EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN – AB DIVISION

Werner.Herr@cern.ch

CAS 2003

Problems and solutions of the exercises in the optics course at the CAS 2003 in Zeuthen.

O. Brüning and W. Herr, AB Division, CERN, 1211-Geneva 23

Abstract

Geneva, Switzerland

15th September 2003

1 Introduction

At the CERN Accelerator School 2003 at DESY, Zeuthen, a course was held on optics design. This course was intended as an introduction for accelerator physicists who want to start the design of accelerator optics as well as for people from other fields who would like to get a basic knowledge of the principles of accelerator design.

The aim was that the participants design a realistic accelerator optics, following plenary lectures on lattice cells [2], insertions [3] and imperfections [4, 5]. This should be done following a series of steps in form of exercises the participants had to solve and implement in an accelerator design program (MAD-X).

After an introduction to MAD-X [6, 7], the participants had to design a periodic machine we well defined properties. In the second part, correction schemes for chromaticity, closed orbit etc. had to be defined and implemented. In the third part, a straight section with a dispersion suppressor and a low β-insertion was inserted into the originally periodic structure. The tunability of this lattice and further improvements to increase the flexibility were studied and implemented where required.

In this paper we discuss the proposed solutions and show some of the techniques used in the design of accelerator lattices. Some references in this note correspond to example files and the solutions which will be made available on the web [8] and on a CD-ROM [9].

2 Exercise 1

2.1 Problem:

Design a machine for protons at a momentum of 20 GeV/c with the following basic parameters:

- Circumference = 1000 m,
- Quadrupole length L_{quad} = 3.0 m,
- 8 FODO cells.
- Dipole length is 5 m, maximum field is 3 T

Apply the knowledge from previous lectures at this school and define a lattice cell according to the boundary conditions (position of dipole magnets and quadrupoles) and find the optics (strength of dipoles and quadrupoles) so that $\beta_{max} \equiv \hat{\beta}$ is around 300m. Implement it into MAD format using thin lenses for all elements and verify the calculations.

2.2 Solution:

Set up a FODO cell of length L like in Fig.1 [2]. The cell length is L=1000 m/8 = 125 m. For simplicity the elements can be distributed equally spaced inside the cell.

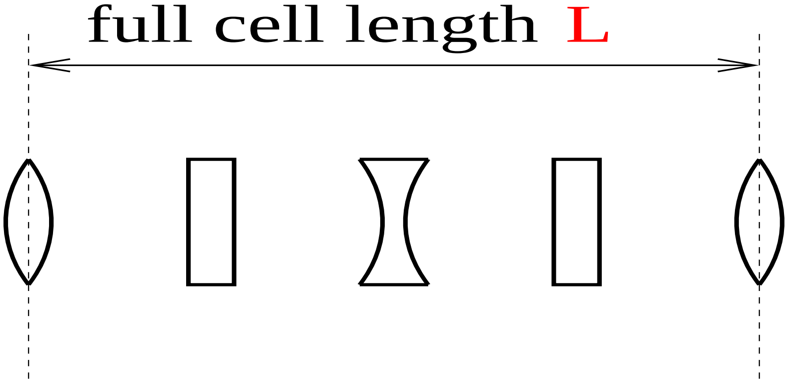

Figure 1: Schematic view of a symmetric FODO cell.

The maximum integrated field of a dipole is 15 Tm. At 20 GeV/c this corresponds to a deflection angle of 0.225. To get a total angle of 2π, we need 2π/0.225 = 27.9 → 28 dipoles. For 8 cells, we therefore need 32 dipoles. We need 8 focusing and 8 defocusing quadrupoles and 32 bending dipoles, i.e. 4 dipoles per cell. The bending angle of the dipoles must be $2\pi/32$ = 0.196349541.

For a FODO cell we get the expression for the phase advance $\phi/2$ per half cell:

$$sin(\phi/2) = \frac{L_{1/2}}{f_{1/2}} \tag{1}$$

where $L_{1/2}$ is the length of the half cell (62.5 m) and $f_{1/2}$ the focal length of the half cell. the expressions for the maximum and minimum betatron function β are written:

$$\hat{\beta} = f_{1/2} \cdot \frac{1 + sin(\phi/2)}{cos(\phi/2)} = L_{1/2} \cdot \frac{1 + sin(\phi/2)}{sin(\phi/2)cos(\phi/2)} \tag{2}$$

and

$$\check{\beta} = f_{1/2} \cdot \frac{1 - sin(\phi/2)}{cos(\phi/2)} = L_{1/2} \cdot \frac{1 - sin(\phi/2)}{sin(\phi/2)cos(\phi/2)} \tag{3}$$

For $\hat{\beta} = 300\,m$ and $L_{1/2} = 62.5\,m$ we get for $\phi/2 = 16.07066^o$. Therefore:

$$f_{1/2} = 62.5/sin(\phi/2) \quad = 225.776\,m \quad \rightarrow \quad f = 112.888\,m \tag{4}$$

this gives for the focusing strength k_1 (positive for focusing and negative for defocusing quadrupole):

$$k_1 = \frac{1}{L_{quad} \cdot f} = \pm\ 0.002952775 \tag{5}$$

This cell can be implemented in MAD as (EX1/ex1_thick.seq):

```
circum=1000.0;
ncell = 8;
lcell = circum/ncell;
lquad = 3.00;
lquad2 = lquad/2.;

// forces and other constants;
// element definitions;

// define bending magnet as multipole
mb: multipole, lrad=dummy, l=10.0,knl={2.0*pi/(4*ncell)};

mq: quadrupole, l=lquad;
qf: mq, k1:=kqf;
qd: mq, k1:=kqd;
kqf =  .295278e-2;
kqd = -.295278e-2;

// sequence declaration;
cascell2: sequence, refer=centre, l=circum;
start_machine: marker, at = 0;
!
   n = 1;
   while (n < ncell+1) {
   qf: qf,   at=(n-1)*lcell + lquad2;
   mb: mb,   at=(n-1)*lcell+0.15*lcell + lquad2;
   mb: mb,   at=(n-1)*lcell+0.35*lcell + lquad2;
   qd: qd,   at=(n-1)*lcell+0.50*lcell + lquad2;
   mb: mb,   at=(n-1)*lcell+0.65*lcell + lquad2;
   mb: mb,   at=(n-1)*lcell+0.85*lcell + lquad2;
!
   n = n + 1;
}
end_machine: marker at=circum;
endsequence;
```

In this cell, the cell starts at the entry of the focusing quadrupole and ends at the entry of the focusing quadrupole of the next cell. It is repeated $ncell$ times to make the ring of 1000 m circumference.

The bending magnets are defined as multipoles and their first component is the bending angle. An alternative, probably simpler implementation would be using only thin elements like in (EX1/ex1_thin.seq):

```
circum=1000.0;
ncell =  8; // Number of cells
lcell = circum/ncell;
lq = 3.00; // Length of a quadrupole

// element definitions;

// define bending magnet as multipole
// we have 4 bending magnets per cell
mb: multipole,lrad=d,knl={2.0*pi/(4*ncell)};

// define quadrupoles as multipoles
qf: multipole,lrad=d,knl={0,0.295278e-2*lq};
qd: multipole,lrad=d,knl={0,-0.295278e-2*lq};

// sequence declaration;
cascell1: sequence, refer=centre, l=circum;
start_machine: marker, at = 0;
!
   n = 1;
   while (n < ncell+1) {
   qf: qf,   at=(n-1)*lcell;
   mb: mb,   at=(n-1)*lcell+0.15*lcell;
   mb: mb,   at=(n-1)*lcell+0.35*lcell;
   qd: qd,   at=(n-1)*lcell+0.50*lcell;
   mb: mb,   at=(n-1)*lcell+0.65*lcell;
   mb: mb,   at=(n-1)*lcell+0.85*lcell;
!
   n = n + 1;
}
end_machine: marker at=circum;
endsequence;
```

Now the cells start and end in the *centres* of the focusing quadrupoles. Both declaration can be executed with the following MAD commands (EX1/ex1_thin.mad):

```
TITLE, 'CAS2003 first exercise';
call file="ex1_thin.seq";

Beam, particle = proton, sequence=cascell1, energy = 450.0,
          NPART=1.05E11, sige=      4.5e-4 ;

use, sequence=cascell1;
select,flag=twiss,column=name,s,x,y,mux,betx,
                         muy,bety,dx,dy;
twiss,save,centre,file=twiss.out;
plot, haxis=s, vaxis=betx, bety, colour=100;

stop;
```

The results are printed to a file *twiss.out* and the graphical output is found in *madx.ps*. Below one finds the Twiss summary printed at the end of the Twiss calculations. The alternative definition of the quadrupoles was used for this run and will be used in all later examples. MAD gives a maximum β ($\hat{\beta}_x$ and $\hat{\beta}_y$) in both planes of 300 m, in good agreement with the desired parameters.

```
++++++ table: summm

          length            orbit5              alfa           gammatr
 1.000000000e+03  -0.000000000e+00   1.989271491e+00   7.090109959e-01

              q1               dq1           betxmax             dxmax
 7.142528897e-01  -7.335940332e-01   2.999996548e+02   3.646157387e+02

           dxrms            xcomax            xcorms                q2
 3.211327329e+02   0.000000000e+00   0.000000000e+00   7.142528897e-01

             dq2           betymax             dymax             dyrms
-7.335940332e-01   2.999996548e+02   0.000000000e+00   0.000000000e+00

          ycomax            ycorms            deltap
 0.000000000e+00   0.000000000e+00   0.000000000e+00
```

3 Exercise 2

3.1 Problem:

Start with the lattice from exercise 1 and modify it so the maximum betatron function $\hat{\beta}$ is around 100 m. The circumference must not be changed.

3.2 Solution:

For the maximum of the β-function we had:

$$\hat{\beta} \ = \ f_{1/2} \ \cdot \ \frac{1 + sin(\phi/2)}{cos(\phi/2)} \ = \ L_{1/2} \ \cdot \ \frac{1 + sin(\phi/2)}{sin(\phi/2)cos(\phi/2)} \tag{6}$$

For a given half cell length $L_{1/2}$ the smallest $\hat{\beta}$ can be found be differentiating the equation and one obtains:

$$sin^3(\phi/2) + 2sin^2(\phi/2) - 1 = 0 \tag{7}$$

The solution is $\phi/2 \ = \ 38.17^o$ and therefore the minimum $\hat{\beta}$ is 3.3302 $\cdot$ $L_{1/2}$. For the present cell length this gives: $\hat{\beta}_{min}$ = 206.375 m, i.e. we cannot reach this value with the present lattice.

The only way to decrease $\hat{\beta}_{min}$ is to reduce the cell length, i.e. increase the number of cells for a given circumference.

We increase the number of cells to 20 and get $L_{1/2}$ = 25. With the same procedure as before we get for the phase advance per half cell: $\phi/2 \ = \ 21.56^o$. With this phase advance we need f$_{1/2}$ = 68.0 m and f = 34.0 m. This gives focusing strengths of k$_1$ $=$ $\pm$ 0.0098.

After modifying the previous sequence with ncell=20 and the quadrupole strength, we get the desired $\hat{\beta}_{min}$ = 100 m.

