Lecture 7
Lattice design with MAD-X

Dr. Suzie Sheehy
suzie.sheehy@physics.ox.ac.uk
@suziesheehy

JAI, Oxford
& STFC/RAL/ASTeC

Lecture 8 – MADX introduction and examples

- This lecture is based on one by Ted Wilson (CERN) which was in turn based on a lecture by V. Ziemann (Uppsala University)
  - Installing/running MADX (on windows)
  - Input of elements and beamlines
  - Beta functions, tunes, dispersion
  - Matching
  - Examples

All credit to V. Ziemann for example input files
More examples are on the MAD-X website!

What is MAD-X?

"A program for accelerator design and simulation with a long history"
Developed from previous versions (MAD, MAD-8, then finally MAD-X in 2002).
User guide: http://mad.web.cern.ch/mad/uguide.html

- Uses a sequence of elements placed sequentially along a *reference orbit*
- *Reference orbit* is path of a charged particle having the central design momentum of the accelerator through idealised magnets (no fringe fields)
- The reference orbit consists of a series of straight line segments and circular arcs
- local curvilinear right handed coordinate system *(x,y,s)*

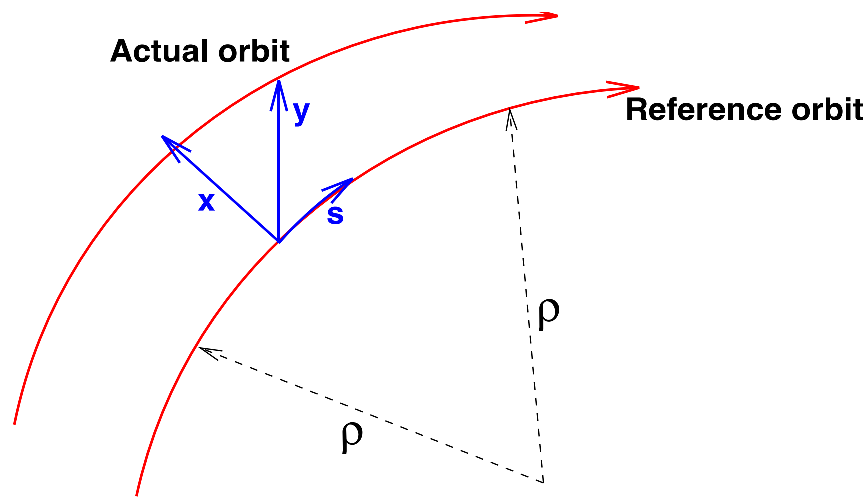


Figure 1: *Local coordinate system as used by MAD-X.*

What can it do?

- You can input magnets (& electrostatic elements) according to the manual
  - User guide: http://madx.web.cern.ch/madx/
- Calculate beta functions, tune, dispersion, chromaticity, momentum compaction numerically.
- Generates tables and plots (.ps)
- (tip: you might need to install ghostscript to view plots)

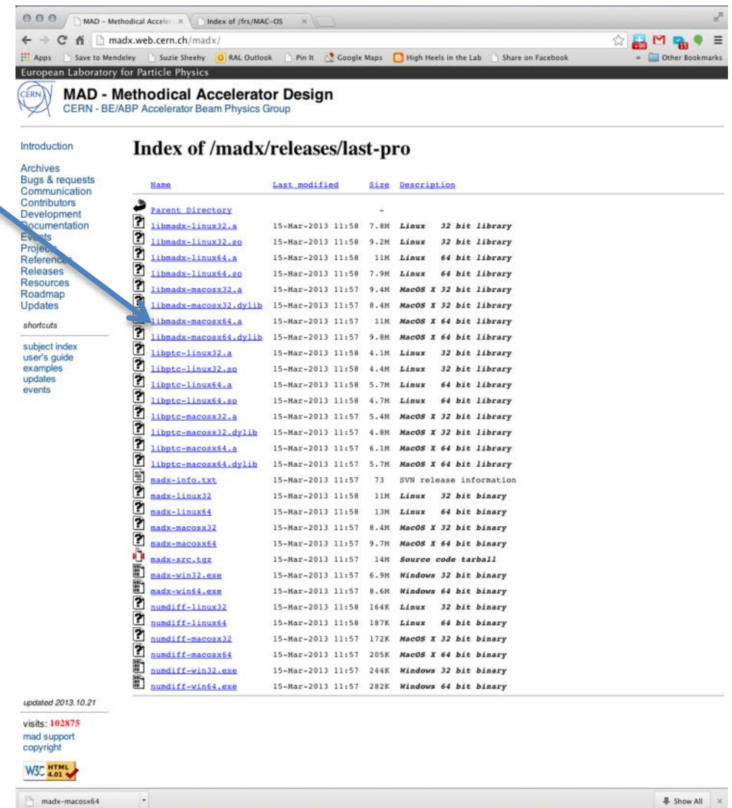
Why choose MAD-X?

- There are any number of tracking and beam optics codes, but MAD-X is *widely used* (especially at CERN), *well maintained* and *well documented*.
  - The more lattice design you do – the more you will appreciate this about MAD-X!
- What it can't do:
  - Acceleration and tracking simultaneously
  - Not so accurate at large excursions from closed orbit (as in an FFAG)
  - Complicated magnet geometries
  - Field maps

Installing MADX

http://madx.web.cern.ch/madx/

Go to 'Releases'
Get latest production version
Choose your system & download:



