MAD for pedestrian

Werner Herr, CERN, AB Division

For all MAD details:

(http://frs.home.cern.ch/frs/Xdoc/mad-X.html)

Where you find all that:

Presentation: /afs/ifh.de/user/w/wfherr/public/mad_tudorial.ps

Examples in: /afs/ifh.de/user/w/wfherr/public/sps

Executable: /afs/ifh.de/user/w/wfherr/public/bin/madx

Source code in: /afs/ifh.de/user/w/wfherr/public/madX

For all MAD details:

(http://frs.home.cern.ch/frs/Xdoc/mad-X.html)

Why MAD ?

(.. or another optics program ?)

- Calculate optics parameters from machine description
- Compute (match) desired quantities
- Simulate and correct machine imperfections
- Simulate beam dynamics
- For circular machines or linacs

What is needed ?

(for any optics program)

- Definition of an element (what is it ?)
- Definition of the position of an element (where is it ?)
- Definition of strengths of an active element (how strong is it ?)
- Directives: MAD commands (what to do with it ?)

Definitions of elements

- Define a CLASS of elements, i.e. all elements with the same properties, (e.g. all bending magnets with 10.0 m length, all elements with the same strength ...), are used in later commands and declarations
- Keywords used to define the type of element. Some examples:

Long dipole (bending) magnet:

MBL: RBEND, L=10.0;

Short dipole (bending) magnet:

MBS: RBEND, L=2.50;

Quadrupole magnet:

MQ: QUADRUPOLE, L=3.3;

Sextupole magnet:

MSF: SEXTUPOLE, L=1.1;

Multipole:

MPM: MULTIPOLE, L=0.000001;

Corrector magnet:

LMC = 0.1;
MCV: VKICKER, L=LMC;
MCH: HKICKER, L=LMC;

Position monitor:

LBPM = 0.2;
BPM: MONITOR, L=LBPM;

Definitions of strengths

Dipole (bending) magnet:

$$k_0 \;=\; \frac{1}{p/c} B_y [\; in \quad T \;] \left[\;=\; \frac{1}{\rho} \;=\; \frac{angle}{l} \right] [\; in \quad rad/m \;]$$

DIP01: RBEND, L=10.0, ANGLE=angle1, K0 = $k_{0,l}$;

DIP02: RBEND, L=2.5, ANGLE=angle2, K0 = $k_{0,s}$; or
DIP02: MBS, ANGLE=angle2, K0 = $k_{0,s}$;

Quadrupole magnet:

$$k_1 \;=\; \frac{1}{p/c} \frac{\delta B_y}{\delta x} [\; in \quad T/m \;] \left[\;=\; \frac{1}{l \cdot f} \right]$$

MQA: QUADRUPOLE, L=3.3, K1 = k_1;

Sextupole magnet:

$$k_2 \;=\; \frac{1}{p/c}\frac{\delta^2 B_y}{\delta^2 x}\,[\;in \quad T/m^2\;]$$

MSXF: SEXTUPOLE, L=1.1, K2 = KLSF;
KLSF = k_2;

Corrector magnet:

The strength of a corrector is an angle (kick) [*in* *rad*]

MCV01: KICKER, L=0.1, KICK := KCV01;
MCV02: KICKER, L=0.1, KICK := KCV02;
MCV03: MCV, KICK := KCV03;

Special case: multipoles

Multipoles: general elements of zero length (thin lens), can be used with one or more components of any order:

MULTIP: MPM, LRAD =0.001,
KNL={k$_{n0}$L, k$_{n1}$L, k$_{n2}$L, k$_{n3}$L,....},
KSL={k$_{s0}$L, k$_{s1}$L, k$_{s2}$L, k$_{s3}$L,....};

$\rightarrow$ KNL = $k_n \cdot L$ (normal components)
$\rightarrow$ KSL = $k_s \cdot L$ (skew components)
Very simple to use:

MUL1: MPM, KNL = {0,k$_1$L,0,0,....};
is equivalent to definition of quadrupole

MUL0: MPM, KNL = {angle,0,0,....};
is equivalent to definition of a bending magnet

Thick and thin lenses

- Thin elements
  - Easy to use
  - Uses (amplitude dependent) kicks → always symplectic
  - Used for tracking
  - Path lengths not correctly described
  - Fringe fields not correctly described
- Thick elements
  - More precise, path length and fringe field correct
  - Not symplectic in tracking
  - May need symplectic integration

Conventions

- Elements are placed along the reference orbit (variable s)
- Horizontal (assumed bending plane) and vertical variables are x and y
- Describes a local coordinate system moving along s
- i.e. x = y = 0 follows the curvilinear system
- Comment lines start with: // or !

