SAD Seminar

Structural Definitions of Beam Line & Component

Since 1986
SAD
Strategic Accelerator Design

Optics Matching

Optical/Geometrical matching
Off-momentum matching
Finite-amplitude matching

Spin matching

SADScript Programming Interface in *Mathematica* Style

Built-in, system- and user-defined functions for accelerators
SAD/Tkinter/KBFrame
Tcl/Tk interface

Particle Tracking

6D full-symplectic tracking
Dynamic aperture survey
Synchrotron radiation

Nonlinear Analysis

Taylor map by automatic differentiation
Lie algebraic map

Emittance Calculation

6D Beam-matrix method
Anomalous emittance
Spin depolarization(SODOM)

24 Dec. 2015 @ KEK
K. Oide

http://acc-physics.kek.jp/SAD/index.html

Thank to N. Akasaka, E. Forest, K. Hirata, S. Kamada, M. Kikuchi, H. Koiso, S. Kuroda, T. Mimashi, A. Morita, E. Nakamura, K. Ohmi, Y. Ohnishi, N. Yamamoto, K. Yokoya, D. Zhou, and all who supported (criticized) SAD…

Conference room SAD

- Conference Room SAD
- Archived BBS ~2014/11
- Archived BBS ~2003/02
- Historical SAD mails

How to get Sources

- Licensing
- CVS Access

How to use SAD

- SAD/FFS Command & SAD Script
- SAD/FFS Examples
- FFS Level/KBFrame
  - SAD/Tkinter Manual (in JP, 706K 10/29/1997)
  - SAD/Tkinter Manual (in JP, 5155K 10/29/1997)
- Orbit correction manual
- Tracking in SAD
- References

SAD Wiki

SAD School

Workshops

- SAD Workshop 2006
- SAD Workshop 1998
  - Proceedings (10.3MB)

Links

- SAD computer Account application

Welcome to SAD/FFS & SADScript

The FFS commands are shown in uppercases. The minimum abbreviated form of each command is enclosed in (). Down to that form each command can be shorten. The optional arguments for the commands are usually enclosed in [].