MAD - Methodical Accelerator Design
CERN - BE/ABP Accelerator Beam Physics Group

Index of /madx/releases/last-pro

Name	Last modified	Size	Description
Parent Directory		-	
libmadx-linux32.a	15-Mar-2013 11:58	7.8M	Linux 32 bit library
libmadx-linux32.so	15-Mar-2013 11:58	9.2M	Linux 32 bit library
libmadx-linux64.a	15-Mar-2013 11:58	11M	Linux 64 bit library
libmadx-linux64.so	15-Mar-2013 11:58	7.9M	Linux 64 bit library
libmadx-macosx32.a	15-Mar-2013 11:57	9.4M	MacOS X 32 bit library
libmadx-macosx32.dylib	15-Mar-2013 11:57	8.4M	MacOS X 32 bit library
libmadx-macosx64.a	15-Mar-2013 11:57	11M	MacOS X 64 bit library
libmadx-macosx64.dylib	15-Mar-2013 11:57	9.8M	MacOS X 64 bit library
libptc-linux32.a	15-Mar-2013 11:58	4.1M	Linux 32 bit library
libptc-linux32.so	15-Mar-2013 11:58	4.4M	Linux 32 bit library
libptc-linux64.a	15-Mar-2013 11:58	5.7M	Linux 64 bit library
libptc-linux64.so	15-Mar-2013 11:58	4.7M	Linux 64 bit library
libptc-macosx32.a	15-Mar-2013 11:57	5.4M	MacOS X 32 bit library
libptc-macosx32.dylib	15-Mar-2013 11:57	4.8M	MacOS X 32 bit library
libptc-macosx64.a	15-Mar-2013 11:57	6.1M	MacOS X 64 bit library
libptc-macosx64.dylib	15-Mar-2013 11:57	5.7M	MacOS X 64 bit library
madx-info.txt	15-Mar-2013 11:57	73	SVN release information
madx-linux32	15-Mar-2013 11:58	11M	Linux 32 bit binary
madx-linux64	15-Mar-2013 11:58	13M	Linux 64 bit binary
madx-macosx32	15-Mar-2013 11:57	8.4M	MacOS X 32 bit binary
madx-macosx64	15-Mar-2013 11:57	9.7M	MacOS X 64 bit binary
madx-src.tgz	15-Mar-2013 11:57	14M	Source code tarball
madx-win32.exe	15-Mar-2013 11:57	6.9M	Windows 32 bit binary
madx-win64.exe	15-Mar-2013 11:57	8.6M	Windows 64 bit binary
numdiff-linux32	15-Mar-2013 11:58	164K	Linux 32 bit binary
numdiff-linux64	15-Mar-2013 11:58	187K	Linux 64 bit binary
numdiff-macosx32	15-Mar-2013 11:57	172K	MacOS X 32 bit binary
numdiff-macosx64	15-Mar-2013 11:57	205K	MacOS X 64 bit binary
numdiff-win32.exe	15-Mar-2013 11:57	244K	Windows 32 bit binary
numdiff-win64.exe	15-Mar-2013 11:57	282K	Windows 64 bit binary

NOTE: You may need to make it executable

```
mv madx-macosx64 madx
chmod u+x madx
./madx
```

How to run MADX

- In command prompt:
- Go to directory (with madx.exe and input files)

```
>> madx.exe < inputfile > outputfile
Or on Mac OSX in terminal:
>> ./madx < inputfile < outputfile
```

- Or you can add the madx.exe location to your path (if you know how…)

An input file – Basic FODO

```
//
// MADX Example 1: FODO cell
// Author: V. Ziemann, Uppsala University
// Date: 060910
// UPDATED SUZIE SHEEHY 04/11/2013 FOR MADX VERSION 5.01.00

TITLE,'Example 1: FODO.MADX';

BEAM, PARTICLE=ELECTRON,PC=3.0;

D: DRIFT,L=1.0;
QF: QUADRUPOLE,L=0.5,K1=0.2;
QD: QUADRUPOLE,L=0.5,K1=-0.2;

FODO: LINE=(QF,5*(D),QD,QD,5*(D),QF);

SETPLOT, POST=2, FONT=-1;

USE, PERIOD=FODO;
TWISS,SAVE,BETX=15.0,BETY=5.0;
PLOT,HAXIS=S, VAXIS=BETX, BETY, NOVERSION=TRUE, TITLE='unmatched beta functions';

//Here MATCH is used as a single command this finds periodic solution
USE, PERIOD=FODO;
MATCH, SEQUENCE=FODO;
TWISS, SAVE;
PLOT,HAXIS=S, VAXIS=BETX, BETY, NOVERSION=TRUE, TITLE='matched beta functions';

Value, TABLE(SUMM,Q1);
Value, TABLE(SUMM,Q2);
WRITE,TABLE=SUMM,FILE=print.dat;
```

// comments out a line

TITLE at top of output

Define particle type and momentum (pc) GeV/c
Or can use ENERGY in GeV.

Elements are given in the manual including definitions of L, K1 etc…

Define a 'line' –can be a cell or a whole beamline that you will USE.

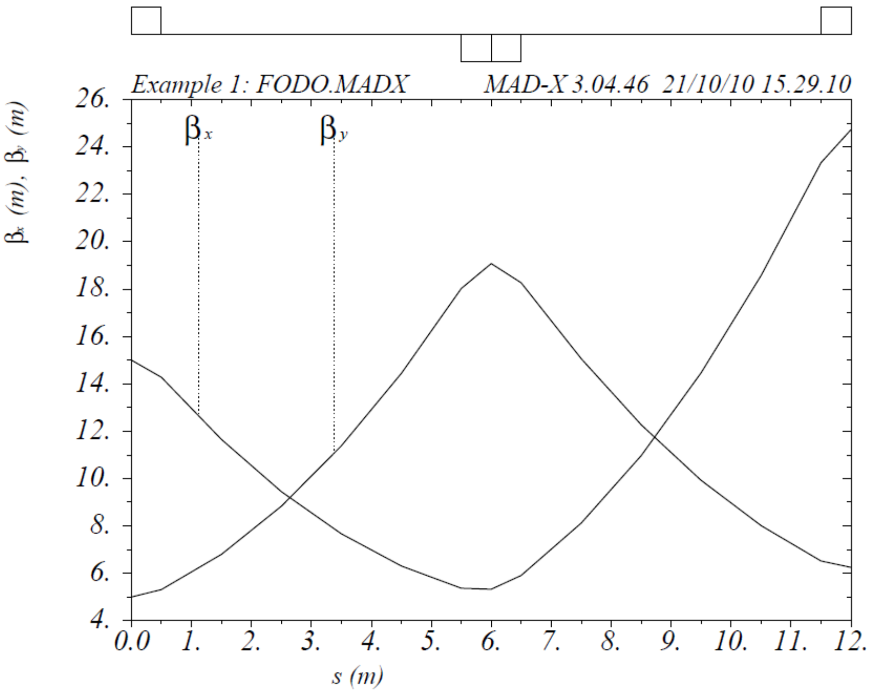
Calculate beta functions from starting values

Plot beta values from TWISS (internal table)