More conventions

- All commands and declarations end with a ;
- Variables can be used in expressions:
  - ANGLE = 2π/NBEND;
  - AIP = ATAN(SX1/SX2);
  - DX := GAUSS()*1.5E-3;
- The assignment symbols = and := have a different behaviour !
  - DX = GAUSS()*1.5E-3;
    The value is computed once and kept in DX
  - DX := GAUSS()*1.5E-3;
    The value is recomputed every time DX is used

Definitions of sequence (position)

Our elements have not yet a position.
A position can be defined at CENTRE or EXIT or ENTRY of an element .
Defined as absolute or relative position:

```
cassps: SEQUENCE, refer=centre, l=6912;
...
MBL01:  MBLA,   at = 102.7484;
MBL02:  MBLB,   at = 112.7484;
MQ01:  MQA,   at = 119.3984;
BPM01:  BPM,   at = 1.75, from MQ01;
COR01:  MCV01,   at = LMCV/2 + LBPM/2,
                 from BPM01;
MBL03:  MBLA,   at = 126.3484;
MBL04:  MBLB,   at = 136.3484;
MQ02:  MQB,   at = 142.9984;
BPM02:  BPM,   at = 1.75, from MQ02;
COR02:  MCV02,   at = LMCV/2 + LBPM/2,
                 from BPM02;
...
ENDSEQUENCE;
```

MAD definition: SPS

```
circum = 6912;
ncell = 108;
lcell = 64;
lquad = 3.085;
lquad2 = 1.5425;
lmb = 6.26;
lmb2 = 3.13;
!
// define bending magnet as multipole
mbsps: multipole,lrad:=0,knl:={0.00727220521664304};
!
qsps: quadrupole,l:=3.085;
qf: qsps,k1:=kqf;
qd: qsps,k1:=kqd;
kqf = 0.0146314747222255;
kqd = -0.0146434433207234;
!
lsf: sextupole,l:=lsex,k2:=ksf;
lsd: sextupole,l:=lsex,k2:=ksd;
ksf = 0.0202844420470744;
ksd = -0.0383942672294916;
!
// Monitors and correctors for orbit correction
bpm: monitor,l:=0.1;
ch: hkicker,l:=0.1;
cv: vkicker,l:=0.1;
!
cassps: sequence, l = 6912;
start_machine: marker, at = 0;
qf, at = 1.5425;
lsf, at = 4.0425;
ch, at = 4.6425;
bpm, at = 4.7425;
mbsps, at = 5.0425;
mbsps, at = 11.4425;
mbsps, at = 23.6425;
mbsps, at = 30.0425;
qd, at = 33.5425;
lsd, at = 36.0425;
cv, at = 36.6425;
```

```
bpm, at = 36.7425;
mbsps, at = 37.0425;
mbsps, at = 43.4425;
mbsps, at = 55.6425;
mbsps, at = 62.0425;
qf, at = 65.5425;
lsf, at = 68.0425;
ch, at = 68.6425;
bpm, at = 68.7425;
mbsps, at = 69.0425;
mbsps, at = 75.4425;
mbsps, at = 87.6425;
mbsps, at = 94.0425;
qd, at = 97.5425;
lsd, at = 100.0425;
cv, at = 100.6425;
.....
.....
.....
.....
cv, at = 6756.6425;
bpm, at = 6756.7425;
mbsps, at = 6757.0425;
mbsps, at = 6763.4425;
mbsps, at = 6775.6425;
mbsps, at = 6782.0425;
qf, at = 6785.5425;
lsf, at = 6788.0425;
ch, at = 6788.6425;
bpm, at = 6788.7425;
mbsps, at = 6789.0425;
mbsps, at = 6795.4425;
mbsps, at = 6807.6425;
mbsps, at = 6814.0425;
qd, at = 6817.5425;
lsd, at = 6820.0425;
cv, at = 6820.6425;
bpm, at = 6820.7425;
mbsps, at = 6821.0425;
mbsps, at = 6827.4425;
mbsps, at = 6839.6425;
mbsps, at = 6846.0425;
qf, at = 6849.5425;
lsf, at = 6852.0425;
```

```
ch, at = 6852.6425;
bpm, at = 6852.7425;
mbsps, at = 6853.0425;
mbsps, at = 6859.4425;
mbsps, at = 6871.6425;
mbsps, at = 6878.0425;
qd, at = 6881.5425;
lsd, at = 6884.0425;
cv, at = 6884.6425;
bpm, at = 6884.7425;
mbsps, at = 6885.0425;
mbsps, at = 6891.4425;
mbsps, at = 6903.6425;
mbsps, at = 6910.0425;
end_machine: marker, at = 6912;
endsequence;
```

