

Lattice Design Lectures

JAI lectures - Michaelmas Term 2020

Hector Garcia Morales

University of Oxford/CERN

October 29, 2020



Outline

Introduction

Install and Run MADX

MADX input and output

Matching and Tracking

Some examples

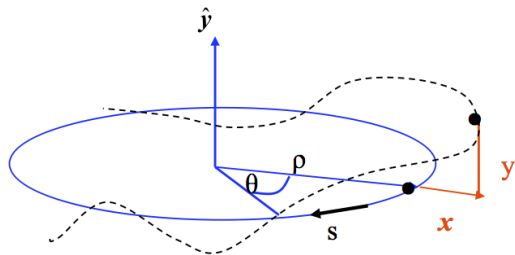
Introduction

- ▶ How to install and run MADX.
- ▶ How to use elements and sequences as inputs.
- ▶ How to extract information about Twiss functions, tunes, orbits...
- ▶ Matching, Tracking and Plotting.
- ▶ Some examples.

What is MADX?

MAD (Methodological Accelerator Design) is a program for accelerator design and simulation.

- ▶ Uses a sequence of elements placed sequentially along a reference orbit,
- ▶ The reference orbit is the path of a charged particle having the central design momentum of the accelerator through ideal magnets.
- ▶ The reference orbit consist of a series of straight line segments and circular arcs.
- ▶ Local curvilinear right handed coordinate system.



What is MADX useful for?

- ▶ Multiplatform.
- ▶ Very flexible and possible to extend.
- ▶ Made for complicated applications.
- ▶ Mainly designed for large projects (LEP, LHC, CLIC, FCC).

What is MADX not useful for?

- ▶ A program for teaching.
- ▶ Easy for beginners.
- ▶ Friendly graphical user interface.

Installing MADX

- ▶ MADX website: <https://mad.web.cern.ch/mad/>
- ▶ Go to releases and download the last version.
- ▶ MAD-NG: under development.

Run MADX

```
hector@hector-OptiPlex-9020: ~  
hector@hector-OptiPlex-9020: ~ 80x24  
-bash-4.2$ madx  
  
+++++  
+   MAD-X 5.06.01  (64 bit, Linux)   +  
+ Support: mad@cern.ch, http://cern.ch/mad +  
+ Release  date: 2020.09.01         +  
+ Execution date: 2020.10.08 14:18:35 +  
+++++  
  
X:> _
```


Run MADX

```
hector@hector-OptiPlex-9020: ~  
hector@hector-OptiPlex-9020: ~ 80x24  
-bash-4.2$ madx job.madx  
  
+++++  
+   MAD-X 5.06.01  (64 bit, Linux)   +  
+ Support: mad@cern.ch, http://cern.ch/mad +  
+ Release  date: 2020.09.01         +  
+ Execution date: 2020.10.08 14:19:17 +  
+++++  
  
!system,"mkdir result";  
  
! Sequence definition  
Option, -echo, -warn,-info;  
START MATCHING  
  
number of sequences: 1  
sequence name: lhcb1  
number of variables: 4  
user given constraints: 4
```

MADX syntax and input structure

```
3_FODO_match.madx
~/scratch0/JAI_lectures

Open [icon] Save [icon]

1 ! ## Definition of elements
2 ! Define two quadrupoles (note the deferred assignments).
3 qf_type: quadrupole, l=1.5, k1:=kf;
4 qd_type: quadrupole, l=1.5, k1:=kd;
5
6 ! ## Definition of the sequence
7 ! A short fodo of 10 m.
8 fodo:sequence, refer=exit, l=10;
9 qf: qf_type, at=5;
10 qd: qd_type, at=10;
11 endsequence;
12
13 ! ## Definition of the strength
14 kf=+0.25;
15 kd:=-kf;
16
17 ! ## Definition of the beam
18 beam, particle=proton, energy=7000;
19
20 ! ## Activation of the sequence
21 use, sequence=fodo;
22
23 ! ## Operations
24 ! A simple twiss and plot
25 select, flag=twiss, column=name,s,betx, bety, alfx,alfy;
```

```
22
23 ! ## Operations
24 ! A simple twiss and plot
25 select, flag=twiss, column=name,s,betx, bety, alfx,alfy;
26 twiss, file=before_matching.twiss;
27 plot, haxis=s, vaxis=betx, bety, colour=100, noversion=true, title='before matching';
28
29 ! ## Matching
30 match, sequence=fodo;
31 global, q1=.25;
32 global, q2=.25;
33 vary, name=kf, step=0.00001;
34 vary, name=kd, step=0.00001;
35 lmdif, calls=50, tolerance=1e-8;
36 endmatch;
37
38 ! ## Operations
39 twiss, file=after_matching.twiss;
40 plot, haxis=s, vaxis=betx, bety, colour=100, noversion=true, title='after matching', interpolate=tr
41
42 ! ## Output
43 value, table(summ,Q1);
44 value, table(twiss,qf, betx);
45
46 ! ## Conversion ps2pdf
47 ! This command assumes that in your system the command ps2pdf is available
48 system, 'ps2pdf madx.ps FODO_match.pdf';
49
50 ! ## Exit
51 quit;
```

MADX basic output

```
hector@hector-HP-EliteBook-840-G1: ~
hector@hector-HP-EliteBook-840-G1: ~ 107x35
select, flag=twiss, column=name,s,betx, bety, alfx,alfy;

twiss, file=FODO.twiss;

enter Twiss module

iteration: 1 error: 0.000000E+00 deltap: 0.000000E+00
orbit: 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00

+++++ table: summ

      length      orbit5      alfa      gammatr
      10          -0          0          0

      q1          dq1          betxmax      dxmax
0.3159191546    -0.4863193631    16.65487108    0

      dxrms      xcomax      xcorms      q2
      0          0          0          0.3159191546

      dq2      betymax      dymax      dyrms
-0.4863193631    16.65487108    0          0

      ycomax      ycorms      deltap      synch_1
      0          0          0          0

      synch_2      synch_3      synch_4      synch_5
      0          0          0          0

      synch_6      synch_8      nflips
      0          0          0

plot, haxis=s, vaxis=betx, bety, colour=100, noversion=true, title='FODO';

Plot - default table plotted: twiss
```

Matching

- ▶ Manually matching optics in MADX might be a hard task.
- ▶ MADX does it for you.
- ▶ How?
 - ▶ Select a sequence.
 - ▶ Initiate the matching routine.
 - ▶ Select variables and constraints.
 - ▶ Two types of constraints:
 - ▶ Global variables.
 - ▶ Local variables.

Tracking and Plotting

Tracking

- ▶ Particles can be tracked through the magnetic elements of your lattice.
- ▶ Two different ways:
 - ▶ MADX TRACK.
 - ▶ PTC (Polymorphic Tracking Code).

Plotting

- ▶ You can plot any output ($\beta, \alpha, x, y...$).

Some examples

Resources

- ▶ Many examples available at the MADX website.
- ▶ MADX Primer
- ▶ I uploaded the examples to the JAI lectures website.