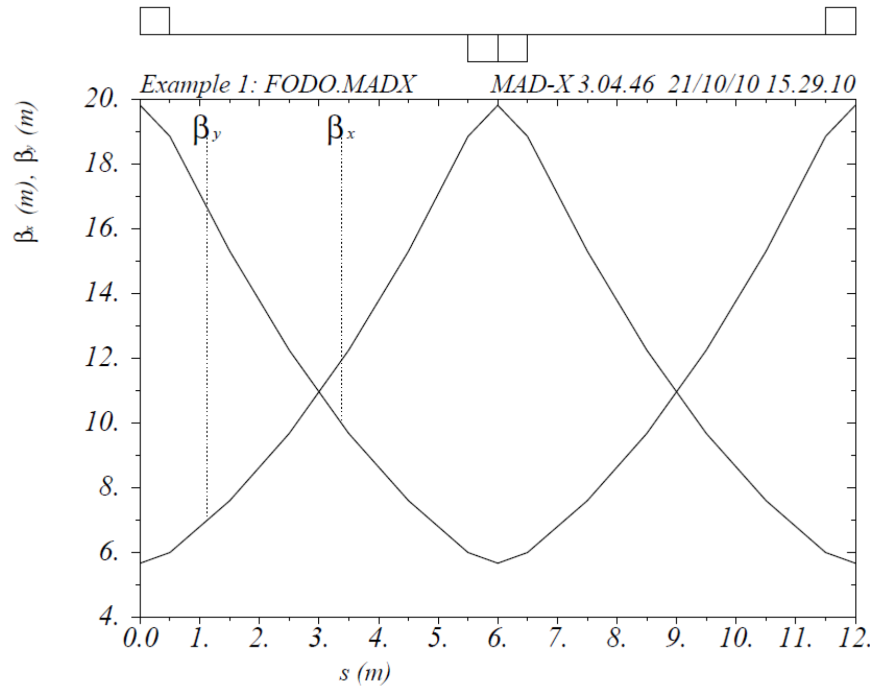
Match the periodic solution (+Plot that)

Output to tables

Result of a MADX run



With βx starting at 15m (βy at 5m)

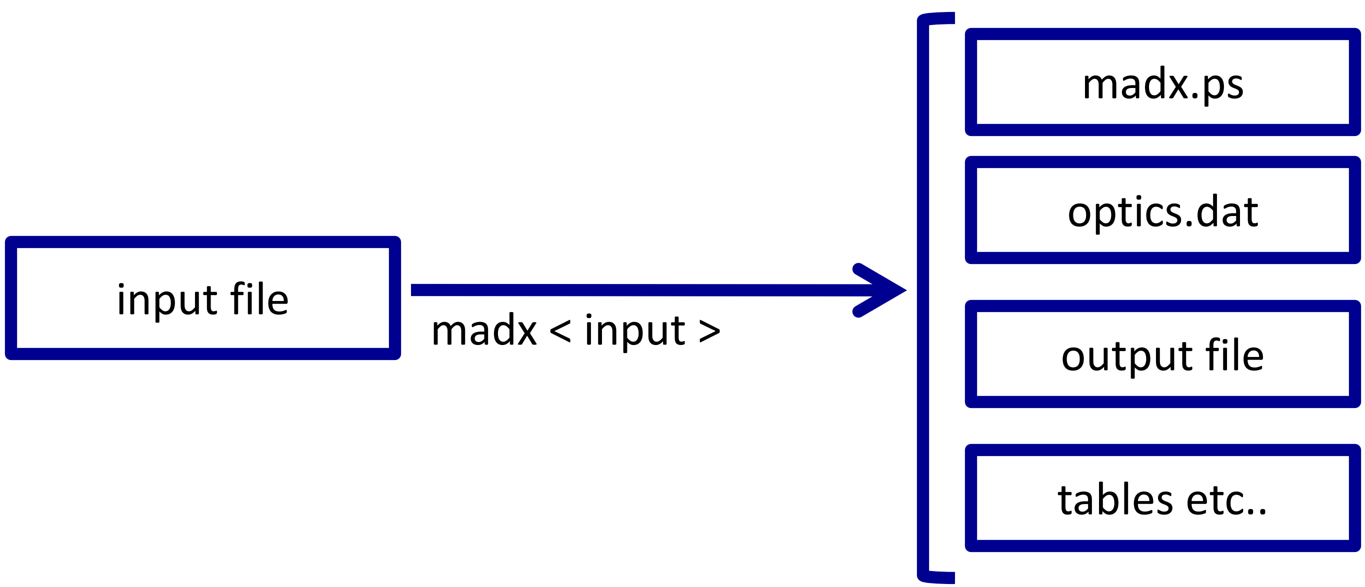


If the line is matched (periodic)

$\beta_{start} = \beta_{end}$

Output files

- Optics.dat
- Your specified output file madx< input > output
- Can specify tables