SPS: alternative

```
// define the total length
circum=6912.0;

// define number of cells and therefore cell length
ncell = 108;
lcell = circum/ncell;

// define lengths of elements and half lengths
lquad = 3.085;
lquad2 = lquad/2.;
lmb   = 6.260;
lmb2  = lmb/2.;
lsex  = 1.0;

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

// define bending magnet as multipole
// the sum of all bending magnets must give: 2*pi !!
mbsps: multipole, lrad=dummy, l=lmb, knl={2.0*pi/(8*ncell)};

// define quadrupole and their strengths
// sign defines focusing or defocusing quadrupole
qsps: quadrupole, l=lquad;
qf: qsps, k1:=kqf;
qd: qsps, k1:=kqd;
kqf =  1.46314747E-02;
kqd = -1.46434433E-02;

// define sextupoles for chromaticity correction
lsf: sextupole, l=lsex,k2:=ksf;
lsd: sextupole, l=lsex,k2:=ksd;
ksf =  2.02844420E-02;
ksd = -3.83942672E-02;

// define orbit correctors and beam position monitors
bpm: monitor, l=0.1;
ch:  hkicker, l=0.1;
cv:  vkicker, l=0.1;
```

```
// sequence declaration, its name is: cassps;
cassps: sequence, refer=centre, l=circum;
start_machine: marker, at = 0;

// This defines ONE cell, repeat NCELL times
// to get the full machine
// SPS has 8 bending magnets per cell
   n = 1;
   while (n < ncell+1) {
   qf: qf,         at=(n-1)*lcell+lquad2;
   lsf: lsf,       at=(n-1)*lcell+lquad2+2.5;
   ch:  ch,        at=(n-1)*lcell+lquad2+3.1;
   bpm: bpm,       at=(n-1)*lcell+lquad2+3.2;
   mbsps: mbsps,   at=(n-1)*lcell+lquad2+3.50;
   mbsps: mbsps,   at=(n-1)*lcell+lquad2+9.90;
   mbsps: mbsps,   at=(n-1)*lcell+lquad2+22.10;
   mbsps: mbsps,   at=(n-1)*lcell+lquad2+28.50;
   qd: qd,         at=(n-1)*lcell+lquad2+32.00;
   lsd: lsd,       at=(n-1)*lcell+lquad2+34.50;
   cv:  cv,        at=(n-1)*lcell+lquad2+35.10;
   bpm: bpm,       at=(n-1)*lcell+lquad2+35.20;
   mbsps: mbsps,   at=(n-1)*lcell+lquad2+35.50;
   mbsps: mbsps,   at=(n-1)*lcell+lquad2+41.90;
   mbsps: mbsps,   at=(n-1)*lcell+lquad2+54.10;
   mbsps: mbsps,   at=(n-1)*lcell+lquad2+60.50;

   n = n + 1;
}
end_machine: marker at=circum;
endsequence;
```

Simple MAD directives

- Define the input
- Define the beam
- Initiate computations (Twiss calculation, error assignment, orbit correction etc.)
- Output results (tables, plotting)
- Match desired parameters

Input definition and selection

- Define the input:
  - call,"sps.seq";
  - Select a file with description of machine
  - Can be split into several files
- Select the machine:
  - USE, sequence=cassps;
  - Select the sequence you want

We still need a beam !

