MAD-X

Bruce Yee Rendón
byee@post.j-parc.jp

Accelerator Division
Japan Proton Accelerator Research Complex (J-PARC)
High Energy Accelerator Research Organization (KEK)

Luis Eduardo Medina Medrano
lmedinam@cern.ch

Beam Department
European Organization for Nuclear Research (CERN)
Universidad de Guanajuato (UG)

Acknowledgement to Werner Herr, CAS

Some formalities

Course scheme:

1. Lectures (Friday 13 and Saturday 14). Introduction to concepts. Exercises.
2. Work in the Exercise in group (Tuesday 17). Assignation of an exercise to each group. Creation of a presentation with the solution.
3. Presentation in group (Thursday 19). Each group will have 10 minutes to expose and questions.

Instructors:

Bruce Yee Rendon

Luis Eduardo Medina Medrano

Disclaimer

- This course is mostly based on **Werner Herr**’s CAS course.
- In some cases, Herr's slides may be used directly.
- This is an **introductory** course, thus, if you are interested in more information and details please go to the next link

http://zwe.web.cern.ch/zwe/

and/or contact him at werner.herr@cern.ch

Contents

I. **Introduction.** *Description of basic concepts and jargon.*

II. **MAD-X language.** *Syntaxis, variables.*

III. **Machine description.** *Magnets and sequences.*

IV. **MAD-X commands.** *Beam, twiss functions, geometry, plots.*

V. **Advanced commands.** *Global matching.*

VI. **What we don't have time for: realistic accelerators.** *Local matching, orbit errors and corrections, particle tracking.*

I. Introduction

- Purpose of an accelerator lattice software
- What is MAD-X?
- Why using MAD-X?
- Three basic components

Purpose of an accelerator lattice software

- Definition of **circular** or **linear** accelerators: machine definition.
- Calculation of its **optic parameters**.
- **Simulation** and **correction** of the accelerator imperfections.
- **Definition** and **matching** of desired properties.
- **Beam dynamics** simulations.
- Etc.

What is MAD-X?

- **MAD-X**: *Methodical Accelerator Design.*
- Latest version after a long development (MAD8, MAD9, MAD-X).
- Allows the **design** of accelerator lattices in order to simulate, **calculate** and **improve** its optic parameters.
- It has been used for more than 20 years in machine desing (PS, SPS, LEP, LHC) and in futures proyects (CLIC, FCC, etc.).
- Official website: madx.web.cern.ch

Why using MAD-X?

- **Multipurpouse**.
- Run in the different platforms (**Windows** and **Linux**).
- **Free** software.
- **Easy** to understand.
- The program is clear and **intuitive**.

Three basic components

When using MAD-X to design a lattice, you have three basic elements in the study:

- A **lattice design**: definition of the accelerator elements, their physical attributes and locations in the machine (sequences). *What is the machine in question?*
- A **beam** description: type of particle, energy, etc. *What will be running in that machine?*
- A series of **tasks** to be performed on that given machine, with that given particle beam. *What do you want to study about that machine?*

II. MAD-X language

- Language features
- Input sentences
- Conventions and Optic variables
- How to run MAD-X?

Language features I

- MAD-X is an **interpreter**:
  - It accepts and executes **statements**.
  - Statements can be commands, actions, declarations, etc.
  - It can be used in an **interactive** way or in **batch**.
- It makes use of many of the functions of a standard programming languange (**loops**, if, **macros**, subroutines, ...).
- Strong use of **C language**.
- **No case sensitive!**

Language features II

- **All** the sentences **end** with a semicolon (;).
- **Comments**:
  - **Start** with two slashes (//) or an exclamation mark (!) if they consist of a single line.
  - Are **enclosed** by (/*) and (*/) if they span over more of one line.
- Use arithmetics expresions (exp, log, sin, …).
- **Differed expressions** make use of := .
- Predefined **constants** (pi, e, mp, me, …).