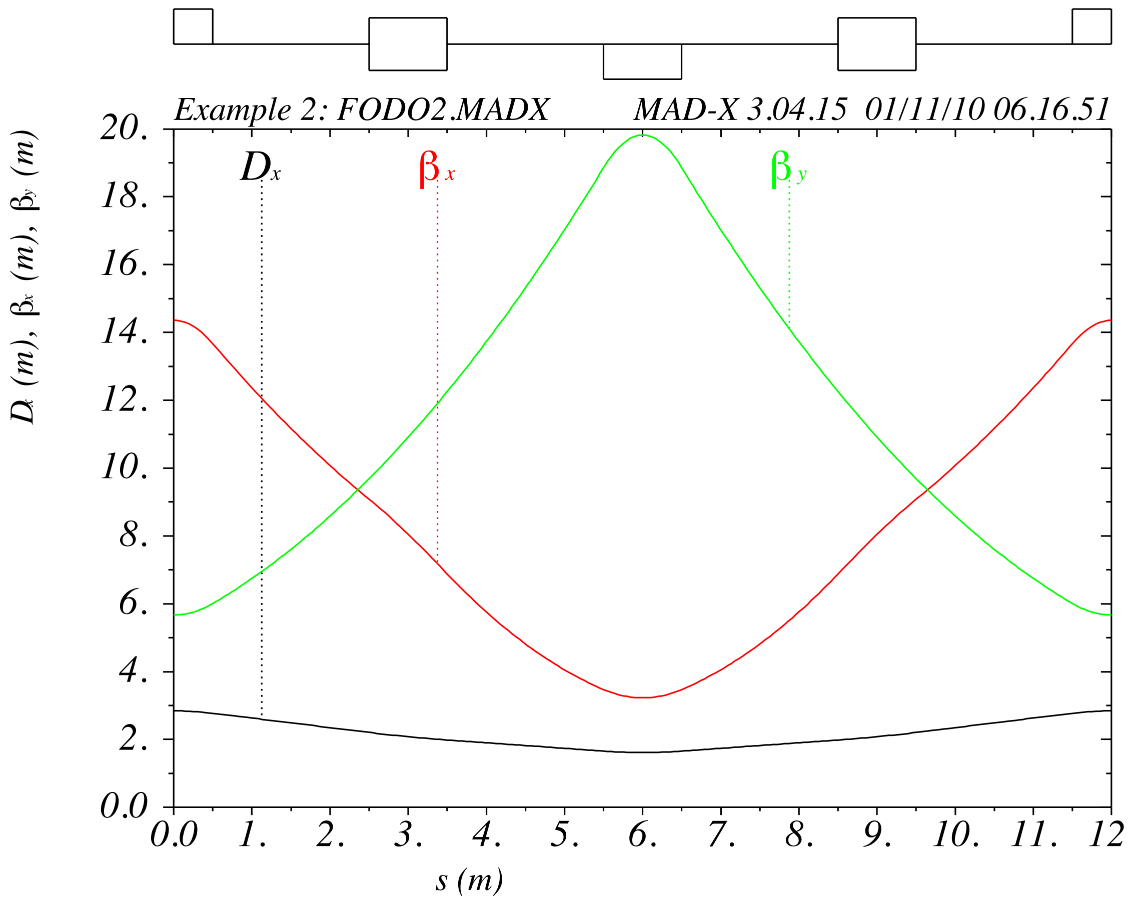


Add bending magnets

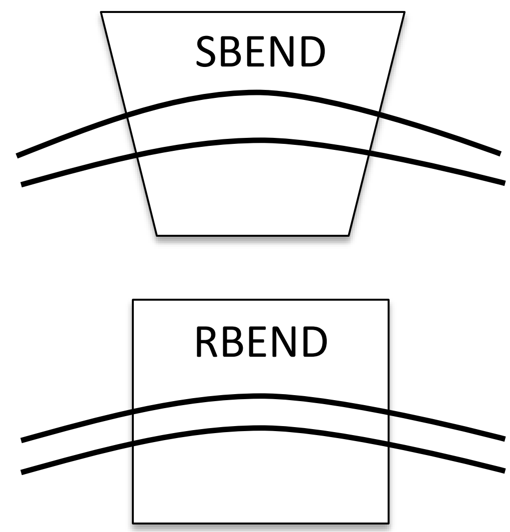
- Can introduce your own parameters (watch the :=)
- Can use alternative ‘sequence’ format
- Let’s add dipoles
- Look at dispersion

```
TITLE,'Example 2: FODO2.MADX';
BEAM, PARTICLE=ELECTRON,PC=3.0;
DEGREE:=PI/180.0;                // for readability

QF: QUADRUPOLE,L=0.5,K1=0.2;       // still half-length
QD: QUADRUPOLE,L=1.0,K1=-0.2;      // changed to full length
B: SBEND,L=1.0,ANGLE=15.0*DEGREE;  // added dipole

FODO: SEQUENCE,REFER=ENTRY,L=12.0;
  QF1:   QF,     AT=0.0;
  B1:    B,      AT=2.5;
  QD1:   QD,     AT=5.5;
  B2:    B,      AT=8.5;
  QF2:   QF,     AT=11.5;
ENDSEQUENCE;

USE, PERIOD=FODO;
//MATCH, SEQUENCE=FODO;     //Uncomment to match
SELECT,FLAG=SECTORMAP,clear;
SELECT,FLAG=TWISS,column=name,s,betx,bety;
TWISS, file=optics.dat,sectormap;

PLOT,HAXIS=S, COLOUR=100, VAXIS=DX, BETX, BETY,
INTERPOLATE=TRUE;

Value, TABLE(SUMM,Q1);
Value, TABLE(SUMM,Q2);
```

Output from FODO with dipole



Note the colours!
We now have dispersion (from bending magnet)
Focusing changed as we used SBEND



- If you add

```
SELECT,FLAG=SECTOR
SELECT,FLAG=TWISS,
TWISS, file=optics
```

- You will ge matrices
- and an out

You can customise

```
select,flag=my_sec
```

Or even select by comp

```
select,flag=my_sec
```

optics.dat

```
@ NAME                %05s "TWISS"
@ TYPE                %05s "TWISS"
@ SEQUENCE            %04s "FODO"
@ PARTICLE            %08s "ELECTRON"
@ MASS                %le       0.000510998902
@ CHARGE              %le                   -1
@ ENERGY              %le          3.000000044
@ PC                  %le                    3
@ GAMMA               %le          5870.854187
@ KBUNCH              %le                    1
@ BCURRENT            %le                    0
@ SIGE                %le                    0
@ SIGT                %le                    0
@ NPART               %le                    0
@ EX                  %le                    1
@ EY                  %le                    1
@ ET                  %le                    1
@ LENGTH              %le                   12
@ ALFA                %le         0.0910331259
@ ORBIT5              %le                   -0
@ GAMMATR             %le          3.314364527
@ Q1                  %le         0.2909501025
@ Q2                  %le         0.1913459932
@ DQ1                 %le        -0.2152485772
@ DQ2                 %le        -0.2276020063
@ DXMAX               %le           2.84272104
@ DYMAX               %le                    0
@ XCOMAX              %le                    0
@ YCOMAX              %le                    0
@ BETXMAX             %le          14.36014477
@ BETYMAX             %le          18.85637615
@ XCORMS              %le                    0
@ YCORMS              %le                    0
@ DXRMS               %le          2.388159047
@ DYRMS               %le                    0
@ DELTAP              %le                    0
@ SYNCH_1             %le                    0
@ SYNCH_2             %le                    0
@ SYNCH_3             %le                    0
@ SYNCH_4             %le                    0
@ SYNCH_5             %le                    0
@ TITLE               %21s "Example 2: FODO2.MADX"
@ ORIGIN              %20s "MAD-X 3.04.15 Darwin"
@ DATE                %08s "01/11/10"
@ TIME                %08s "08.52.11"
* NAME                             S                 BETX                 BETY
$ %s                             %le                  %le                  %le
 "FODO$START"                      0          14.36014477          5.674619595
 "QF1"                           0.5          13.67114589          6.007906055
 "DRIFT_0"                       2.5          9.081076899          9.690010744
 "B1"                            3.5          6.880079877          12.25970476
 "DRIFT_0"                       5.5          3.471640314          18.85637615
 "QD1"                           6.5          3.471640314          18.85637615
 "DRIFT_0"                       8.5          6.880079877          12.25970476
 "B2"                            9.5          9.081076899          9.690010744
```