Using the previous MAD input and with the necessary modifications (ncell = 20, k$_1$ $=$ $\pm$ 0.0098, EX2/ex2_thin.seq and EX2/ex2_thin.mad) we get the output below.

```
++++++ table: summ

           length             orbit5               alfa            gammatr
  1.000000000e+03   -0.000000000e+00    1.789932275e-01    2.363642010e+00

               q1                dq1            betxmax              dxmax
  2.395724243e+00   -2.515616751e+00    1.000267291e+02    3.441962504e+01

            dxrms             xcomax             xcorms                 q2
  2.900240314e+01    0.000000000e+00    0.000000000e+00    2.395724243e+00

              dq2            betymax              dymax              dyrms
 -2.515616751e+00    1.000267291e+02    0.000000000e+00    0.000000000e+00

           ycomax             ycorms             deltap
  0.000000000e+00    0.000000000e+00    0.000000000e+00
```

4 Exercise 3

4.1 Problem:

Start with the lattice from exercise 2 and modify it so you can correct the chromaticity for both planes to zero. Try first to calculate approximately the required strengths. Implement your correction scheme in your previous MAD files and verify your calculation. Use MAD to compute the exact strengths required by matching the global parameters Q'_x and Q'_y (in MAD names: DQ1 and DQ2). Compare the results with your calculations.

4.2 Solution:

From exercise 2 we know that the horizontal and vertical tunes are $\phi \cdot ncell$ = 21.56$^o \cdot 2 \cdot ncell$ = 2.396. The natural chromaticity is [2]:

$$\xi_0 \;=\; -\frac{1}{\pi} tan(\phi/2) \cdot ncell \;=\; -2.515$$

The precise numbers we can get from the MAD summary table.

For the correction we need an element where the $focusing\ length\ f$ is amplitude dependent, e.g. a sextupole where the focusing length (i.e. the derivative of the force) depends linearly on the amplitude. Furthermore, we have to $sort$ the particles in amplitude according to their momentum. This can be done with horizontal dispersion. Therefore we put sextupoles in regions $with$ horizontal dispersion. (Remark: sextupoles in regions $without$ dispersion can be used to deliberately control third order resonances without affecting the chromaticity).

In order to correct both, the horizontal and vertical chromaticities to zero, we need two independent groups of sextupoles. A natural choice is next to the quadrupoles. We shall call the two groups SF and SD, when they are next to the focusing or defocusing quadrupoles. The tune change due to a sextupole is easily written:

$$\Delta Q \;=\; -\frac{1}{4\pi}\int \beta k_2 \cdot x\; ds \;=\; -\frac{1}{4\pi}\int \beta k_2 \cdot D_x \frac{\delta p}{p}\; ds \;=\; -\frac{1}{4\pi}\int \beta k_2 D_x ds \cdot \frac{\delta p}{p} \;=\; Q' \;\cdot\; \frac{\delta p}{p}$$

For thin lenses the chromaticity change for two groups of sextupoles is in the horizontal plane:

$$\Delta Q'_x \;=\; -\frac{1}{4\pi}\sum_{F\ sextupole} k_2^F l_s D_x^F \beta_x^F + \frac{1}{4\pi}\sum_{D\ sextupole} k_2^D l_s D_x^D \beta_x^D$$

or for a periodic structure

$$\Delta Q'_x \;=\; -\frac{1}{4\pi}\cdot ncell \cdot (k_2^F l_s D_x^F \beta_x^F - k_2^D l_s D_x^D \beta_x^D)$$

For $\Delta Q'_y$ the horizontal and vertical β-functions have to be exchanged in the formulae.

$$\Delta Q'_y \;=\; -\frac{1}{4\pi}\cdot ncell \cdot (-k_2^F l_s D_x^F \beta_y^F + k_2^D l_s D_x^D \beta_y^D)$$

These two equations form a system of equation with the two unknowns k_2^F and k_2^D when the desired $\Delta Q'$ changes are given.

From the MAD output for the previous exercise we know the Twiss parameters at the sextupoles (Tab.1). It is clear that the dispersion function at the position of the sextupoles should be large to minimize the required strengths. Furthermore, at the position of the two groups, the horizontal and vertical β-functions should be as different as possible. Therefore our choice to place the sextupoles next to the quadrupoles was a good choice.

	D_x	β_x	β_y
QF	34.427	100.027	46.265
QD	23.739	46.265	100.027

Table 1: Twiss parameters at lattice sextupoles.

Since we want to correct the chromaticities to zero, we take:

$$\Delta Q'_x \;=\; -\,Q'_x \quad and \quad \Delta Q'_y \;=\; -\,Q'_y$$

Then we solve the above system of equations with the numbers from the table.
We find: $k_2^F l_s \;=\; 0.017041/ncell$ and $k_2^D l_s \;=\; -\,0.024714/ncell$.

To our MAD description from the previous exercise we add the following lines (EX3/ex3.seq):

```
// define the sextupoles as multipole
lsex = 0.00001; // dummy length, only used in the sequence;
ksf :=  +0.017041/20.0;
ksd :=  -0.024714/20.0;

// ATTENTION: must use knl:=  and NOT knl= to match later !
msf: multipole, lrad=dummy, knl:={0,0,ksf};
msd: multipole, lrad=dummy, knl:={0,0,ksd};
```

add change the definition of our cell:

```
   n = 1;
   while (n < ncell+1) {
   qf: qf,     at=(n-1)*lcell;
   msf: msf,   at=(n-1)*lcell + lsex/2.0;
   mb: mb,     at=(n-1)*lcell+0.15*lcell;
   mb: mb,     at=(n-1)*lcell+0.35*lcell;
   qd: qd,     at=(n-1)*lcell+0.50*lcell;
   msd: msd,   at=(n-1)*lcell+0.50*lcell + lsex/2.0;
   mb: mb,     at=(n-1)*lcell+0.65*lcell;
   mb: mb,     at=(n-1)*lcell+0.85*lcell;
!
   n = n + 1;
}
```

This gives the following global parameters:

```
++++++ table: summ
```

```
           length           orbit5             alfa          gammatr
   1.00000000e+03  -0.00000000e+00   1.78993227e-01   2.36364201e+00

               q1              dq1          betxmax            dxmax
   2.39572424e+00  -6.30299369e-03   1.00026729e+02   3.44270344e+01

            dxrms           xcomax           xcorms               q2
   2.91482386e+01   0.00000000e+00   0.00000000e+00   2.39572424e+00

              dq2          betymax            dymax            dyrms
  -6.11732076e-03   1.00026729e+02   0.00000000e+00   0.00000000e+00

           ycomax           ycorms           deltap
   0.00000000e+00   0.00000000e+00   0.00000000e+00
```

The chromaticities are corrected to values smaller than 0.01 for both planes. We get for Q'_x (dq1) = -6.3 10^{-3} and Q'_y (dq2) = -6.1 10^{-3}.

To get the exact values for the sextupole strengths, MAD can be used to MATCH these values. An example is given below (EX3/ex3.mad):

```
TITLE, s='MAD-X test';
call file="ex3.seq";
option,-echo;
// option,debug,-echo;

Beam, particle = proton, sequence=cascell3, energy = 45.0,
          NPART=1.05E11, sige=      4.5e-4 ;

use, period=cascell3;

match, sequence=cascell3;
   vary,name=ksf, step=0.00001;
   vary,name=ksd, step=0.00001;
   global,sequence=cascell3,DQ1=0.0;
   global,sequence=cascell3,DQ2=0.0;
   Lmdif, calls=10, tolerance=1.0e-21;
endmatch;

select,flag=twiss,column=name,s,betx,muy,bety,dx,dy;
twiss,save,centre,deltap=0.0,file=twiss2.out;
plot, haxis=s, vaxis=x, y;
stop;
```

The output produced (Twiss summary) is:

```
++++++ table: summ

          length           orbit5             alfa          gammatr
  1.00000000e+03  -0.00000000e+00   1.78993227e-01   2.36364201e+00

              q1              dq1          betxmax            dxmax
  2.39572424e+00   1.07207358e-14   1.00026729e+02   3.44270344e+01

           dxrms           xcomax           xcorms               q2
  2.91482386e+01   0.00000000e+00   0.00000000e+00   2.39572424e+00

             dq2          betymax            dymax            dyrms
  2.41967395e-15   1.00026729e+02   0.00000000e+00   0.00000000e+00

          ycomax           ycorms           deltap
  0.00000000e+00   0.00000000e+00   0.00000000e+00
```

It can be seen that the chromaticity is now practically zero. The values found from the matching procedure were:
ksf = 8.54169767E-04 (= 0.017083395/20) and
ksd = -1.23874055E-03 (= -0.024774811/20)
We find that our original values are already very close !

5 Exercise 4

5.1 Problem:

Start with the lattice from the previous exercise and assume random misalignments of the quadrupoles of r.m.s. 0.1 mm in the horizontal and 0.2 mm in the vertical plane. Calculate the expected r.m.s. orbit and verify with MAD.

5.2 Solution:

Assuming the thin lens approximation, one can compute the r.m.s. orbit distortion. The deflection $\delta x'_i$ due to a (here horizontal) displacement Δx_i of a quadrupole is:

$$\delta x'_i = k_1 \cdot l_{quad} \cdot \Delta x$$

The orbit response x_j at a position j due to a single kick δx_i at position i is [4]:

$$x_j \; = \; \frac{\sqrt{\beta_i \beta_j}}{2 \; sin(\pi Q_x)} cos(\pi Q_x - (\mu_i - \mu_j)) \cdot \delta x'_i$$

Here Q is the horizontal tune and μ the phase function around the machine. Taking the average of the squared displacements at all positions, and using an average $\beta \; = \; \bar{\beta}$:

$$< x^2 > \;\; = (\; \frac{\sqrt{\bar{\beta}}}{2 \; sin(\pi Q_x)})^2 \;\; \cdot \;\; \frac{1}{2} \sum \bar{\beta} \delta x'^2_i$$

where we got a factor $\frac{1}{2}$ from averaging $\cos^2(\mu)$ and we write for x_{rms}:

$$\sqrt{< x^2 >} \; = \; x_{rms} \; = \; \frac{\sqrt{\bar{\beta}}}{2\sqrt{2} \; sin(\pi Q_x)} \sqrt{\sum \bar{\beta} \delta x'^2_i}$$

Assuming N identical cells with $N_d = 2N$ quadrupoles we can re-write the sum:

$$x_{rms} \; = \; \frac{\sqrt{\bar{\beta}}}{2\sqrt{2} \; sin(\pi Q_x)} \;\; \cdot \;\; \sqrt{N_d} \; \sqrt{\bar{\beta}} \delta x'_{rms}$$

$$x_{rms} \; = \; \frac{\sqrt{N_d} \; \bar{\beta}}{2\sqrt{2} \; sin(\pi Q_x)} \;\; \cdot \;\; \delta x'_{rms}$$

We use the expressions for thin lenses [2]:

$$\bar{\beta} = \frac{L_{cell}}{sin(\phi)}$$

and:

$$k_1 l_{quad} = \frac{4 sin(\phi/2)}{L_{cell}}$$

and obtain after some mathematics:

$$x_{rms} \; = \; \frac{\sqrt{N_d}}{\sqrt{2} \; sin(\pi Q_x) cos(\phi/2)} \cdot \Delta x'_{rms}$$

For a r.m.s. quadrupole displacement of 0.1 mm we can use the above expression and get as an estimate for $x_{rms} \; \approx \; 0.5$ mm and $y_{rms} \; \approx \; 1.0$ mm.