Input sentences

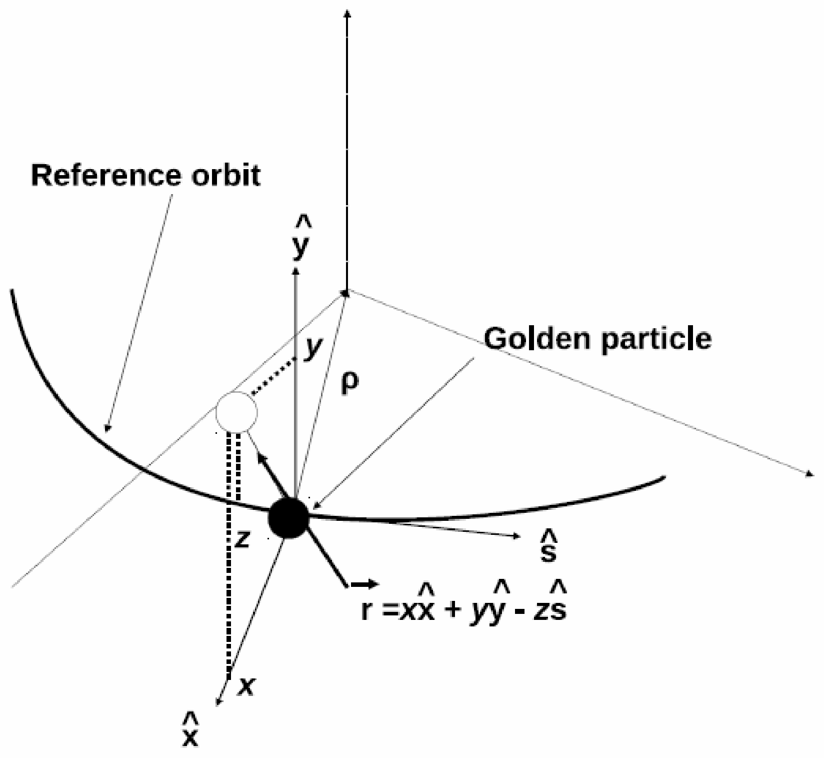
- Standard **assignments**:
  - Machine parameters properties.
  - Lattice structure.
  - Beam parameters definitions.
  - Errors and imperfections assigments.

- Standard **actions**:
  - Calculation of the lattice functions.
  - Machine corrections.

Convections

- **Units:** all parameters are in terms of **SI** units, except the **energy**, expressed on **GeV**.

- **Coordinate system:**
  - The elements are located around the **reference orbit** ($\hat{s}$).
  - $\hat{x}$ is the **horizontal** direction (the plane in which the beam is bent) and $\hat{y}$ is the **vertical** direction.



Courtesy of W. Herr.

Optic variables I

- Coordinates in MAD-X (all referred to the ideal orbit):
  - x and y: **Horizontal** and **vertical** positions x and y, in [m].
  - px and py: **Horizontal** and **vertical** canonical **momenta** p_x and p_y **divided** by the reference momentum: $\mathrm{px} = p_x/p_0$ and $\mathrm{py} = p_y/p_0$, in [1].
  - s: **Arc length** s along the reference orbit, in [m].
  - deltap: **Momentum deviation** from the design momentum, $\mathrm{deltap} = \Delta p/p_0$, in [1]. This quantity is used to **normalize** element strengths.

Optic variables II

- Twiss functions in MAD-X:
  - betx and bety: **Horizontal** and **vertical** beta functions β_x and β_y, in [m].
  - alfx and alfy: **Horizontal** and **vertical** alpha functions α_x and α_y, in [1].
  - mux and muy: **Horizontal** and **vertical** phase advances μ_x and μ_y, in [2π].
  - dx: Dispersion function D_x, in [m].

How to run MAD-X? Interactive mode

- In **Windows**, run the executable.
- In **Linux**, execute ./madx in the containing directory.

```
X: ==> angle = 2*pi/1232;
X: ==> value, angle;
X: ==> value, asin(1.0)*2;
X: ==> dx = gauss()*2.0;
X: ==> value, dx;
X: ==> value, dx;
X: ==> dx := gauss()*2.0;        Differed expression
X: ==> value, dx;
X: ==> value, dx;
```

How to run MAD-X? Interactive mode

- After writting your script in a separe file my.file (a file with a given lattice, for example), you can call it in **Windows** and **Linux** after opening MAD-X:

```
> madx
X: ==> call, file = “my.file”;
```

- In Linux, you can also type in the terminal

```
> madx < “my.file”
```

III. Machine description

- General format and classes
- Element definition: Dipoles
- Element definition: Quadrupoles
- Sequences

General format and classes

- All the machine elements must be defined.
- Element definitions follow a **general format:**

name: keyword, attributes;

- Elements may be described one-by-one, or as a member of a **class**.
  - All objects belonging to the same class **share the same properties**.