Matching

- Matching lets MAD-X do the tedious work for you!
- Before MATCH select at least one sequence (USE)
- Initiated by the MATCH command
- Initiating:

```
MATCH, SEQUENCE='name1', 'name2',..,'nameX';
```

- Can define constraints & variables (magnets) to achieve aim

```
MATCH, SEQUENCE = FODO;
CONSTRAINT,SEQUENCE=FODO, RANGE=#E, MUX=0.1666666, MUY=0.25;
VARY, NAME=QF->K1, STEP=1E-6;
VARY, NAME=QD->K1, STEP=1E-6;
LMDIF,CALLS=500,TOLERANCE=1E-20;
ENDMATCH;
```

Matching input file

```
TITLE,'Example 3: MATCH1.MADX';
BEAM, PARTICLE=ELECTRON,PC=3.0;

D: DRIFT,L=1.0;
QF: QUADRUPOLE,L=0.5,K1:=0.2;
QD: QUADRUPOLE,L=0.5,K1:=-0.2;

FODO: LINE=(QF,5*(D),QD,QD,5*(D),QF);
USE, PERIOD=FODO;

//....match phase advance at end of cell to 60 and 90 degrees
MATCH, SEQUENCE=FODO;
CONSTRAINT,SEQUENCE=FODO,RANGE=#E,MUX=0.16666666,MUY=0.25;
VARY,NAME=QF->K1,STEP=1E-6;
VARY,NAME=QD->K1,STEP=1E-6;
LMDIF,CALLS=500,TOLERANCE=1E-20;
ENDMATCH;

SELECT,FLAG=SECTORMAP,clear;
SELECT,FLAG=TWISS,column=name,s,betx,alfx,bety,alfy;
TWISS, file=optics.dat,sectormap;

PLOT,HAXIS=S, VAXIS=BETX, BETY;
Value, TABLE(SUMM,Q1);  // verify result
Value, TABLE(SUMM,Q2);
```

Matching commands

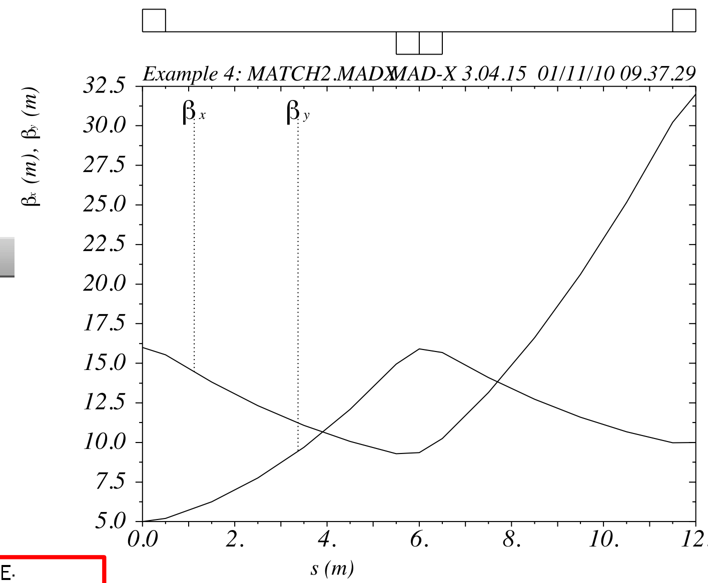
Print out final values of matching

Matching example

- Demonstration MATCH1.MADX

Fitting beta functions

- Use MATCH2.MADX

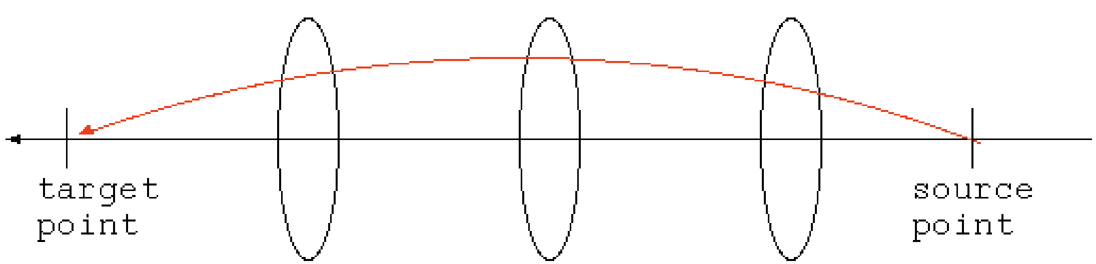


match2.out

```
Initial Penalty Function =   0.87329908E+02


call:       5   Penalty function =   0.14687632E+02
call:       8   Penalty function =   0.52122712E+00
call:      11   Penalty function =   0.14867643E-03
call:      14   Penalty function =   0.17048760E-10
call:      17   Penalty function =   0.86820304E-19
call:      20   Penalty function =   0.66895405E-27
 ++++++++++ LMDIF ended: converged successfully
call:      20   Penalty function =   0.66895405E-27
fodo$end:1                 betx         4      3.20000000E+01     3.20000000E+01     5.04870979E-
fodo$end:1                 bety         4      1.00000000E+01     1.00000000E+01     6.18466950E-20
qf->k1                     1.21494427E-01     -1.00000000E+20     1.00000000E+20
qd->k1                    -1.58047975E-01     -1.00000000E+20     1.00000000E+20

 GXPLOT-X11  1.50 initialized

 plot number =  1

  ++++++++++++++++++++++++++++++++++++++++++++
  +                MAD-X 3.04.15             +
  + Code Modification Date: 15.01.2008       +
  + Execution Time Stamp: 01.11.10 09.39.37 +
  ++++++++++++++++++++++++++++++++++++++++++++
//

// MADX Example 1: FODO matching final beta function
```

Transfer matrix matching

- Sometimes want to constrain transfer matrix elements to some value.
- For example $R_{16}=0$ and $R_{26}=0$ will make the horizontal position and angle independent of the momentum after a beamline.
- This is called an 'Achromat'.
- Other versions are imaginable
- point-to-point imaging→ R12 = 0.
  - This means sin(μ)=0 or a phase advance of a multiple of π.



Examples in MAD-X

- FODO arcs
- Dispersion suppressor
- ‘Telescopes’ for low-β
- Synchrotron radiation lattices + achromats

Is that it?