Some computations need to know the type of beam and its properties:

- **Particle type**
- **Energy**
- **Emittance, number of bunches, intensity ....**

BEAM, PARTICLE=name, MASS=mass, NPART=Nb, CHARGE=q, ENERGY=E,.....;

BEAM, PARTICLE=proton, NPART=1.1E11, ENERGY=450,.....;

Definition of actions

- **Select an action (calculation of Twiss parameters around the machine) and its desired output (MAD variables):**

```
SELECT, flag=twiss,
column=name,s,x,y,mux,betx,muy,bety,dx,dy;
TWISS, sequence=cassps, centre,file=output;
```

- **Select an action and selective output:**

```
// only elements atrting with Q
SELECT, pattern="^Q.*",flag=twiss,
column=name,s,betx,bety;
TWISS, sequence=cassps, centre,file=beta.output;
```

MAD input:

```
TITLE, s='MAD-X test';

// Read input file with machine description
call file="sps.seq";
option,-echo;

// Define the beam for the machine
Beam, particle = proton, sequence=cassps, energy = 450.0,
          NPART=1.05E11, sige=      4.5e-4 ;

// Use the sequence with the name: cassps
use, sequence=cassps;

// Define the type and amount of output
select,flag=twiss,column=name,s,betx,bety;

// Execute the Twiss command to calculate the Twiss parameters
// Compute at the centre of the element and write to: twiss.out
twiss,save,centre,file=twiss.out;

// Plot the horizontal and vertical beta function between the
// 10th and 16th occurence of a defocussing quadrupole
plot, haxis=s, vaxis=x, betx, bety,colour=100,
      range=qd[10]/qd[16];
stop;
```

MAD output:

* NAME	S	BETX	BETY
$ %s	%le	%le	%le
"START_MACHINE"	0	103.8655172	20.28396999
"QF"	1.5425	107.5443191	19.4745051
"DRIFT_0"	3.31375	102.7903919	20.53014061
"LSF"	4.0425	99.40931301	21.35848368
"DRIFT_1"	4.5675	97.01509425	21.99682641
"CH"	4.6425	96.67590362	22.09086191
"BPM"	4.7425	96.22475412	22.21734845
"DRIFT_2"	4.9175	95.43828038	22.44174104
"MBSPS"	5.0425	94.87888061	22.6043912
"DRIFT_3"	8.2425	81.2298941	27.44060719
"MBSPS"	11.4425	68.87370328	33.57101279
"DRIFT_4"	17.5425	48.90078379	48.84202894
"MBSPS"	23.6425	33.6256111	68.81585713
"DRIFT_3"	26.8425	27.49099947	81.1745502
"MBSPS"	30.0425	22.64918352	94.82743288
"DRIFT_5"	31.02125	21.42644605	99.26175075
"QD"	33.5425	19.5134888	107.4973054
"DRIFT_0"	35.31375	20.57132771	102.741629
"LSD"	36.0425	21.40124015	99.3593687
"DRIFT_1"	36.5675	22.04066868	96.96434368
"CV"	36.6425	22.13485623	96.62504093
"BPM"	36.7425	22.26154431	96.17374313
"DRIFT_2"	36.9175	22.48628633	95.38701314
"MBSPS"	37.0425	22.64918352	94.82743288
"DRIFT_3"	40.2425	27.49099947	81.1745502
"MBSPS"	43.4425	33.6256111	68.81585713
"DRIFT_4"	49.5425	48.90078379	48.84202894
"MBSPS"	55.6425	68.87370328	33.57101279
"DRIFT_3"	58.8425	81.2298941	27.44060719
"MBSPS"	62.0425	94.87888061	22.6043912
"DRIFT_5"	63.02125	99.31172842	21.38364491
"QF"	65.5425	107.5443191	19.4745051

. . . . . .