Element definition: Dipoles

- For bending magnets (dipoles):

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

- A dipole defined uniquely:

DIP01: SBEND, L=10.0, ANGLE=$angle$, K0 = k_0;

- Defining a class:

MBL: SBEND, L=10.0, ANGLE = 0.0145444;

- Creation of magnets (instances) from the same class:

DIP02: MBL; ! (instances of the class MBL)
DIP03: MBL; ! (instances of the class MBL)

Courtesy of W. Herr.

Element definition: Quadrupoles

- For quadrupoles:

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

- A quadrupole defined uniquely:

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

- Defining a class:

MQ: QUADRUPOLE, L=3.3, K1 = 1.23E-02;

- Creation of a magnet (instance) from the same class:

QUAD01: MQ;

Courtesy of W. Herr.

Sequences

```
circum = 6912;
// bending magnets as thin lenses
mbsps: multipole,knl={0.007272205};

// quadrupoles and sextupoles
kqf =  0.0146315;
kqd = -0.0146434;
qfsps: quadrupole,l=3.085,k1 := kqf;
qdsps: quadrupole,l=3.085,k1 := kqd;
lsf: sextupole,l=1.0,  k2 =  1.9518486E-02;
lsd: sextupole,l=1.0,  k2 = -3.7618842E-02;

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

cassps: sequence, l = circum;
start_machine: marker, at = 0;
qfsps,  at = 1.5425;
        ⋮
```

A constant

Dipole defined as multipole!

Elements definitions

Sequence starts here…

```
        ⋮
lsf, at = 3.6425;
ch,  at = 4.2425;
bpm, at = 4.3425;
mbsps, at = 5.0425;
mbsps, at = 11.4425;
mbsps, at = 23.6425;
mbsps, at = 30.0425;
        ⋮
bpm, at = 6884.3425;
mbsps, at = 6885.0425;
mbsps, at = 6891.4425;
mbsps, at = 6903.6425;
mbsps, at = 6910.0425;
end_machine: marker, at = 6912;
endsequence;
```

… and finishes here

Courtesy of W. Herr.

IV. MAD-X commands

- Basic commands
- Main script
- Output: Twiss summary and Twiss table
- Output: Plot and Survey

Basic commands

- Declaration of commands follow a **general format**:

command, attributes;

- Some **basic** commands (we have already saw some of them):

  - **Call** an external **file** (the definition of a machine, for example):

call, file = “my.file”;

  - **Print** a value in the terminal:

value, variable_name;

Basic commands

- Define a **beam**:

beam, particle = [proton | electron], energy = value;

- Compute the value of the **twiss functions** at each element:

select, flag = twiss, column = [name, s, betx, bety, mux, …];
twiss, save, centre, file = "my.twiss";

- Make a **plot**:

plot, haxis = s, vaxis = [betx, bety, ...], colour = 100;

- Geometry of the ring:

survey, file = "my.survey";

Main script

```
// Read input file with machine description
call file="sps.seq";
```

Read the accelerator lattice

```
// Define the beam for the machine
Beam, particle=proton, sequence=cassps, energy=450.0;
```

The type of beam and its energy

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

Use the sequence of the lattice

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

Keep the accelerator parameter in each element of the lattice

```
// 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;
```

Keep the information in a twiss file

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

Plot a variable in a range

```
// get the geometrical layout (survey)
survey,file=survey.cas;
```

Obtain the study geometry.