SAD/FFS Examples

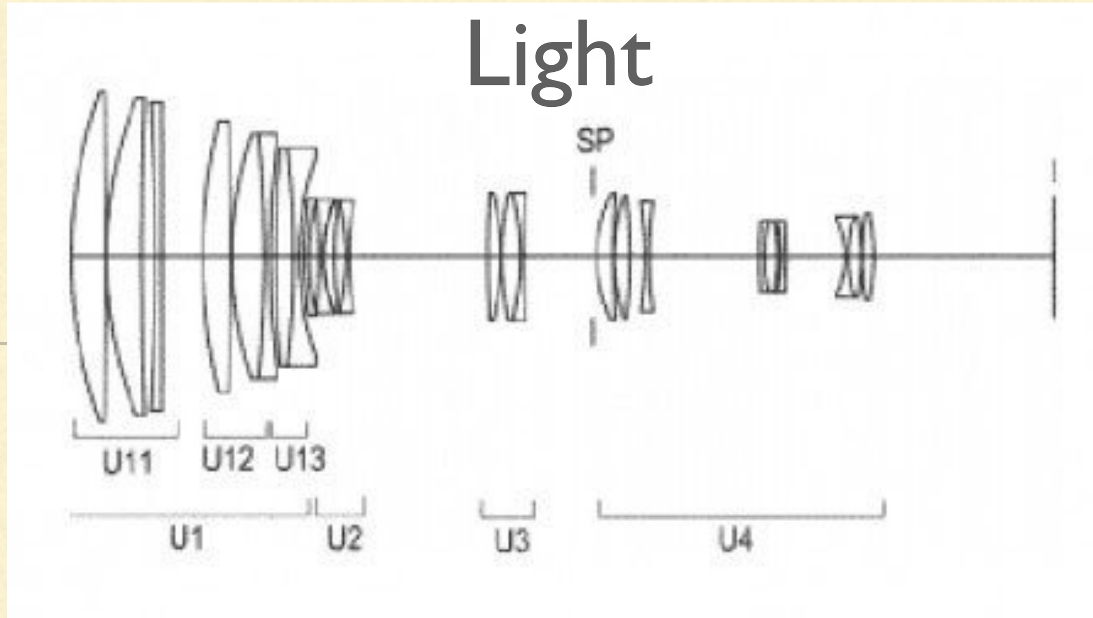
_

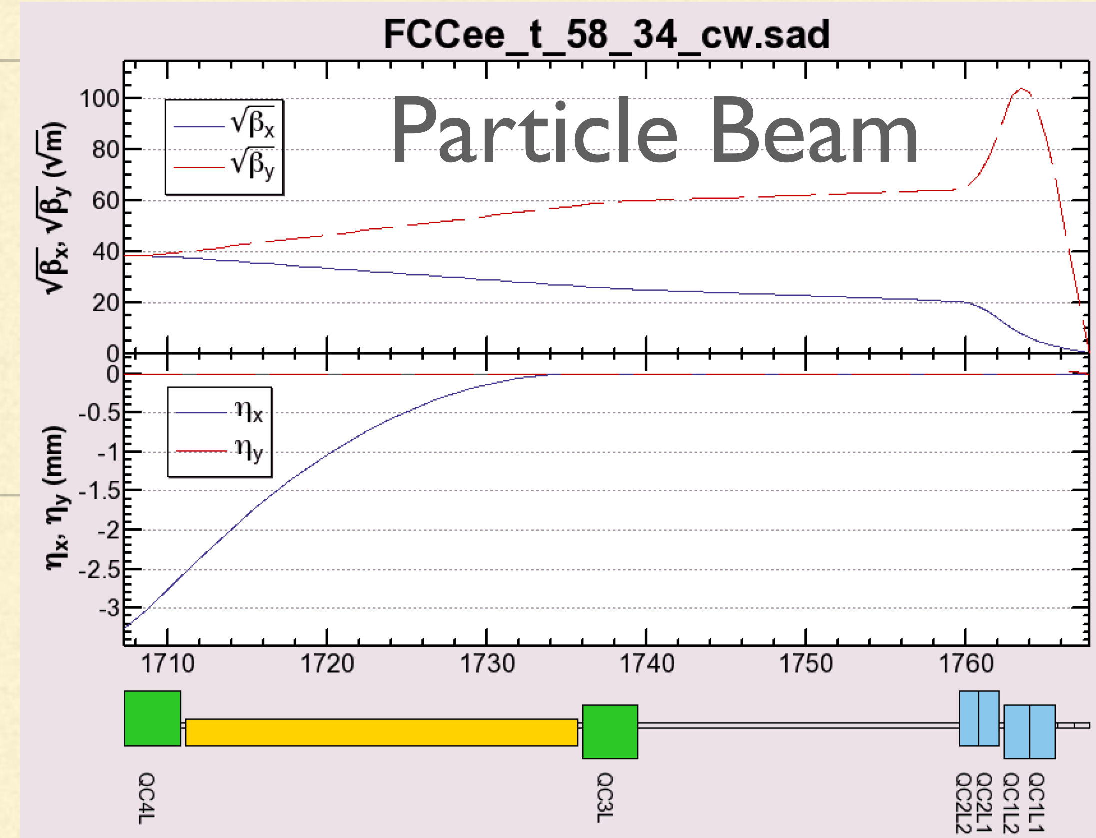
- ABORT
- APPEND(APP)
- ATTRIBUTE(ATTR)
- BYE
- command-syntax
- components
- CALCULATE(CAL)
- CHROMATICITY(CHRO)
- CLOSE(CLO)
- COUPLE(COUP)
- DISPLAY(DISP)
  - ACCELERATION(A)
  - ALL
  - BEAM(B)
  - DUMPOPTICS(D)
  - GEOMETRY(G)
  - OGEOMETRY(OG)
  - ORBIT(O)
  - pattern-string
  - PHYSICAL(P)
  - region