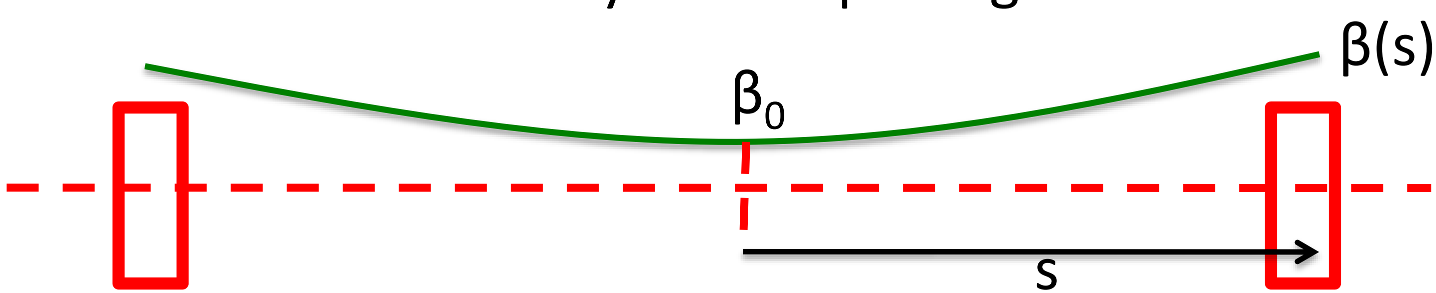
- 'the not-so-ideal world'
- What happens to α,β,γ if we stop focusing for a distance?

$$\beta(s) = \beta_0 - 2\alpha_0 s + \gamma_0 s^2$$
$$\alpha(s) = \alpha_0 - \gamma_0 s$$
$$\gamma(s) = \gamma_0$$

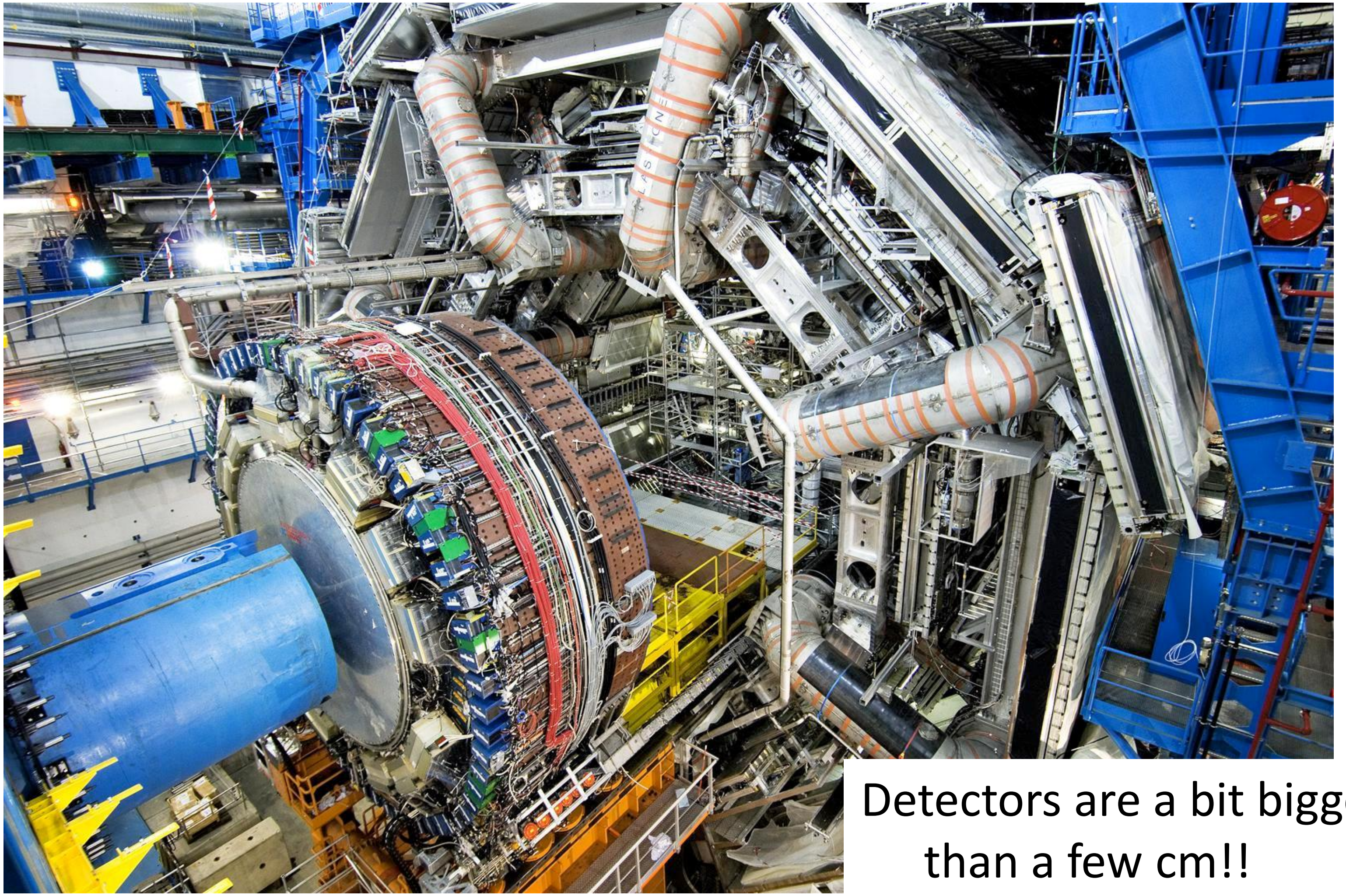
The drift length:

$$M = \begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix}$$

- If we take the center of a drift (α_0=0), we find $\beta(s) = \beta_0 + \frac{s^2}{\beta_0}$
- It doesn't matter what you do – β will grow!



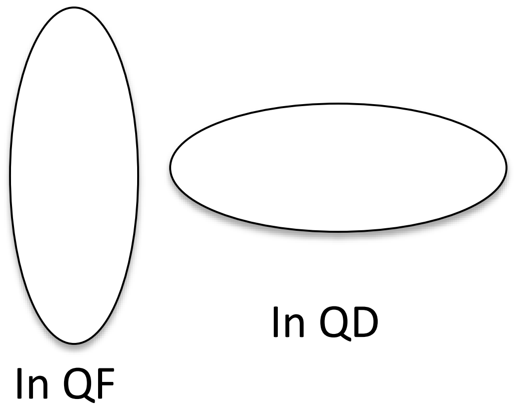
Seems fine, until...