This can be verified with MAD by adding the lines:

```
eoption,add=false,seed=22021955;
select,flag=error,pattern="q.*";
ealign dx:=tgauss(3.0)*0.1e-3,dy:=tgauss(3.0)*0.2e-3;
```

This will assign random errors in the quadrupoles (selected all elements starting with ’q’) following Gaussian distributions with r.m.s. values of 0.1 mm in the horizontal and 0.2 mm in the vertical plane. The SEED option allows to specify the starting point for the random number generator.

The MAD summary is given below and shows good agreement with the expectation. The x$_{rms}$ (xcorms) is 0.48 mm and y$_{rms}$ is 1.14 mm.

```
++++++ table: summm

           length            orbit5              alfa           gammatr
   1.00000000e+03   -0.00000000e+00    1.78992963e-01    2.36364375e+00

               q1               dq1           betxmax             dxmax
   2.39572424e+00   -2.51838414e+00    1.00026729e+02    3.44619881e+01

            dxrms            xcomax            xcorms                q2
   2.92578856e+01    1.23559802e-03    4.80265442e-04    2.39572424e+00

              dq2           betymax             dymax             dyrms
  -2.51838414e+00    1.00026729e+02    1.25978040e-02    5.91226626e-03

           ycomax            ycorms            deltap
   2.69993776e-03    1.13922436e-03    0.00000000e+00
```

6 Exercise 5

6.1 Problem:

Start with the lattice from the previous exercise and add the necessary equipment to be able to correct the closed orbit in both planes, (using MAD). Estimate first the maximum necessary strength of the orbit correctors assuming a maximum quadrupole displacement of 1 mm.

6.2 Solution:

The estimate of the necessary corrector strength is straightforward. The angular deflection of the beam passing through a quadrupole at an offset Δx is:

$$\delta x' \;=\; k_1 \cdot l \cdot \Delta x$$

Assuming the correctors next to the quadrupole and a maximum offset of 1 mm, we get about 30 μrad. To allow for some additional errors of monitors (scaling and positioning) or field errors from dipoles and a reserve we define the required strength as 150 μrad, which is a rather safe value. For the energy of 20 GeV of our beam this requires corrector magnets with an integrated field of $\approx$ 0.010 Tm, which can easily be provided (LHC orbit correctors provide 1.89 Tm).

To correct the orbit (with MAD or other orbit correction programs), we have to add orbit monitors and orbit correctors to the lattice. We decide to add one monitor and one corrector at each quadrupole. Monitors and orbit correctors in MAD can be defined to act only in one or in both planes (see MAD-X manual). While we want our monitors to measure the orbit in both planes, we define horizontal correctors at the focusing and vertical orbit correctors at defocusing quadrupoles. This needs an extra (reserve) contribution to the corrector strength since not all imperfections can be corrected locally. We suggest the following sequence as one possibility (EX5/ex5.seq):

```
circum=1000.0;
ncell = 20;
lcell = circum/ncell;
lquad = 3.00;
lquad2 = lquad/2.;
lsex = .01;
lbpm  = 0.001;
lbpm2 = lbpm/2.;
lcor  = 0.001;
lcor2 = lcor/2.;

// forces and other constants;
// element definitions;

// define orbit monitors and correctors
bpm: monitor, l = lbpm;
ch : hkicker, l = lcor;
cv : vkicker, l = lcor;

// define bending magnet as multipole
mb: multipole, lrad=dummy, knl={2.0*pi/(4*ncell)};

// define the quadrupoles as multipole
kqf =  .980000e-2;
```

```
kqd = -.980000e-2;
qf: multipole, lrad=dummy, knl={0,lquad*kqf};
qd: multipole, lrad=dummy, knl={0,lquad*kqd};

// define the sextupoles as multipole
!ksf :=  +0.017041/20.0;
!ksd :=  -0.024714/20.0;
ksf :=  +0.0;
ksd :=  -0.0;

// ATTENTION: must use knl:=  not knl= to match !
msf: multipole, lrad=dummy, knl:={0,0,ksf};
msd: multipole, lrad=dummy, knl:={0,0,ksd};
!msf: multipole, lrad=dummy, knl:={0,0,ksf};
!msd: multipole, lrad=dummy, knl:={0,0,ksd};

// sequence declaration;
cascell4: sequence, refer=centre, l=circum;
start_machine: marker, at = 0;
!
   n = 1;
   while (n < ncell+1) {
   qf: qf,     at=(n-1)*lcell;
   bpm: bpm,   at=(n-1)*lcell + lbpm2;
   ch : ch,    at=(n-1)*lcell + lbpm + lcor2;
   msf: msf,   at=(n-1)*lcell + lsex/2.0;
   mb: mb,     at=(n-1)*lcell+0.15*lcell;
   mb: mb,     at=(n-1)*lcell+0.35*lcell;
   qd: qd,     at=(n-1)*lcell+0.50*lcell;
   bpm: bpm,   at=(n-1)*lcell+0.50*lcell + lbpm2;
   cv:  cv,    at=(n-1)*lcell+0.50*lcell + lbpm + lcor2;
   msd: msd,   at=(n-1)*lcell+0.50*lcell + lsex/2.0;
   mb: mb,     at=(n-1)*lcell+0.65*lcell;
   mb: mb,     at=(n-1)*lcell+0.85*lcell;
!
   n = n + 1;
}
end_machine: marker at=circum;
endsequence;
```

To correct an orbit with MAD, one single command for each plane is required (EX5/ex5.mad):

```
correct,flag=ring,mode=micado,error=1.0e-7,ncorr=20,
        plane=x,clist="c.tab",mlist="m.tab";
correct,flag=ring,mode=micado,error=1.0e-7,ncorr=20,
        plane=y,clist="c.tab",mlist="m.tab";
```

The desired algorithms (MICADO), the plane, the number of correctors (5) and some output files are passed as parameters.

A following execution of a Twiss command will make use of the computed correction and give the corrected orbit.

7 Exercise 6

7.1 Problem:

Start with the lattice from the previous exercise and first double the circumference to 2000 m and the number of cells to 40. Change the phase advance to $\phi = 60^o$ per cell. Insert a straight section: i.e. 2 cells without bending magnets but keep the same focusing. Modify now the lattice to keep the horizontal dispersion function small ($<$ 1-2 m) along this straight section. At this stage do not change the focusing properties in any of the cells. Such straight sections with very small dispersion are very useful for the installation of RF equipment, wigglers, undulators, beam instrumentation, collimation systems etc., or to house an experiment.

7.2 Solution:

We increase the circumference by changing the value in the MAD input to 2000.0 m and the number of cells to 40. We define two sections without dipoles and let them be the cells number 35 and 36. These define our straight section. Before 35 and behind 36 we have to modify the lattice to make a $dispersion\ suppressor$. Two types of suppressors we know: missing magnet dispersion suppressors and half-field dispersion suppressors [3].

To change the phase advance, we insert the following lines into the MAD input (EX6/HF.mad):

```
match, sequence=cascell6;
   vary,name=kqf, step=0.00001;
   vary,name=kqd, step=0.00001;
   global,sequence=cascell6,Q1=6.700;
   global,sequence=cascell6,Q2=6.650;
   Lmdif, calls=10, tolerance=1.0e-21;
endmatch;
```

Our phase advance per cell is now close to 60^o. This allows us to use both types of dispersion suppressors.

We reserve the cells 31 to 34 and 37 to 40 for setting up the dispersion suppressors. We do not change the strengths in the quadrupoles, i.e. we keep the focusing properties of the FODO cell.

7.2.1 Missing magnet dispersion suppressor

From the conditions for a missing magnet dispersion suppressor with 60^o phase advance we find the we need 2 cells on each side of the straight section. The outmost cell without bending magnets and the inner cell with the normal bending strength [3]. In the MAD description this is done easiest by restricting the loop over cells to the regular arc cells and adding the tuning and straight section cells explicitly.

We have to be careful with the definition of the strength of the bending magnets because their number is now not 4*ncell. In the case of the missing magnet case a simple counting is enough, for the reduced field suppressor the fields have to be integrated to give 2π. Note that we have no sextupoles in the diespersion suppressors and the straight section (EX6/casMM.seq).

```
// defines a missing magnet dispersion suppressor
circum=2000.0;
ncell = 40;
lcell = circum/ncell;
lquad = 3.00;
lquad2 = lquad/2.;
lsex = .0001;
```

```
nnorm = 144;

// forces and other constants;
// element definitions;

// define bending magnet as multipole
mb: multipole, lrad=dummy, knl={2.0*pi/nnorm};

// define the quadrupoles as multipole
kqf =  .980000e-2;
kqd = -.980000e-2;
qf: multipole, lrad=dummy, knl:={0,lquad*kqf};
qd: multipole, lrad=dummy, knl:={0,lquad*kqd};

// define the sextupoles as multipole
ksf :=  +0.017041/20.0;
ksd :=  -0.024714/20.0;

// ATTENTION: must use knl:=  not knl= to match !
msf: multipole, lrad=dummy, knl:={0,0,ksf};
msd: multipole, lrad=dummy, knl:={0,0,ksd};

// sequence declaration;
cascell6: sequence, refer=centre, l=circum;
start_machine: marker, at = 0;
!
   n = 1;
   while (n < ncell-9) {
   qf: qf,     at=(n-1)*lcell;
   msf: msf,   at=(n-1)*lcell + lsex/2.0;
   mb: mb,     at=(n-1)*lcell+0.15*lcell;
   mb: mb,     at=(n-1)*lcell+0.35*lcell;
   qd: qd,     at=(n-1)*lcell+0.50*lcell;
   msd: msd,   at=(n-1)*lcell+0.50*lcell + lsex/2.0;
   mb: mb,     at=(n-1)*lcell+0.65*lcell;
   mb: mb,     at=(n-1)*lcell+0.85*lcell;
!
   n = n + 1;
}
   qf: qf,     at=(ncell-10)*lcell;
   mb: mb,     at=(ncell-10)*lcell+0.15*lcell;
   mb: mb,     at=(ncell-10)*lcell+0.35*lcell;
   qd: qd,     at=(ncell-10)*lcell+0.50*lcell;
   mb: mb,     at=(ncell-10)*lcell+0.65*lcell;
   mb: mb,     at=(ncell-10)*lcell+0.85*lcell;

   qf: qf,     at=(ncell-9)*lcell;
   mb: mb,     at=(ncell-9)*lcell+0.15*lcell;
   mb: mb,     at=(ncell-9)*lcell+0.35*lcell;
   qd: qd,     at=(ncell-9)*lcell+0.50*lcell;
   mb: mb,     at=(ncell-9)*lcell+0.65*lcell;
   mb: mb,     at=(ncell-9)*lcell+0.85*lcell;

//   start dispersion suppressor
   qf: qf,     at=(ncell-8)*lcell;
   qd: qd,     at=(ncell-8)*lcell+0.50*lcell;

   qf: qf,     at=(ncell-7)*lcell;
   mb: mb,     at=(ncell-7)*lcell+0.15*lcell;
   mb: mb,     at=(ncell-7)*lcell+0.35*lcell;
   qd: qd,     at=(ncell-7)*lcell+0.50*lcell;
   mb: mb,     at=(ncell-7)*lcell+0.65*lcell;
   mb: mb,     at=(ncell-7)*lcell+0.85*lcell;
//   end dispersion suppressor

//   begin straight section
   qf: qf,     at=(ncell-6)*lcell;
   qd: qd,     at=(ncell-6)*lcell+0.50*lcell;

   qf: qf,     at=(ncell-5)*lcell;
   qd: qd,     at=(ncell-5)*lcell+0.50*lcell;
//   end straight section

//   start dispersion suppressor
   qf: qf,     at=(ncell-4)*lcell;
   mb: mb,     at=(ncell-4)*lcell+0.15*lcell;
   mb: mb,     at=(ncell-4)*lcell+0.35*lcell;
   qd: qd,     at=(ncell-4)*lcell+0.50*lcell;
   mb: mb,     at=(ncell-4)*lcell+0.65*lcell;
   mb: mb,     at=(ncell-4)*lcell+0.85*lcell;

   qf: qf,     at=(ncell-3)*lcell;
   qd: qd,     at=(ncell-3)*lcell+0.50*lcell;
//   end dispersion suppressor

   qf: qf,     at=(ncell-2)*lcell;
   mb: mb,     at=(ncell-2)*lcell+0.15*lcell;
   mb: mb,     at=(ncell-2)*lcell+0.35*lcell;
   qd: qd,     at=(ncell-2)*lcell+0.50*lcell;
   mb: mb,     at=(ncell-2)*lcell+0.65*lcell;
   mb: mb,     at=(ncell-2)*lcell+0.85*lcell;

   qf: qf,     at=(ncell-1)*lcell;
   mb: mb,     at=(ncell-1)*lcell+0.15*lcell;
   mb: mb,     at=(ncell-1)*lcell+0.35*lcell;
   qd: qd,     at=(ncell-1)*lcell+0.50*lcell;
   mb: mb,     at=(ncell-1)*lcell+0.65*lcell;
   mb: mb,     at=(ncell-1)*lcell+0.85*lcell;
end_machine: marker at=circum;
```

```
endsequence;
```