http://acc-physics.kek.jp/SAD/SADHelp.html

Please go to this page and use browser's search

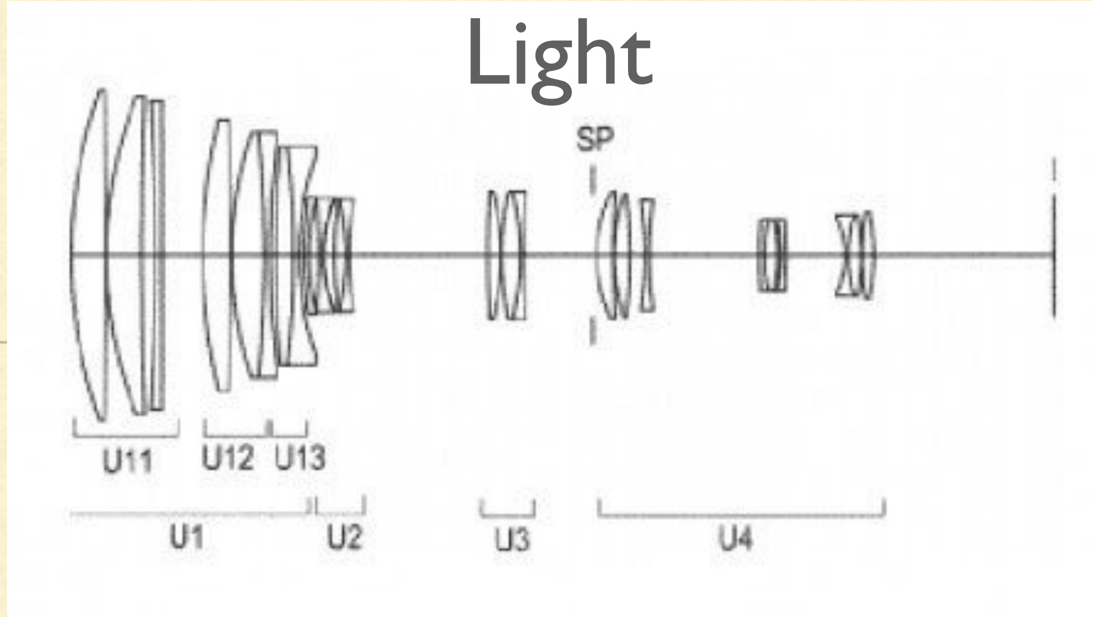
Optics

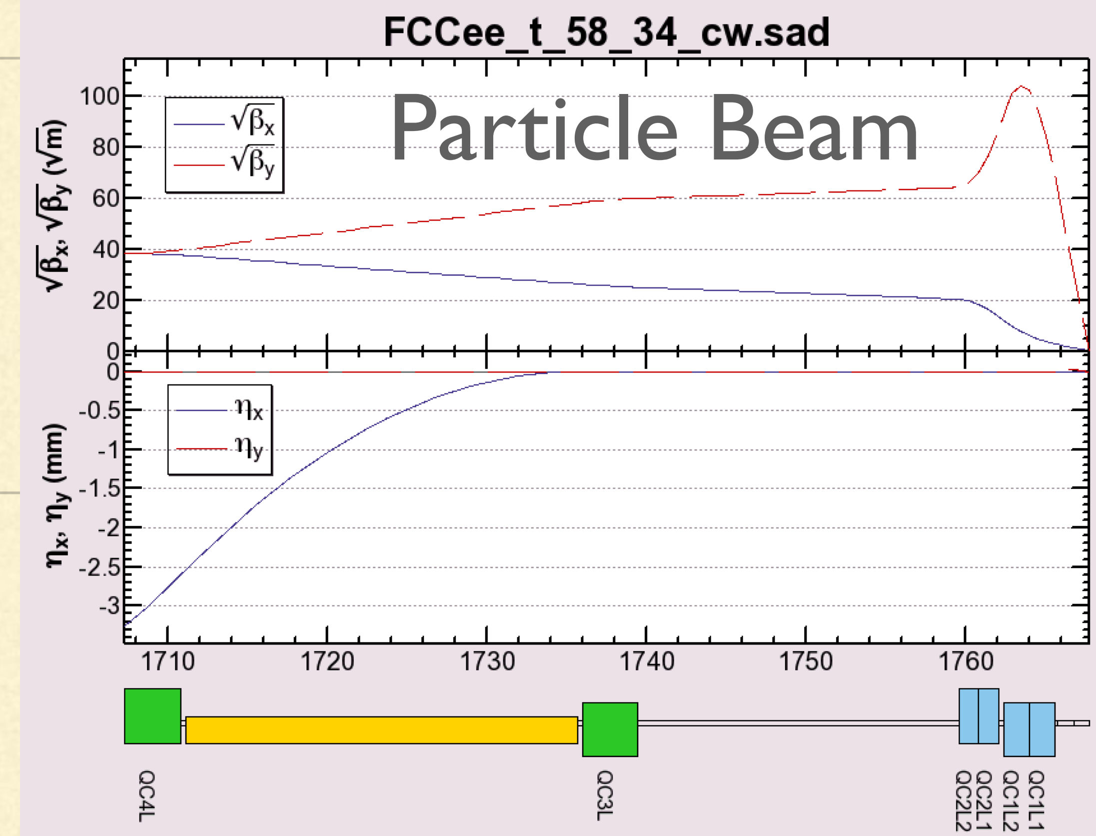




	Light	Particle
Focusing	Axially symmetric, mostly	Axially asymmetric
Emittance	$\lambda/4\pi \approx 50$ nm	1 mm - 1 pm
f-number	$f \gtrsim 1.0$	$f \gtrsim 10$
Transverse Aberrations	large, corrected by surface curvatures, aspherical lenses, etc.	small
Chromaticity	depends on the lens materials; corrected by their combinations	all lenses(drifts) have same chromatics effects; corrected by dispersion + sextupoles
Zoom	by moving lenses physically	by changing strengths of lenses
Storage	$F \lesssim 10{,}000$?	$> 10^{10}$ turns
Wave behavior	visible	mostly negligible

Optics





	Light	Particle
Focusing	Axially symmetric, mostly	Axi...
Emittance	$\lambda/4\pi \approx 50$ nm	
f-number	f ≳ 1.0	...10
Transverse Aberrations	large, co... cu...	small
	...n the lens materials; ...rrected by their combinations	all lenses(drifts) have same chromatics effects; corrected by dispersion + sextupoles
Zoom	by moving lenses physically	by changing strengths of lenses
Storage	F ≲ 10,000?	> 10^{10} turns
Wave behavior	visible	mostly negligible

Despite a number of these differences, the light/particle beam optics have fundamental similarities!