. . . . . .

MAD output:

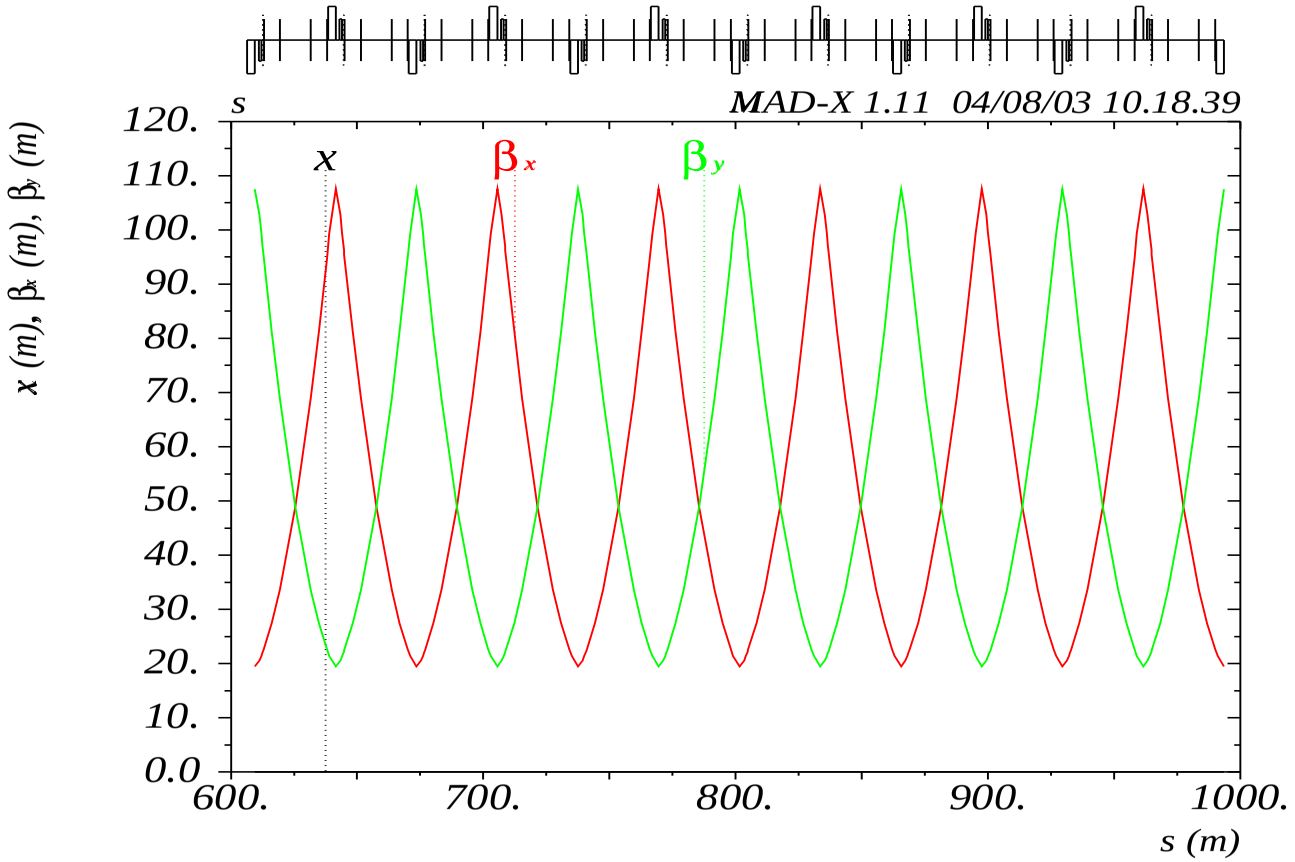
* NAME	S	BETX	BETY
$ %s	%le	%le	%le
"QF"	1.5425	107.5443191	19.4745051
"QD"	33.5425	19.5134888	107.4973054
"QF"	65.5425	107.5443191	19.4745051
"QD"	97.5425	19.5134888	107.4973054
"QF"	129.5425	107.5443191	19.4745051
"QD"	161.5425	19.5134888	107.4973054
"QF"	193.5425	107.5443191	19.4745051
"QD"	225.5425	19.5134888	107.4973054
"QF"	257.5425	107.5443191	19.4745051
"QD"	289.5425	19.5134888	107.4973054
"QF"	321.5425	107.5443191	19.4745051
"QD"	353.5425	19.5134888	107.4973054
"QF"	385.5425	107.5443191	19.4745051
"QD"	417.5425	19.5134888	107.4973054
"QF"	449.5425	107.5443191	19.4745051
"QD"	481.5425	19.5134888	107.4973054
"QF"	513.5425	107.5443191	19.4745051
"QD"	545.5425	19.5134888	107.4973054
"QF"	577.5425	107.5443191	19.4745051
"QD"	609.5425	19.5134888	107.4973054
"QF"	641.5425	107.5443191	19.4745051
"QD"	673.5425	19.5134888	107.4973054
"QF"	705.5425	107.5443191	19.4745051
"QD"	737.5425	19.5134888	107.4973054
"QF"	769.5425	107.5443191	19.4745051
"QD"	801.5425	19.5134888	107.4973054
"QF"	833.5425	107.5443191	19.4745051
"QD"	865.5425	19.5134888	107.4973054
"QF"	897.5425	107.5443191	19.4745051
"QD"	929.5425	19.5134888	107.4973054
"QF"	961.5425	107.5443191	19.4745051
"QD"	993.5425	19.5134888	107.4973054
"QF"	1025.5425	107.5443191	19.4745051