7.2.2 Half-field dispersion suppressor

The half-field dispersion suppressor is conceptually even simpler and for a phase advance of 60^o per cell we need 3 cells on each side of the straight section with half the bending strength. We follow the same strategy as before and use the following lattice description (EX6/casHF.seq):

```
// defines a half-field dispersion suppressor
circum=2000.0;
ncell = 40;
lcell = circum/ncell;
lquad = 3.00;
lquad2 = lquad/2.;
lsex = .0001;

nnorm = 148;

// forces and other constants;
// element definitions;

// define bending magnet as multipole
mb: multipole, lrad=dummy, knl={2.0*pi/nnorm};
// define bending magnet with half strength
mb2: multipole, lrad=dummy, knl={1.0*pi/(nnorm)};

// define the quadrupoles as multipole
kqf =  .980000e-2;
kqd = -.980000e-2;
qf: multipole, lrad=dummy, knl:={0,lquad*kqf};
qd: multipole, lrad=dummy, knl:={0,lquad*kqd};

// define the sextupoles as multipole
ksf :=  +0.017041/20.0;
ksd :=  -0.024714/20.0;

// ATTENTION: must use knl:=  not knl= to match !
msf: multipole, lrad=dummy, knl:={0,0,ksf};
msd: multipole, lrad=dummy, knl:={0,0,ksd};

// sequence declaration;
cascell6: sequence, refer=centre, l=circum;
start_machine: marker, at = 0;
!
   n = 1;
   while (n < ncell-9) {
   qf: qf,     at=(n-1)*lcell;
   msf: msf,   at=(n-1)*lcell + lsex/2.0;
   mb: mb,     at=(n-1)*lcell+0.15*lcell;
   mb: mb,     at=(n-1)*lcell+0.35*lcell;
   qd: qd,     at=(n-1)*lcell+0.50*lcell;
   msd: msd,   at=(n-1)*lcell+0.50*lcell + lsex/2.0;
   mb: mb,     at=(n-1)*lcell+0.65*lcell;
   mb: mb,     at=(n-1)*lcell+0.85*lcell;
!
   n = n + 1;
}
   qf: qf,     at=(ncell-10)*lcell;
   mb: mb,     at=(ncell-10)*lcell+0.15*lcell;
   mb: mb,     at=(ncell-10)*lcell+0.35*lcell;
   qd: qd,     at=(ncell-10)*lcell+0.50*lcell;
   mb: mb,     at=(ncell-10)*lcell+0.65*lcell;
   mb: mb,     at=(ncell-10)*lcell+0.85*lcell;

//   begin dispersion suppressor
   qf: qf,     at=(ncell-9)*lcell;
   mb2: mb2,   at=(ncell-9)*lcell+0.15*lcell;
   mb2: mb2,   at=(ncell-9)*lcell+0.35*lcell;
   qd: qd,     at=(ncell-9)*lcell+0.50*lcell;
   mb2: mb2,   at=(ncell-9)*lcell+0.65*lcell;
   mb2: mb2,   at=(ncell-9)*lcell+0.85*lcell;

   qf: qf,     at=(ncell-8)*lcell;
   mb2: mb2,   at=(ncell-8)*lcell+0.15*lcell;
   mb2: mb2,   at=(ncell-8)*lcell+0.35*lcell;
   qd: qd,     at=(ncell-8)*lcell+0.50*lcell;
   mb2: mb2,   at=(ncell-8)*lcell+0.65*lcell;
   mb2: mb2,   at=(ncell-8)*lcell+0.85*lcell;

   qf: qf,     at=(ncell-7)*lcell;
   mb2: mb2,   at=(ncell-7)*lcell+0.15*lcell;
   mb2: mb2,   at=(ncell-7)*lcell+0.35*lcell;
   qd: qd,     at=(ncell-7)*lcell+0.50*lcell;
   mb2: mb2,   at=(ncell-7)*lcell+0.65*lcell;
   mb2: mb2,   at=(ncell-7)*lcell+0.85*lcell;
//   end dispersion suppressor

//   begin straight section
   qf: qf,     at=(ncell-6)*lcell;
   qd: qd,     at=(ncell-6)*lcell+0.50*lcell;

   qf: qf,     at=(ncell-5)*lcell;
   qd: qd,     at=(ncell-5)*lcell+0.50*lcell;
//   end straight section

//   begin dispersion suppressor
   qf: qf,     at=(ncell-4)*lcell;
   mb2: mb2,   at=(ncell-4)*lcell+0.15*lcell;
   mb2: mb2,   at=(ncell-4)*lcell+0.35*lcell;
   qd: qd,     at=(ncell-4)*lcell+0.50*lcell;
   mb2: mb2,   at=(ncell-4)*lcell+0.65*lcell;
   mb2: mb2,   at=(ncell-4)*lcell+0.85*lcell;

   qf: qf,     at=(ncell-3)*lcell;
   mb2: mb2,   at=(ncell-3)*lcell+0.15*lcell;
   mb2: mb2,   at=(ncell-3)*lcell+0.35*lcell;
   qd: qd,     at=(ncell-3)*lcell+0.50*lcell;
   mb2: mb2,   at=(ncell-3)*lcell+0.65*lcell;
   mb2: mb2,   at=(ncell-3)*lcell+0.85*lcell;

   qf: qf,     at=(ncell-2)*lcell;
   mb2: mb2,   at=(ncell-2)*lcell+0.15*lcell;
   mb2: mb2,   at=(ncell-2)*lcell+0.35*lcell;
   qd: qd,     at=(ncell-2)*lcell+0.50*lcell;
   mb2: mb2,   at=(ncell-2)*lcell+0.65*lcell;
   mb2: mb2,   at=(ncell-2)*lcell+0.85*lcell;
//   end dispersion suppressor

   qf: qf,     at=(ncell-1)*lcell;
   mb: mb,     at=(ncell-1)*lcell+0.15*lcell;
   mb: mb,     at=(ncell-1)*lcell+0.35*lcell;
   qd: qd,     at=(ncell-1)*lcell+0.50*lcell;
   mb: mb,     at=(ncell-1)*lcell+0.65*lcell;
   mb: mb,     at=(ncell-1)*lcell+0.85*lcell;
end_machine: marker at=circum;

endsequence;
```

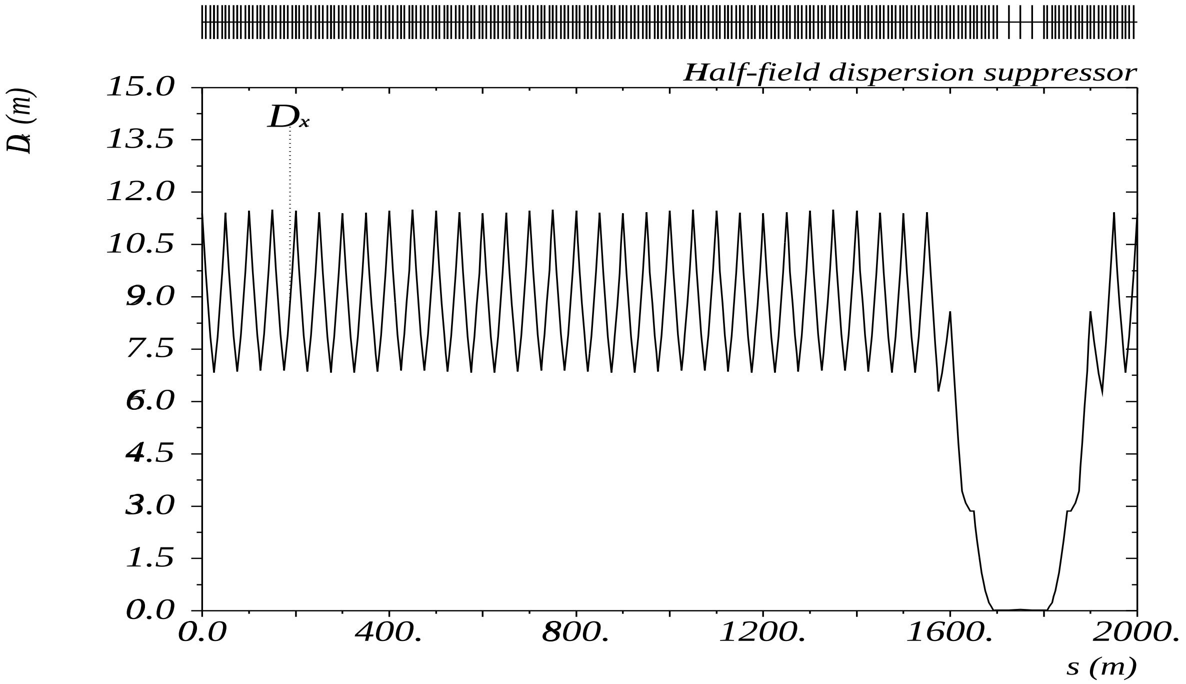

Figure 2: Horizontal dispersion with half-field dispersion suppressor.