Courtesy of W. Herr.

```
stop;
```

Output: Twiss summary

```
++++++ table: summm
       length          orbit5              alfa         gammatr
         6912              -0  0.00167526597      24.4885807

           q1             dq1           betxmax           dxmax
  26.57999204  -8.828683153e-09     108.7763569     2.575386926

        dxrms          xcomax            xcorms              q2
  1.926988371               0                 0     26.62004577

          dq2         betymax             dymax           dyrms
4.9186549e-08     108.7331749                 0               0

       ycomax          ycorms            deltap         synch_1
            0               0                 0               0
```

- Machine length (circumference), in [m] → length
- Momentum compaction factor α_p, in [1] → alfa
- Transition energy gamma γ_{tr}, in [1] → gammatr
- Horizontal tune Q_x, in [1] → q1
- Maximum horizontal beta β_x^+, in [m] → betxmax
- Horizontal chromaticity Q'_x, in [1] → dq1
- Vertical tune Q_y, in [1] → q2
- Vertical chromaticity Q'_y, in [1] → dq2
- Maximum vertical beta β_y^+, in [m] → betymax
- Momentum deviation Q_y, in [1] → deltap

Courtesy of W. Herr.

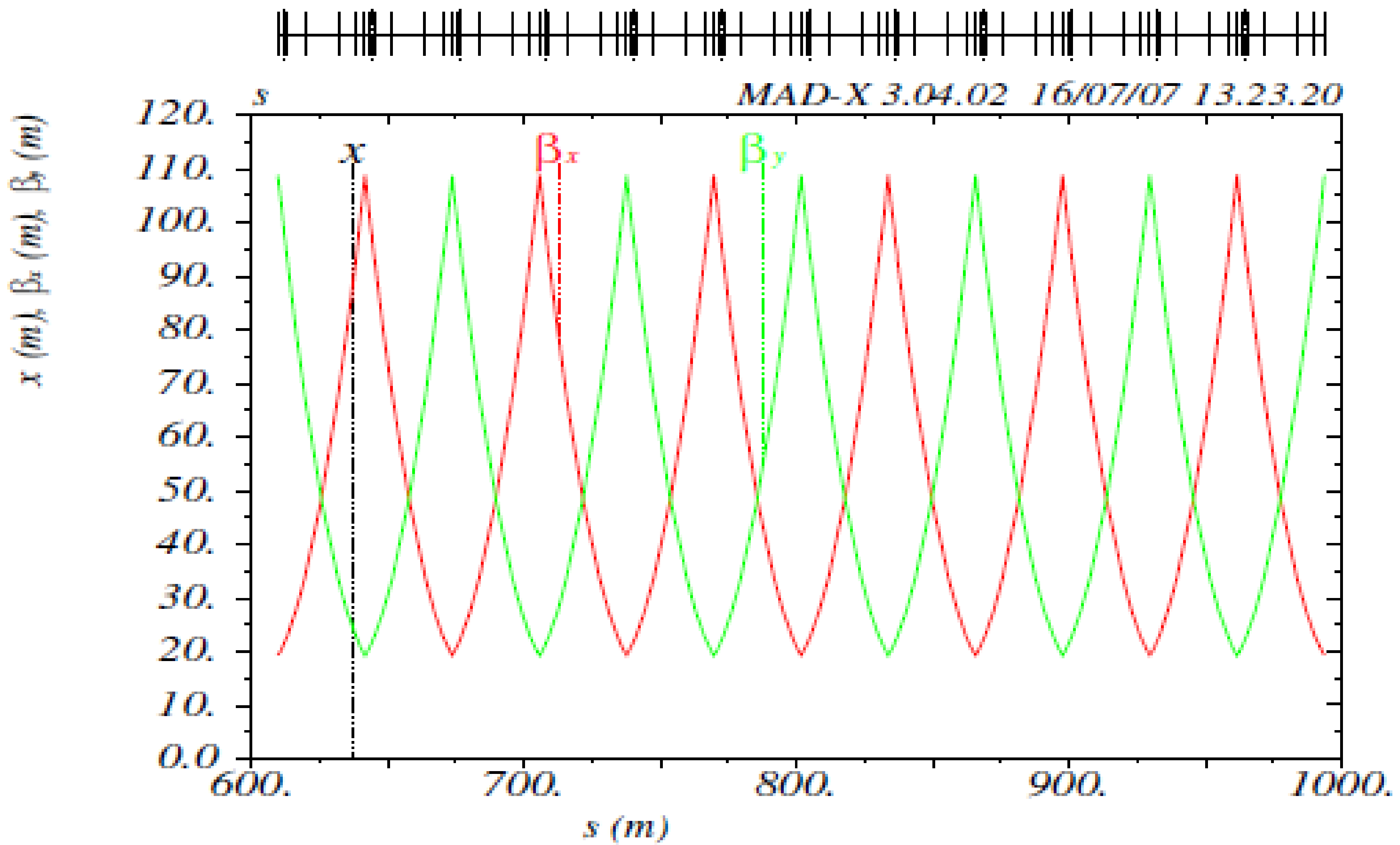
Output: Twiss table

* NAME	S	BETX	BETY
$ %s	%le	%le	%le
"CASSPS$START"	0	101.5961579	20.70328425
"START_MACHINE"	0	101.5961579	20.70328425
"DRIFT_0"	0.77125	105.1499566	19.94571028
"QF"	1.5425	108.7763569	19.26082066
"DRIFT_1"	2.5925	103.8571423	20.21112973
"LSF"	3.6425	99.07249356	21.29615787
"DRIFT_2"	3.9424975	97.73017837	21.6309074
"CH"	4.2425	96.39882586	21.97666007
"DRIFT_3"	4.2925	96.17800362	22.03535424
"BPM"	4.3425	95.95748651	22.0943539
"DRIFT_4"	4.6925025	94.4223997	22.51590816
"MBSPS"	5.0425	92.90228648	22.95242507
"DRIFT_5"	8.2425	79.69728195	27.63752778
"MBSPS"	11.4425	67.74212222	33.5738988
"DRIFT_6"	17.5425	48.41469349	48.35614376
"MBSPS"	23.6425	33.6289371	67.68523387
"DRIFT_5"	26.8425	27.68865546	79.6433337
"MBSPS"	30.0425	22.99821861	92.85270185
"DRIFT_7"	31.7925	20.96178735	100.6058286
"QD"	33.5425	19.29915001	108.7331749
"DRIFT_1"	34.5925	20.25187715	103.8118608

....

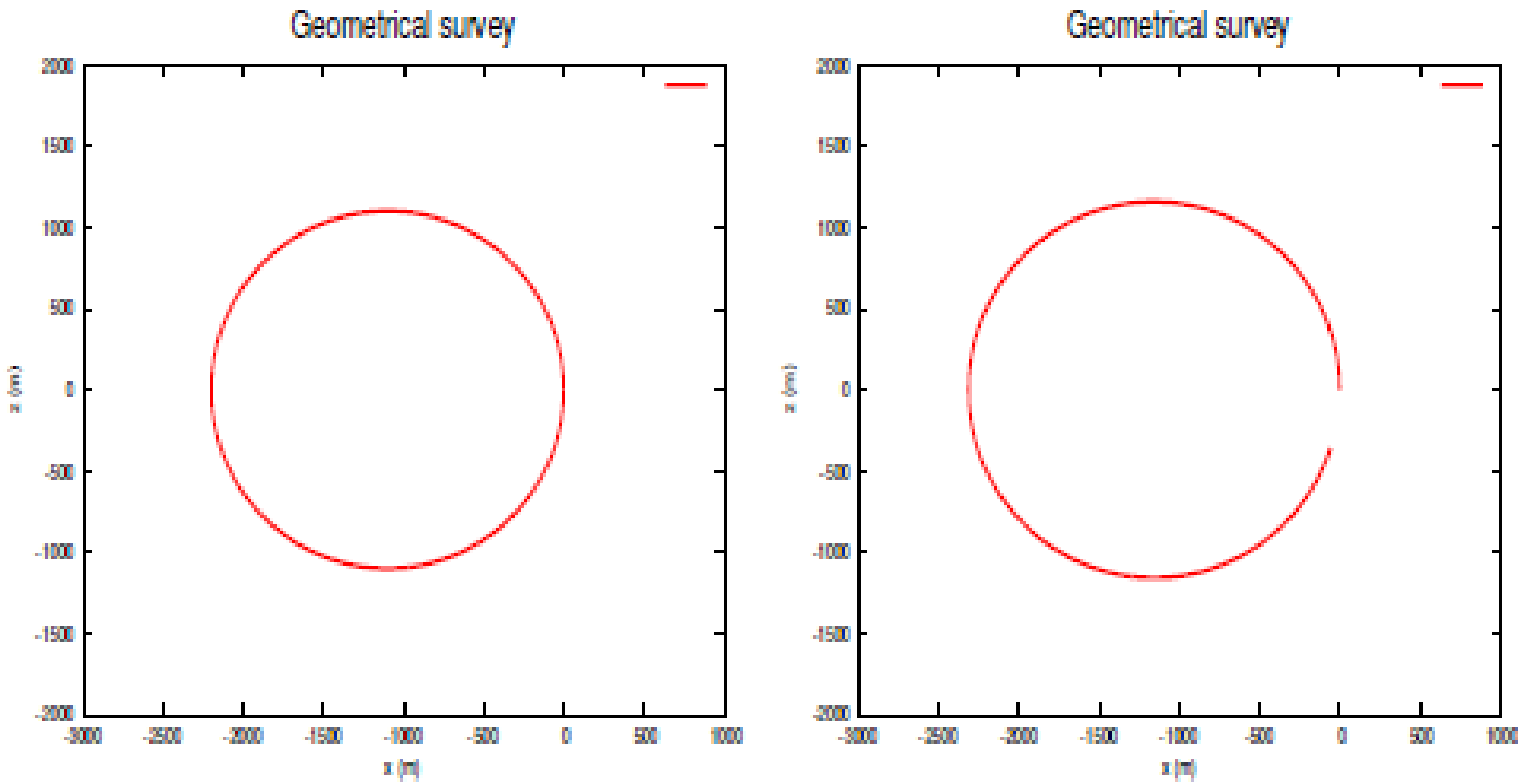
Courtesy of W. Herr.