Graphical output

- **Plotting of Twiss functions**
- **Plot β_x and β_y as function of position s**
- **Between 10th and 16th quadrupole:**

```
plot, haxis=s, vaxis= betx, bety, colour=100,
      range=qd[10]/qd[16];
```

Graphical output



Matching global parameters

- Adjust strengths etc. to get desired properties (e.g. tune, chromaticity)
- Define the properties you want and the elements to vary

Example, match horizontal and vertical tunes:

```
use, sequence=cassps;
match, sequence=cassps;
    vary,name=kqf, step=0.00001;
    vary,name=kqd, step=0.00001;
    global,sequence=cassps,Q1=26.58;
    global,sequence=cassps,Q2=26.62;
    Lmdif, calls=10, tolerance=1.0e-21;
endmatch;

! check and plot the result of the matching
select,flag=twiss,column=name,s,mux,muy;
twiss,save,centre,file=twiss.out;
plot, haxis=s, vaxis=mux, muy,colour=100,
    range=qd[10]/qd[16];
```

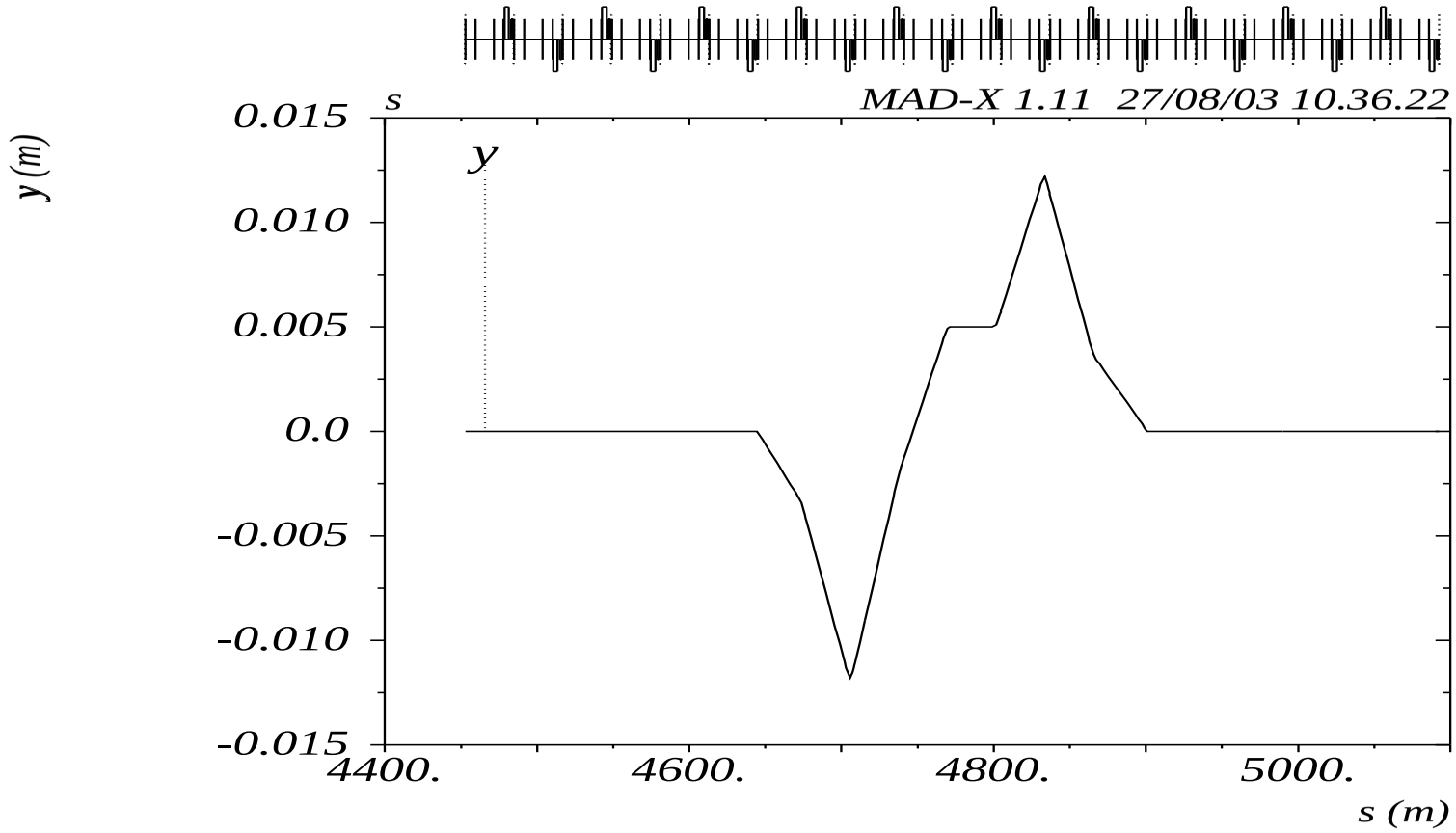
Matching in a range

- The matching can be restricted to a small range, keeping the rest untouched
- Boundary conditions at limits of the range

Example: local vertical orbit bump

```
match, sequence=cassps;
   vary,name=kcv73, step=0.00001;
   vary,name=kcv74, step=0.00001;