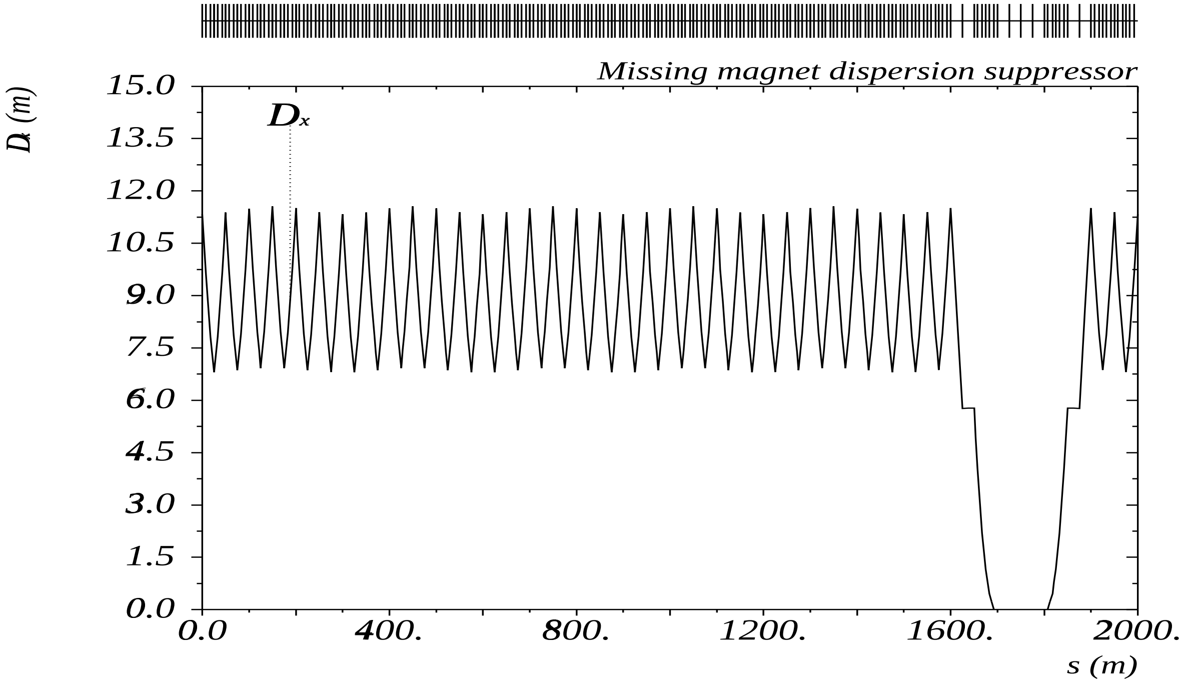

Figure 3: Horizontal dispersion with missing magnet dispersion suppressor.

We can see from Figs.2 and 3 that the dispersion D$_x$ is reduced as wanted. In a second step one would use individual control of the quadrupoles in the straight section and dispersions suppressors to get the desired properties and to make the matching perfect.

8 Exercise 7

8.1 Problem:

Start from the previous lattice and design a symmetric insertion with a low-β section in a dispersion free region. The β should be smallest and should have a waist at an ”interaction point”. Try to follow these steps:

- Insert an insertion into the previous lattice, using the optics with the half field dispersion suppressor. For that, replace the ”straight section” of the previous exercise.
- Leave 8.5 m on each side of the interaction point empty for possible equipment.
- Design a fully symmetric insertion with 4 independent quadrupoles on each side. The corresponding quadrupoles on the left and the right side should have the same strengths.
- Calculate the initial settings for these quadrupoles for reasonable values of β_x and β_y and a waist at the interaction point. Correct a possible β-beating.
- Compute the settings for your low-β insertion to get a small ratio of β_x/β_y or β_y/β_x. The ratio is limited by the maximum β-function in the quadrupoles. Which is the smallest ratio when $\hat{\beta}$ in the quadrupoles is limited to 1000 m ?

8.2 Solution:

To place an insertion, we cut the lattice and add the desired elements between the ends.

8.2.1 Element layout

For the low-β insertion we would like a fully symmetric layout, i.e. it should start and end on a quadrupole of the same type. We therefore rearrange the cells slightly to start and end the insertion at a defocusing quadrupole. This will serve as a reference, i.e. any change of the optics inside must not effect the optics at these limiting quadrupoles. Doing this, we increase the circumference by the length of one half cell. The matching of the insertion will ensure that changes will not propagate outside into the regular lattice. For that purpose we insert markers (LEFT, RIGHT) into the lattice at their position. We have inserted another marker (MARK1) to indicate the end of the regular cells. We choose the QD of the cell 36 as the left and the QD of cell 37 as the right end. The QF of cell 37 becomes part of the insertion (QF4.R). For the insertion we use 4 symmetric quadrupoles on each side, equally spaced by half the cell length. The distance l^* between the centre of the insertion (IP) and the closest quadrupole depends on the space requirements. We require 8.5 m free space on each side and therefore l^* = 8.5 m + quadrupole half length = 10.0 m.

The total length of the insertion becomes 3.5 times the cell length (we use one half cell for the insertion) plus 2 times the distance between quadrupole and IP (l^*). The circumference of the machine is increased by this length. We decide to build a fully symmetric insertion, i.e. the strengths of the corresponding quadrupoles left and right of the IP are the same. We define all elements as thin elements. The sequence is now as shown below (EX7/mat1.seq):

```
circum=2000.0;
ncell =  40;
lcell = circum/ncell;
lcell2 = lcell/2.;
lq = 3.00; // quadrupole length
ls = 1.1; // sextupole length
lstar = 10.0;
lins = 4*lcell + 2*lstar-lcell2;
```

```
circum=circum+lins+lcell2;
// forces and other constants;

// element definitions;

nnorm = 146;

ip: marker;

mb:multipole, lrad:= 5.0,  knl:={2.0*pi/nnorm};
mb2:multipole, lrad:= 5.0,  knl:={1.0*pi/nnorm};

qf:multipole, lrad:= dummy,  knl:={0,lq*kqf};
qd:multipole, lrad:= dummy,  knl:={0,lq*kqd};

q1.l: multipole, lrad=dummy, knl:={0,lq*kq1.l};
q2.l: multipole, lrad=dummy, knl:={0,lq*kq2.l};
q3.l: multipole, lrad=dummy, knl:={0,lq*kq3.l};
q4.l: multipole, lrad=dummy, knl:={0,lq*kq4.l};

q1.r: multipole, lrad=dummy, knl:={0,lq*kq1.l};
q2.r: multipole, lrad=dummy, knl:={0,lq*kq2.l};
q3.r: multipole, lrad=dummy, knl:={0,lq*kq3.l};
q4.r: multipole, lrad=dummy, knl:={0,lq*kq4.l};

kq1.r := kq1.l;
kq2.r := kq2.l;
kq3.r := kq3.l;
kq4.r := kq4.l;

// some guessing for the initial strengths
kq1.l  =  -0.01;
kq2.l  =   0.01;
kq3.l  =  -0.01;
kq4.l  =   0.01;

mscbh: sextupole, l=ls, k2:=ksf;
mscbv: sextupole, l=ls, k2:=ksd;

kqf =  1.32813125E-02;
kqd = -1.32893092E-02;


ksf =  .01;
ksd = -.01;

// sequence declaration;

cascell5: sequence, refer=centre, l=circum;
start_machine: marker, at = 0;
```

```
!
   n = 1;
   while (n < ncell-5) {
   qf: qf,        at=(n-1)*lcell + 0.00*lcell ;
   mscbh: mscbh,  at=(n-1)*lcell + 0.00*lcell + ls/2.0;
   mb: mb,        at=(n-1)*lcell + 0.15*lcell ;
   mb: mb,        at=(n-1)*lcell + 0.35*lcell ;
   qd: qd,        at=(n-1)*lcell + 0.50*lcell ;
   mscbv: mscbv,  at=(n-1)*lcell + 0.50*lcell + ls/2.0;
   mb: mb,        at=(n-1)*lcell + 0.65*lcell ;
   mb: mb,        at=(n-1)*lcell + 0.85*lcell ;
!
   n = n + 1;
}
// begin dispersion suppressor
   qf: qf,        at=(ncell-6)*lcell + 0.00*lcell ;
   mark1: marker, at=(ncell-6)*lcell + 0.00*lcell ;
   mb2: mb2,      at=(ncell-6)*lcell + 0.15*lcell ;
   mb2: mb2,      at=(ncell-6)*lcell + 0.35*lcell ;
   qd: qd,        at=(ncell-6)*lcell + 0.50*lcell ;
   mb2: mb2,      at=(ncell-6)*lcell + 0.65*lcell ;
   mb2: mb2,      at=(ncell-6)*lcell + 0.85*lcell ;
!
   qf: qf,        at=(ncell-5)*lcell + 0.00*lcell ;
   mb2: mb2,      at=(ncell-5)*lcell + 0.15*lcell ;
   mb2: mb2,      at=(ncell-5)*lcell + 0.35*lcell ;
   qd: qd,        at=(ncell-5)*lcell + 0.50*lcell ;
   mb2: mb2,      at=(ncell-5)*lcell + 0.65*lcell ;
   mb2: mb2,      at=(ncell-5)*lcell + 0.85*lcell ;
!
   qf: qf,        at=(ncell-4)*lcell + 0.00*lcell ;
   mb2: mb2,      at=(ncell-4)*lcell + 0.15*lcell ;
   mb2: mb2,      at=(ncell-4)*lcell + 0.35*lcell ;
   qd: qd,        at=(ncell-4)*lcell + 0.50*lcell ;
   left: marker,  at=(ncell-4)*lcell + 0.50*lcell ;
   mb2: mb2,      at=(ncell-4)*lcell + 0.65*lcell ;
   mb2: mb2,      at=(ncell-4)*lcell + 0.85*lcell ;
// end dispersion suppressor
!
// begin straight section, low beta insertion
   q4.l: q4.l,    at=(ncell-3)*lcell + 0.00*lcell ;
   q3.l: q3.l,    at=(ncell-3)*lcell + 0.50*lcell ;
   q2.l: q2.l,    at=(ncell-3)*lcell + 1.00*lcell ;
   q1.l: q1.l,    at=(ncell-3)*lcell + 1.50*lcell ;

   IP: IP,        at=(ncell-3)*lcell + 1.50*lcell +lstar;

   q1.r: q1.r,    at=(ncell-3)*lcell +2*lstar + 1.50*lcell ;
   q2.r: q2.r,    at=(ncell-3)*lcell +2*lstar + 2.00*lcell ;
   q3.r: q3.r,    at=(ncell-3)*lcell +2*lstar + 2.50*lcell ;
   q4.r: q4.r,    at=(ncell-3)*lcell +2*lstar + 3.00*lcell ;
// end straight section, low beta insertion
!
// begin dispersion suppressor
   qd: qd,        at=lins+(ncell-3)*lcell + 0.00*lcell ;
   right: marker, at=lins+(ncell-3)*lcell + 0.00*lcell ;
   mb2: mb2,      at=lins+(ncell-3)*lcell + 0.15*lcell ;
   mb2: mb2,      at=lins+(ncell-3)*lcell + 0.35*lcell ;
   qf: qf,        at=lins+(ncell-3)*lcell + 0.50*lcell ;
   mb2: mb2,      at=lins+(ncell-3)*lcell + 0.65*lcell ;
   mb2: mb2,      at=lins+(ncell-3)*lcell + 0.85*lcell ;
!
   qd: qd,        at=lins+(ncell-2)*lcell + 0.00*lcell ;
   mb2: mb2,      at=lins+(ncell-2)*lcell + 0.15*lcell ;
   mb2: mb2,      at=lins+(ncell-2)*lcell + 0.35*lcell ;
   qf: qf,        at=lins+(ncell-2)*lcell + 0.50*lcell ;
   mb2: mb2,      at=lins+(ncell-2)*lcell + 0.65*lcell ;
   mb2: mb2,      at=lins+(ncell-2)*lcell + 0.85*lcell ;
!
   qd: qd,        at=lins+(ncell-1)*lcell + 0.00*lcell ;
   mb2: mb2,      at=lins+(ncell-1)*lcell + 0.15*lcell ;
   mb2: mb2,      at=lins+(ncell-1)*lcell + 0.35*lcell ;
   qf: qf,        at=lins+(ncell-1)*lcell + 0.50*lcell ;
   mb2: mb2,      at=lins+(ncell-1)*lcell + 0.65*lcell ;
   mb2: mb2,      at=lins+(ncell-1)*lcell + 0.85*lcell ;
// end dispersion suppressor
!
!
// start of regular cell
   qd: qd,        at=lins+(ncell)*lcell + 0.00*lcell ;
   mscbv: mscbv,  at=lins+(ncell)*lcell + 0.00*lcell + ls/2.0;
   mb: mb,        at=lins+(ncell)*lcell + 0.15*lcell ;
   mb: mb,        at=lins+(ncell)*lcell + 0.35*lcell ;
end_machine: marker at=circum;
endsequence;
```