SAD deck

```
! MAIN Level
! COMMAND arg [arg1 …] ;
OFF CTIME; ON ECHO; ! a recommended setting of these flags.
! input after '!' is skipped.

element_type_command element_name = (key = value|expression [unit]
[key1 = value1 …]) [element_name1 = …] ;
…
LINE line_name = ([n*|-] element_name|line_name ….) [line_name1 = …] ;
…
FFS [USE=line_name];
! enters the FFS level
! Comments can be written within (* *).

command|expression [arg [arg1…]] [;] …

! Commands take args as many as acceptable: it is recommended to use
';' to terminate a command for clarification.
```

SAD deck (2)

```
! The units are basically in MKSA, eV, radian. Some FFS
commands take tune ≡ 1/2π (NX, NY), degree (CHI1, CHI2, CHI3),
etc as default.

! Expressions in FFS are Mathematica-like, can represent data
or programs.

! FFS commands can be used in an expression using a function
FFS["commands [; …]"] .

! The input stream on a terminal can be interrupted by

end

! and resumed by

in 77
```

Data types in FFS

- Real number: (no integer type)

```
1   3.14   3e8
```

- Character string:

```
"Hello World!"
"\"Hello\n\t World!\""
"A long character string can be written in multiple \
lines by placing \"\\\" at the end of a line."
```

- Symbol: starting with an alphabet or $, case-sensitive.