   vary,name=kcv76, step=0.00001;
   vary,name=kcv77, step=0.00001;
   constraint, range=bpmv71,y=0.0,py=0.0;
   constraint, range=bpmv75,y=0.005,py=0.0;
   constraint, range=bpmv79,y=0.0,py=0.0;
   Lmdif, calls=10, tolerance=1.0e-21;
endmatch;
```

Graphical output



Error assignment

- Can define alignment errors:

  ! assign error to all elements starting with Q

  select,flag=error,pattern="Q.*";

  ealign,dx:=tgauss(3.0)*1.0e-4,dy:=tgauss(3.0)*2.0e-4;

- Can define field errors of any order:

  ! assign error to all elements of class bed

  Select, flag=error, class=bed;

  Efcomp, radius:=0.017, order:=0,

  dkn:={7e-1,1e-2,4e-5,1e-5,1e-6,0,0},

  dks:={7e-1,2e-2,1e-4,6e-4,3e-6,0,0,0,0};

- Remember the := !

How to use MAD-X ?

- Interactively:
  - Type madx then input the commands on keybord
  - Type madx then call an input file: call,file=sps.mad;
- Batch mode:
  - Type madx < sps.mad;
- Our example:
  - sps.mad: MAD-X commands
  - sps.seq: machine description

What we do not need ...

- Particle tracking, dynamic aperture
- Higher order effects
- IBS, beam-beam elements
- Geometric survey
- Equilibrium emittance (leptons)
- RF and acceleration