8.2.2 Initial parameter settings

The initial strength of the insertion quadrupoles has to be calculated or matched with MAD. This can be done with the following command sequence (EX7/mat1a.mad):

```
TITLE, s='MAD-X test';
call file="mat1.seq";
option,-echo;

Beam, particle = proton, sequence=cascell5, energy = 20.0;

use, period=cascell5;

match, sequence=cascell5,range=left/IP,betx=28.16,alfx=-0.596,
                                       bety=86.97,alfy=1.7373;
```

```
   vary,name=kq1.l, step=0.00001;
   vary,name=kq2.l, step=0.00001;
   vary,name=kq3.l, step=0.00001;
   vary,name=kq4.l, step=0.00001;
   constraint,range=left,sequence=cascell5,betx=28.2,bety=87.0;
   constraint,range=IP,sequence=cascell5,betx=20.0,alfx=0.0,
                                             bety=80.0,alfy=0.0;
   constraint,range=right,sequence=cascell5,betx=28.2,bety=87.0;
   Lmdif, calls=100, tolerance=1.0e-21;
endmatch;

select,flag=twiss,clear;
select,flag=twiss,column=name,s,x,y,mux,betx,alfx,muy,
                                        bety,alfy,dx,dpx;
twiss,save,file=twiss0.out;
plot, haxis=s, vaxis=betx, bety;
plot, haxis=s, vaxis=dx;
plot, haxis=s, vaxis=betx, bety,range=mark1/end_machine;
plot, haxis=s, vaxis=dx,range=mark1/end_machine;

match, sequence=cascell5;
   vary,name=kqf, step=0.00001;
   vary,name=kqd, step=0.00001;
   constraint,pattern="ˆqf.*",sequence=cascell5,betx=86.5,bety=29.8;
   constraint,pattern="ˆqd.*",sequence=cascell5,bety=86.5,betx=29.8;
   Lmdif, calls=100, tolerance=1.0e-21;
endmatch;

select,flag=twiss,clear;
select,flag=twiss,column=name,s,x,y,mux,betx,alfx,muy,
                                        bety,alfy,dx,dpx;
twiss,save,file=twiss0.out;
plot, haxis=s, vaxis=betx, bety;
plot, haxis=s, vaxis=dx;
plot, haxis=s, vaxis=betx, bety,range=mark1/end_machine;
plot, haxis=s, vaxis=dx,range=mark1/end_machine;

stop;
```

Here we consider only the lattice between our markers $left$ and $right$. This significantly simplifies the matching. However, now we do not consider a closed solution and we have to give the initial optics parameters at the beginning of the range. The Twiss functions β_x and β_y at these markers (i.e. the QD) we know from the unperturbed lattice. The β_x and β_y at the third marker (the IP) we set initially to 20 and 80 metres. These are typical values for the l* = 10 m we have chosen.

For the correction of the β beating we have varied the strength of the main quadrupoles QD and QF and requested the horizontal and vertical β-functions the same at all quadrupoles, i.e. no beating at all.

Below we show an alternative matching procedure. Since our insertion is defined fully symmetric, it is sufficient to match only one half like (the constraint $\alpha_x = \alpha_y = 0$ will enforce the symmetry) (EX7/mat1c.mad):

```
TITLE, s=’MAD-X test’;
```

```
call file="mat1.seq";
option,-echo;

Beam, particle = proton, sequence=cascell5, energy = 20.0;

use, period=cascell5;

match, sequence=cascell5,range=left/IP,betx=28.16,alfx=-0.596,
                                       bety=86.97,alfy=1.7373;
   vary,name=kq1.l, step=0.00001;
   vary,name=kq2.l, step=0.00001;
   vary,name=kq3.l, step=0.00001;
   vary,name=kq4.l, step=0.00001;
   constraint,range=left,sequence=cascell5,betx=28.2,bety=87.0;
   constraint,range=IP,sequence=cascell5,betx=20.0,alfx=0.0,
                                         bety=80.0,alfy=0.0;
   Lmdif, calls=100, tolerance=1.0e-21;
endmatch;

select,flag=twiss,clear;
select,flag=twiss,column=name,s,x,y,mux,betx,alfx,muy,
                                        bety,alfy,dx,dpx;
savebeta, label=out, place=right;
twiss,save,file=twiss0.out;
plot, haxis=s, vaxis=betx, bety;
plot, haxis=s, vaxis=dx;
plot, haxis=s, vaxis=betx, bety,range=mark1/end_machine;
plot, haxis=s, vaxis=dx,range=mark1/end_machine;

match, sequence=cascell5;
   vary,name=kqf, step=0.00001;
   vary,name=kqd, step=0.00001;
   constraint,pattern="ˆqf.*",sequence=cascell5,betx=out->bety,
                                                bety=out->betx;
   constraint,pattern="ˆqd.*",sequence=cascell5,betx=out->betx,
                                                bety=out->bety;
   Lmdif, calls=100, tolerance=1.0e-21;
endmatch;

select,flag=twiss,clear;
select,flag=twiss,column=name,s,x,y,mux,betx,alfx,muy,
                                        bety,alfy,dx,dpx;
twiss,save,file=twiss1.out;
plot, haxis=s, vaxis=betx, bety;
plot, haxis=s, vaxis=dx;
plot, haxis=s, vaxis=betx, bety,range=mark1/end_machine;
plot, haxis=s, vaxis=dx,range=mark1/end_machine;

stop;
```

In this example we also demonstrate the use of the SAVEBETA option. The optics parameters can be calculated at a specified position (PLACE), given a name (LABEL) and re-used in later calculations (e.g. as matching constraints) like in our example. It can be used with the BETA0 command or by directly accessing the various parameters, like in our example.

For the matching we have to give some well educated guess for the strengths of the quadrupoles we want to match. In particular the sign and the order of magnitude of the strength must be approximately correct. Starting with no or totally wrong values could lead to unphysical or no result of the matching procedure. We find for the strengths:

```
kq1a.l =  -8.70994188E-03;
kq2.l  =   1.48084944E-02;
kq3.l  =  -1.45673434E-02;
kq4.l  =   1.19074354E-02;
```

and implement them into the sequence (EX7/mat2.seq). In the Figs.4 to 7 we have the graphical results of our effort. We show the horizontal and vertical β-functions around the insertion and for the full machine in Figs.4 and 5. The functions are symmetric as wanted and matched to the arcs.

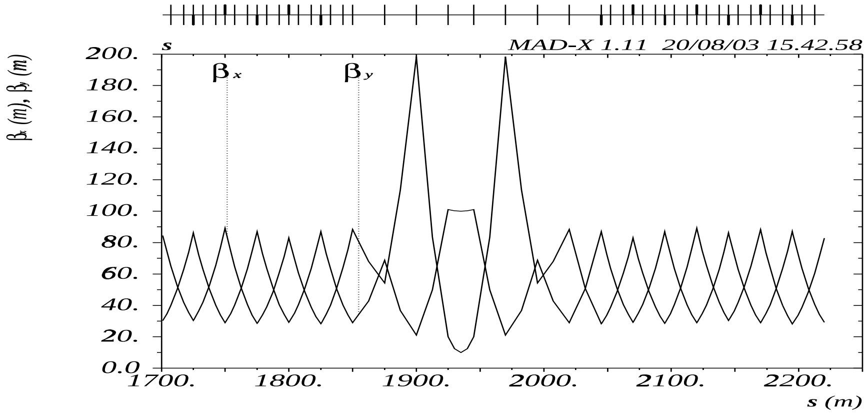

Figure 4: Matched β^* section. We show the horizontal and vertical β-function around the insertion.

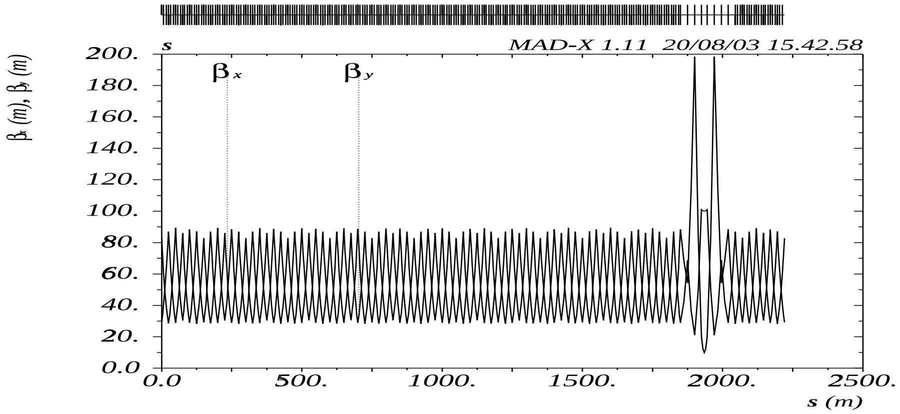

Figure 5: Matched β^* section. We show the horizontal and vertical β-function for the full machine.

In the Figs.6 and 7 we show the horizontal dispersion D_x. Over the full length of the insertion the dispersion is practically zero and fully matched to the machine.