Detectors are a bit bigger than a few cm!!

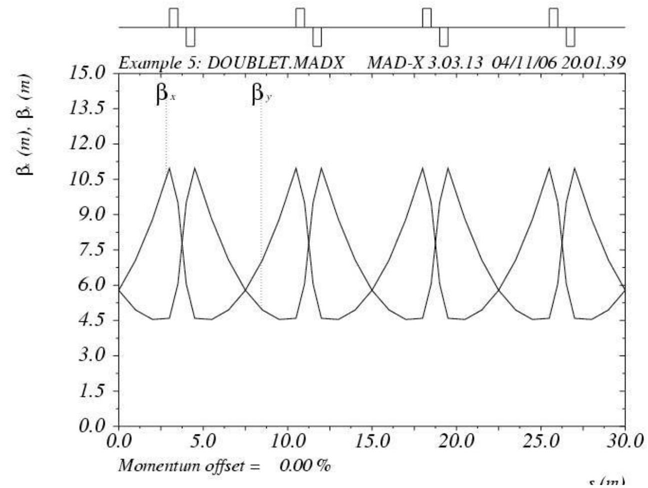
FODO Arcs

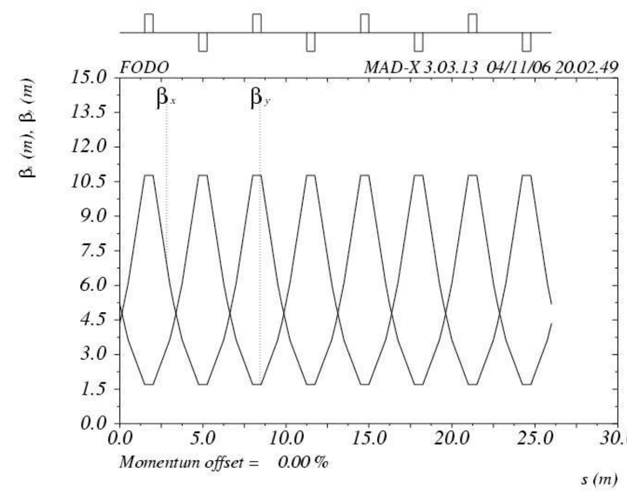
- Usually in colliders – take beam between interaction regions
- Simple and tunable (β_x large in QF, β_y at QD)
- Moderate quad strengths
- The beam is not round
- In arcs dipoles generate dispersion



FD Doublet Lattice

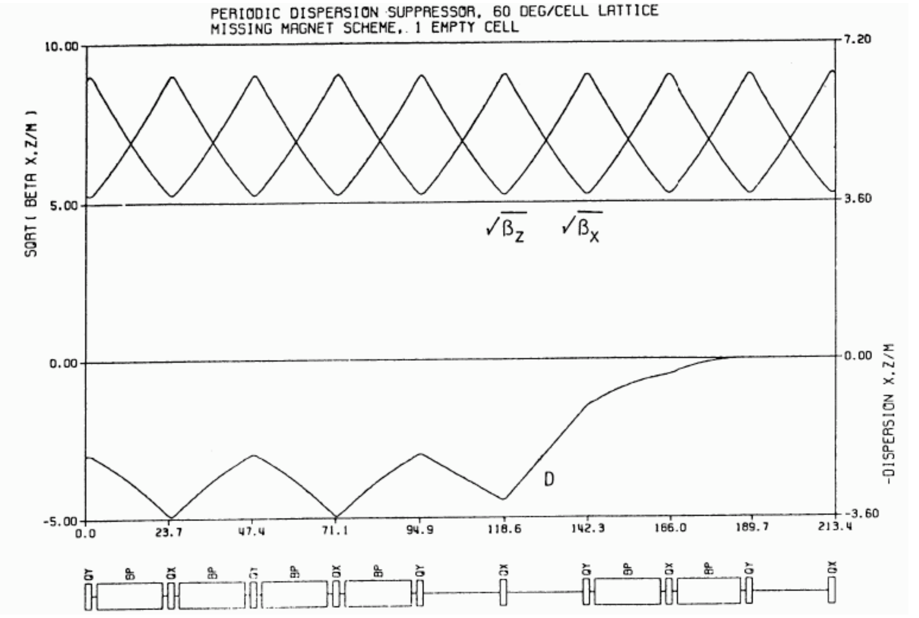
- More space between quads
- Stronger quad strengths
- Round beams
- Used in CTF3 linac





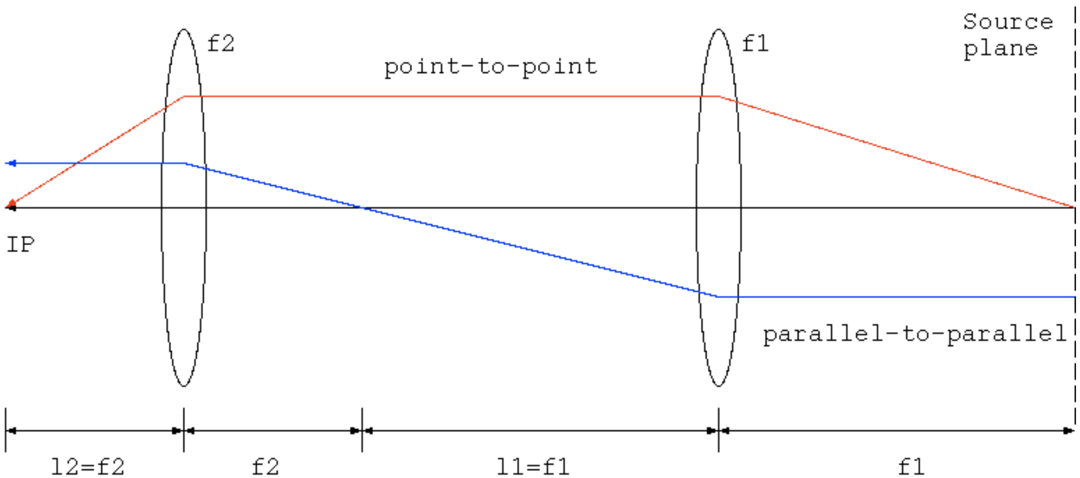
Dispersion suppressor

- Want small spot size at interaction point
- Spot size: $\sqrt{\varepsilon\beta + (D\Delta p / p)^2}$
- Missing magnet dispersion suppression scheme
- Works with proper phase advance between elements



Bring D from FODO cell value to zero by a forced oscillation around ~½ of that in the FODO cells