Output: Plots



Courtesy of W. Herr.

Output: Survey

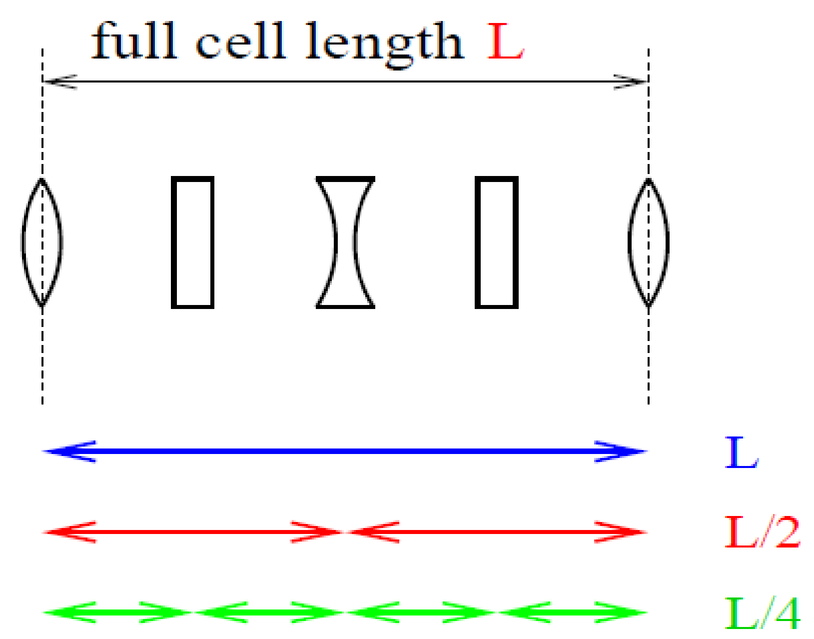


Courtesy of W. Herr.

Exercise I

- *Design a proton accelerator with the following properties:*
  - *Particle momentum: 20 GeV/c.*
  - *Circumference: 1000 m.*
  - *Dipole length: 5 m.*
  - *Maximum dipole field: 3 T.*
  - *Quadrupole length: 3 m.*

Use 8 FODO cells, the most basic cell used in accelerators



Courtesy of W. Herr.