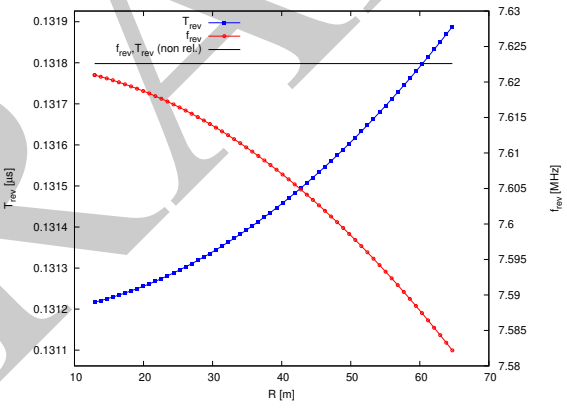

Fig. 3.25 A scan of radius-dependent revolution frequency. An analytical model of a cyclotron dipole is used, featuring uniform field (no radial gradient, at this point)

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.

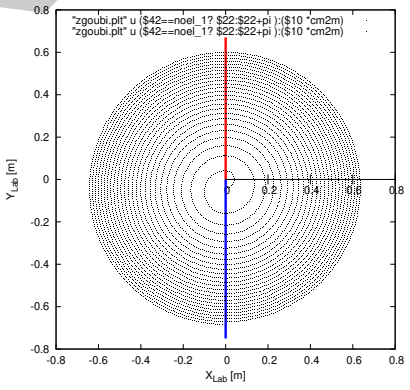

Fig. 3.26 Twenty five turn spiral trajectory of a proton accelerated in a uniform 0.5 T field from 20 keV to 5 MeV at a rate of 200 kV per turn (a 100 kV gap voltage). The vertical thick line materializes the gap, the upper half (red) corresponds to the first occurrence of CAVITE in the sequence (Tab.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

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  # lst 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"
```

(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 $E = E_k + M$.

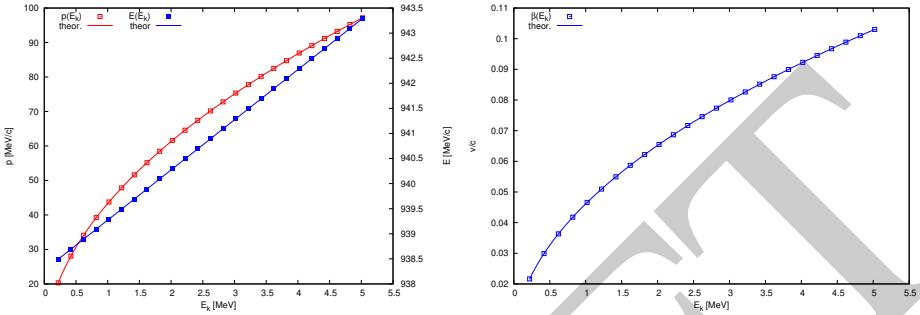

Fig. 3.27 Energy dependence of, left: proton momentum p (left axis) and total energy E (right axis) and of, right: proton normalized velocity $\beta = v/c$. Markers: from raytracing; solid lines: theoretical expectation

(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, $\beta(E_k) = p/(E_k + M)$.

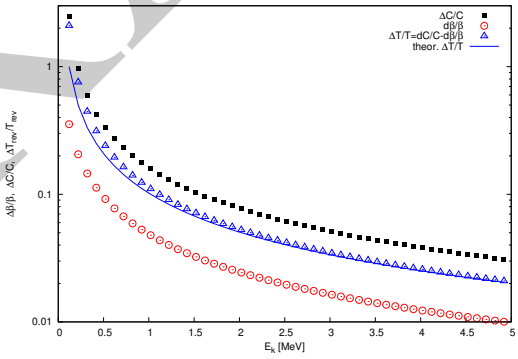

Fig. 3.28 Relative variation of velocity $\Delta\beta/\beta$ (empty circles), circumference $\Delta C/C$ (solid disks) and revolution time $\Delta T/T$ (triangles), as a function of energy, from raytracing. Theoretical expectation for the latter is also displayed (solid line), for comparison

(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.

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'
```

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