Telescope and low β



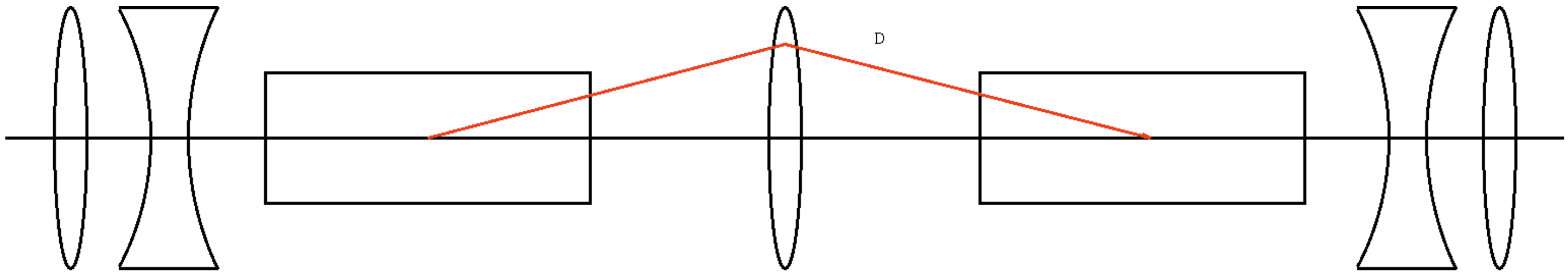
$$\begin{pmatrix} 1 & l_1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -1/f_1 & 1 \end{pmatrix} \begin{pmatrix} 1 & l_1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 - l_1/f_1 & 2l_1 - l_1^2/f_1 \\ -1/f_1 & 1 - l_1/f_1 \end{pmatrix}$$

For one module with l_1=f_1 $\begin{pmatrix} 0 & f_1 \\ -1/f_1 & 0 \end{pmatrix}$

For both modules: $R = \begin{pmatrix} f_2/f_1 & 0 \\ 0 & -f_1/f_2 \end{pmatrix}$

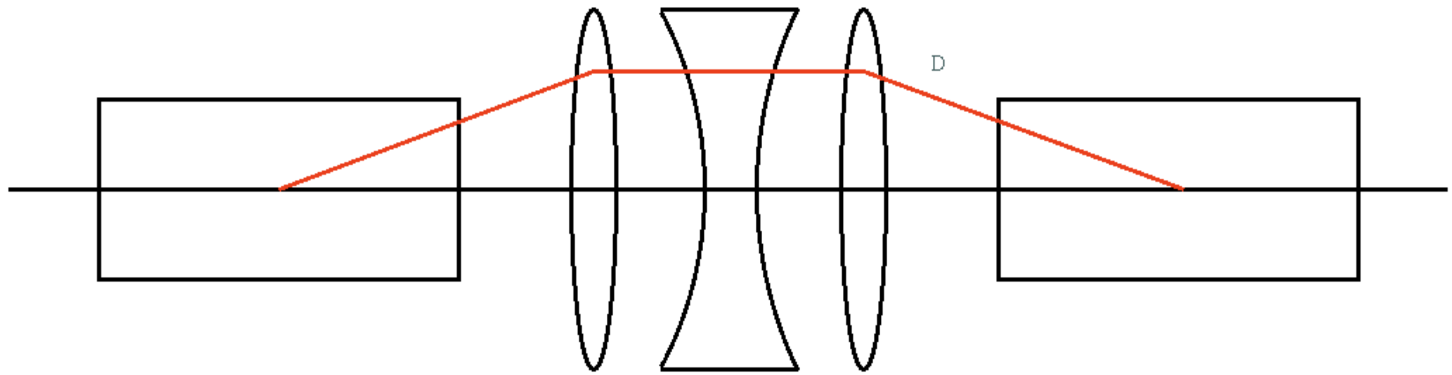
- Used in colliders to achieve small beam size at IP
- Doublet or Triplet
- Want:
  - Point-to-point R_{12}=0
  - Parallel to parallel R_{21}=0
  - R_{11}=demagnification
- Ratio of focal lengths
- Needs to work in both planes with doublets/triplets

Double bend achromat



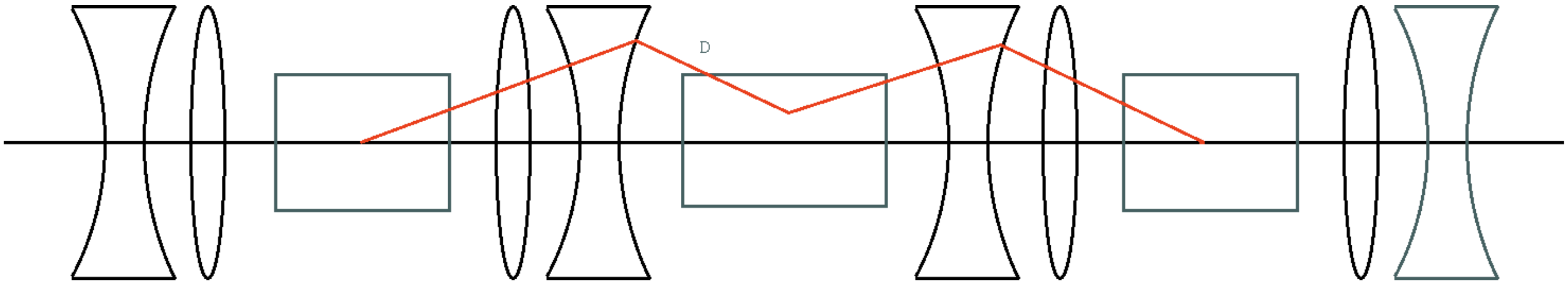
- One dipole generates dispersion and the next, which is 180 degrees apart will take it out again
- Remember: the dispersion is the orbit of a particle with slightly too high momentum w.r.t the reference particle
- Quadrupoles are used to make βx in dipoles small

Triplet achromat



- Do the 180 degrees in the horizontal plane and the beta matching by quads between dipoles
- very compact, few magnets, but not flexible

Triple bend achromat



- Small emittance.
- Very flexible due to large number of quadrupoles.
- Adjacent drift space can be made long to accommodate undulators/wigglers.

Resources

- Many examples available at the MAD-X website
  - A helpful 'primer' by W. Herr: http://madx.web.cern.ch/madx/doc/madx_primer.pdf

You can always ask me or another lecturer – though we can't promise to know the answer!