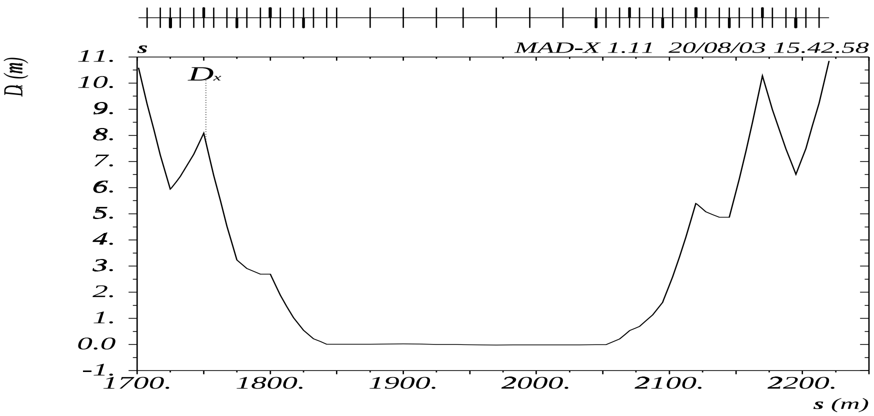

Figure 6: Matched β^* section. We show the horizontal D$_x$-function around the insertion.

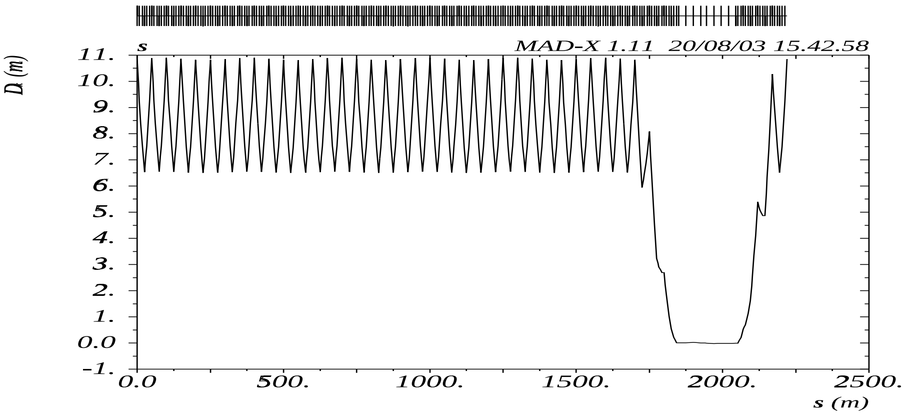

Figure 7: Matched β^* section. We show the horizontal D$_x$-function for the full machine.

8.2.3 Tuning range (flat beams)

We would like to increase the ratio of the vertical and horizontal β-functions and find the possible limits. After inserting the initial values into the sequence declaration, we can set up the matching of the insertion like (EX7/mat2.mad):

```
TITLE, s='MAD-X test';
call file="mat2.seq";
option,-echo;

Beam, particle = proton, sequence=cascell5, energy = 20.0;

use, period=cascell5;

select,flag=twiss,clear;
select,flag=twiss,column=name,s,x,y,mux,betx,alfx,
                                    muy,bety,alfy,dx,dpx;
savebeta, label = beta0_ini, place = start_machine;
twiss,file=twiss1.out;
plot, haxis=s, vaxis=betx, bety;
plot, haxis=s, vaxis=dx;
```

```
match, sequence=cascell5,beta0=beta0_ini;
   vary,name=kq1.l, step=0.00001;
   vary,name=kq2.l, step=0.00001;
   vary,name=kq3.l, step=0.00001;
   vary,name=kq4.l, step=0.00001;
   constraint,range=ip,sequence=cascell5,betx=2.0,bety=600,alfx=0.0,
                                        alfy=0.0,dx=0.0,dpx=0.0;
   Lmdif, calls=100, tolerance=1.0e-21;
endmatch;

select,flag=twiss,clear;
select,flag=twiss,column=name,s,x,y,mux,betx,muy,bety,dx,dy;
twiss,save,centre,file=twiss2.out;
plot, haxis=s, vaxis=betx, bety;
plot, haxis=s, vaxis=dx;
plot, haxis=s, vaxis=betx, bety,range=mark1/end_machine;
plot, haxis=s, vaxis=dx,range=mark1/end_machine;

stop;
```

We show again another possibility for the matching strategy. We use the SAVEBETA option to store the optics parameters at the beginning of the machine and start from there. We require the derivatives of both β-functions to be zero and define the wanted horizontal β-function β_x in the matching procedure. The vertical β-function β_y will automatically adjust. As an illustration we show the β-functions for the full machine and in the insertion for β_x = 2.0 m and β_y = 600.0 m in the Figs.8 and 9. (This is not the maximum ratio or optimum, just for illustration.)

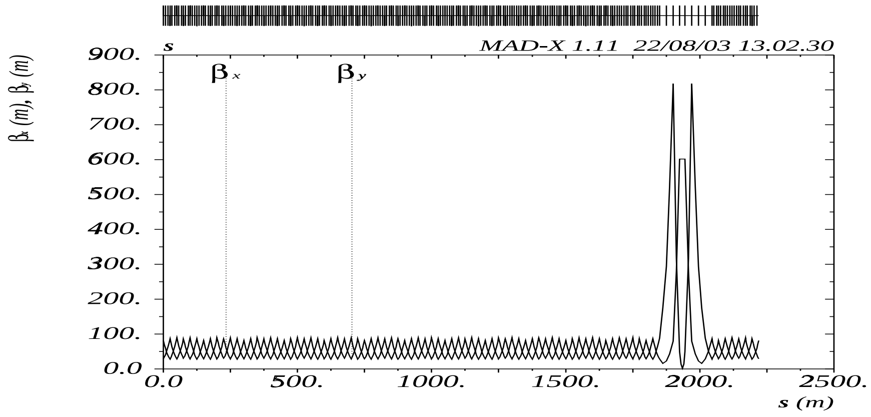

Figure 8: Matched β^* section. We show the horizontal and vertical β-functions for the full machine.

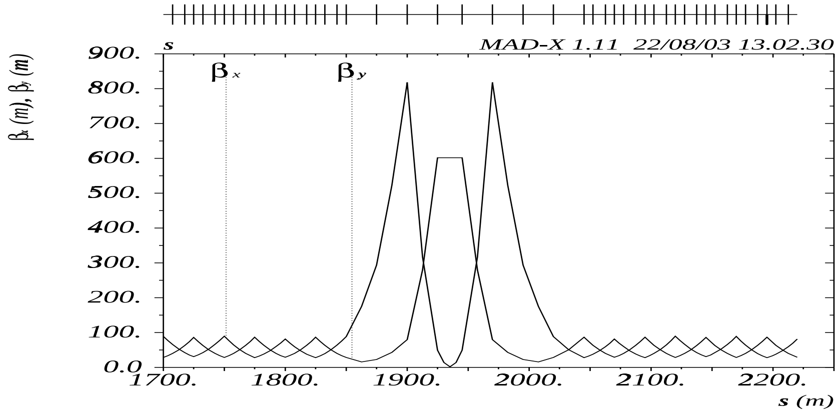

Figure 9: Matched β^* section. We show the horizontal and vertical β-functions in the insertion.

9 Exercise 8

9.1 Problem:

Use the lattice from exercise 7 and try to answer the following questions.

- Can you make the ratio of β_x/β_y or β_y/β_x to become 1 (e.g. $\beta_x = \beta_y$ = 50 m) ? What is the best you can do ?
- Replace the innermost quadrupole by a quadrupole doublet and re-assess the last question.
- For discussion only: what steps do you propose to get something like $\beta_x = \beta_y$ = 10 m (or smaller) ?

9.2 Solution:

While flat beams with a large β ratio are typically used in e^+e^- colliders, in hadron colliders we often use round beams, i.e. the horizontal and vertical beam sizes are the same or very similar. Our setup is not made for such a scheme but small improvements are possible. In Figs.10 and 11 we have tried to match β_x and β_y both to β_x = β_y = 50 m. It can be seen that we did not fully succeed and furthermore, we have created a very strong beating around the machine that cannot be corrected. We shall try to

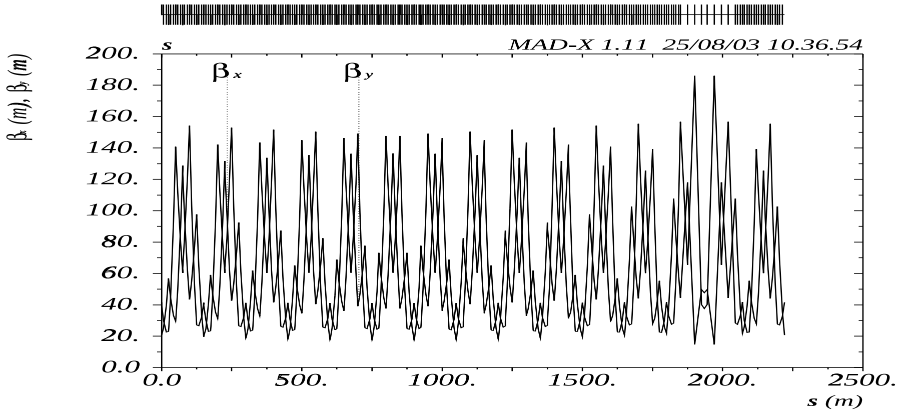

Figure 10: Match round beams at interaction point with a single quadrupole magnet Q1. We show the horizontal and vertical β-functions for the full machine. $\beta_x^* = \beta_y^*$ = 50.0 m.

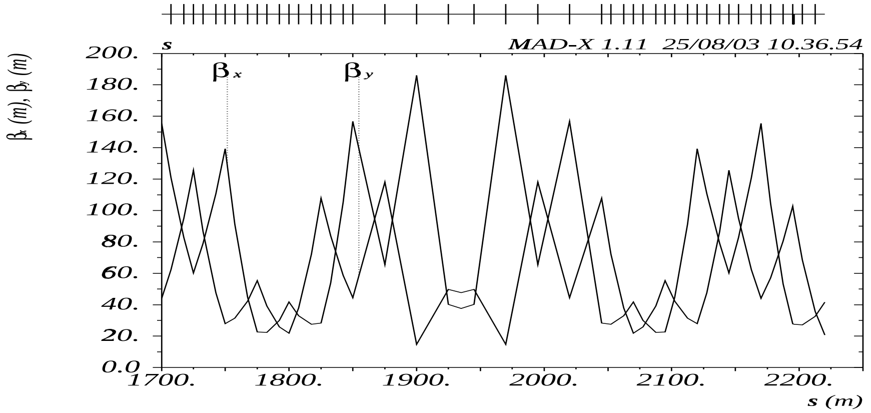

Figure 11: Match round beams at interaction point with a single quadrupole magnet Q1. We show the horizontal and vertical β-functions in the insertion. $\beta_x^* = \beta_y^*$ = 50.0 m.

improve the layout slightly by adding another quadrupole next to the innermost insertion quadrupole like (EX8/matdoub.seq):

```
   q4.l: q4.l,    at=(ncell-3)*lcell + 0.00*lcell ;
   q3.l: q3.l,    at=(ncell-3)*lcell + 0.50*lcell ;
   q2.l: q2.l,    at=(ncell-3)*lcell + 1.00*lcell ;
// additional quadrupole Q1A.L
   q1a.l: q1a.l,  at=(ncell-3)*lcell + 1.50*lcell - lq;
   q1.l: q1.l,    at=(ncell-3)*lcell + 1.50*lcell ;

   IP: IP,        at=(ncell-3)*lcell + 1.50*lcell +lstar;

   q1.r: q1.r,    at=(ncell-3)*lcell +2*lstar + 1.50*lcell ;
// additional quadrupole Q1A.R
   q1a.r: q1a.r,  at=(ncell-3)*lcell +2*lstar + 1.50*lcell + lq;
   q2.r: q2.r,    at=(ncell-3)*lcell +2*lstar + 2.00*lcell ;
   q3.r: q3.r,    at=(ncell-3)*lcell +2*lstar + 2.50*lcell ;
   q4.r: q4.r,    at=(ncell-3)*lcell +2*lstar + 3.00*lcell ;
```

We let the strength be independent but with left right symmetry. The result of this additional quadrupole can be seen in Figs.12 and 13. This time the betatron functions can easily be matched to $\beta_x^* = \beta_y^*$ = 50.0 m. Not only are the β-functions equal, also the induced beating is largely reduced to an acceptable level.

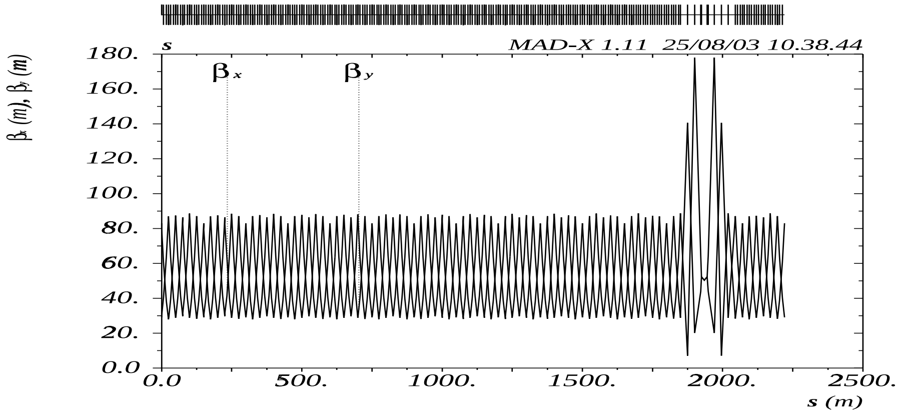

Figure 12: Match round beams at interaction point with Q1 doublet magnet. We show the horizontal and vertical β-functions for the full machine. $\beta_x^* = \beta_y^*$ = 50.0 m.

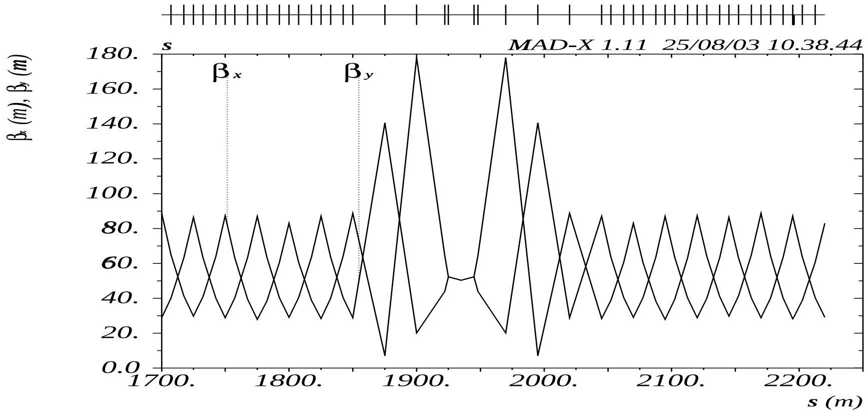

Figure 13: Match round beams at interaction point with Q1 doublet magnet. We show the horizontal and vertical β-functions in the insertion. $\beta^*_x = \beta^*_y$ = 50.0 m.

10 Exercise 9

10.1 Problem:

For the operation the optics usually needs some ”knobs” for fine tuning, such as tune or dispersion.

- The dispersion can become slightly mismatched. Insert individually powered quadrupoles into the dispersion suppressor to control the dispersion beating.

10.2 Solution:

10.2.1 Fine tuning of horizontal dispersion

We can further improve the lattice by adding matching quadrupoles in the dispersion suppressor region to control the horizontal dispersion and finally matching the chromaticity to zero: The additional quadrupoles in the dispersion suppressor region become necessary when the phase advance per cell deviates from the nominal 60^o. We use the round beam option with the doublet quadrupole from the previous exercise for this optimization (EX9/matfine.seq and EX9/matfine.mad).

```
TITLE, s='MAD-X test';
call file="matfine.seq";
option,-echo;

Beam, particle = proton, sequence=cascell5, energy = 20.0;

use, period=cascell5;

match, sequence=cascell5,range=left/IP,betx=28.16,alfx=-0.596,
                                       bety=86.97,alfy=1.7373;
   vary,name=kq1.l, step=0.00001;
   vary,name=kq2.l, step=0.00001;
   vary,name=kq3.l, step=0.00001;
   vary,name=kq4.l, step=0.00001;
   vary,name=kq1a.l, step=0.00001;
   constraint,range=left,sequence=cascell5,betx=28.2,bety=87.0;
   constraint,range=IP,sequence=cascell5,alfx=0.0,
                                         betx= 50.0,
                                         bety= 50.0,
                                         alfy=0.0;
   constraint,range=right,sequence=cascell5,betx=28.2,bety=87.0;
   Lmdif, calls=100, tolerance=1.0e-21;
endmatch;

select,flag=twiss,clear;
select,flag=twiss,column=name,s,x,y,mux,betx,alfx,muy,bety,alfy,dx,dpx;
savebeta,label=beta0_ini, place=start_machine;
twiss,save,file=twiss1.out;
plot, haxis=s, vaxis=betx, bety;
plot, haxis=s, vaxis=dx;
plot, haxis=s, vaxis=betx, bety,range=mark1/end_machine;
plot, haxis=s, vaxis=dx,range=mark1/end_machine;

match, sequence=cascell5,beta0=beta0_ini;
```

```
!  vary,name=kq1a.l, step=0.00001;
!  vary,name=kq1.l, step=0.00001;
!  vary,name=kq2.l, step=0.00001;
!  vary,name=kq3.l, step=0.00001;
!  vary,name=kq4.l, step=0.00001;
   vary,name=kqfd1l, step=0.00001;   // match quads for dispersion control
   vary,name=kqfd2l, step=0.00001;   // match quads for dispersion control
   constraint,range=ip,sequence=cascell5,betx=50.0;
   constraint,range=ip,sequence=cascell5,bety=50.0;
   constraint,range=ip,sequence=cascell5,alfx=00.0;
   constraint,range=ip,sequence=cascell5,alfy=00.0;
   constraint,range=ip,sequence=cascell5,dx=0.0;
   constraint,range=ip,sequence=cascell5,dpx=0.0;
   Lmdif, calls=100, tolerance=1.0e-21;
endmatch;

// finally get the chromaticity right
match, sequence=cascell5;
   vary,name=ksf, step=0.00001;
   vary,name=ksd, step=0.00001;
   global,sequence=cascell5,DQ1=0.0;
   global,sequence=cascell5,DQ2=0.0;
   Lmdif, calls=10, tolerance=1.0e-21;
endmatch;

select,flag=twiss,clear;
select,flag=twiss,column=name,s,x,y,mux,betx,alfx,muy,bety,alfy,dx,dpx;
twiss,save,file=twiss2.out;
plot, haxis=s, vaxis=betx, bety;
plot, haxis=s, vaxis=dx,vmin=0.0,vmax=15.0;
plot, haxis=s, vaxis=betx, bety,range=mark1/end_machine;
plot, haxis=s, vaxis=dx,vmin=0.0,vmax=15.0,range=mark1/end_machine;

stop;
```

The Figs. 14 and 15 show the result of this D_x optimisation. There is no significant mismatch around the whole machine.

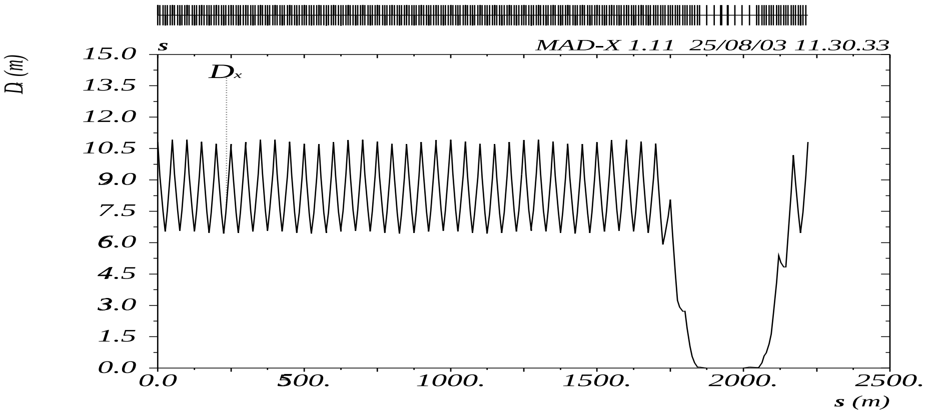

Figure 14: Matched β^* section. We show the horizontal dispersion around the machine.

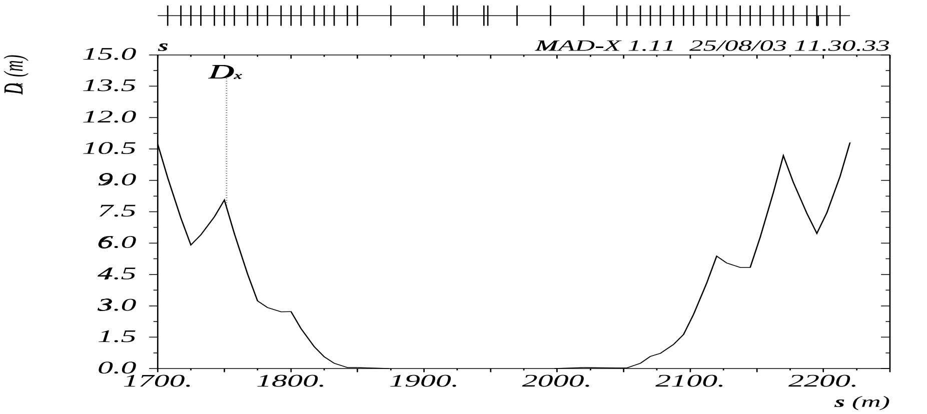

Figure 15: Matched β^* section. We show the horizontal dispersion in the insertion.

References

[1] *The MAD-X Home Page, version February 2003*,
$http://frs.home.cern.ch/frs/Xdoc/mad-X.html$.

[2] B. Holzer, *Lattice cells*, this school.

[3] B. Holzer, *Insertions*, this school.

[4] O. Brüning, *Linear imperfections*, this school.

[5] O. Brüning, *Non-linear imperfections*, this school.

[6] W. Herr, *MAD-X for pedestrian*, this school.

[7] W. Herr, *A MAD-X Primer*, to be published.

[8] $http://cern.ch/Werner.Herr/COURSE/$

[9] *CAS 2003, Optics Design*, CD-ROM (LINUX) with the material of the course, CERN, Geneva 2003.