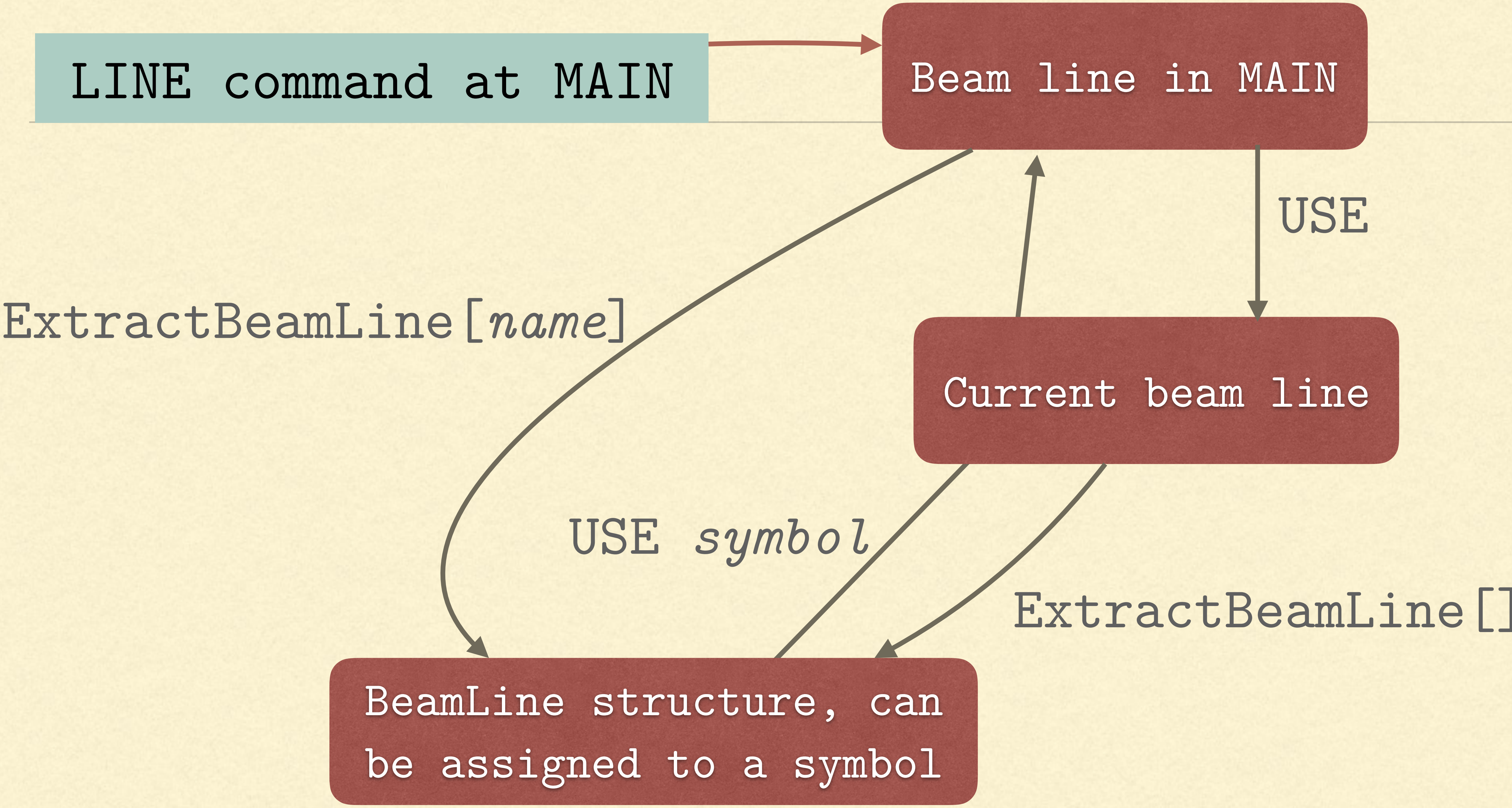
- List:

head[*body1*, …]

```
{1, 2, 3}   ≡ List[1, 2, 3]
(1 + 2 + 3)*4   ≡ Times[Plus[1, 2, 3], 4]
{1, 2, 3}[[2]]   ≡ Part[{1, 2, 3}, 2]
If[ x > 0, y = Sin[x] -1] ≡ If[ Greater[x, 0], Set[ y, Plus[Sin[x], -1]]]
```

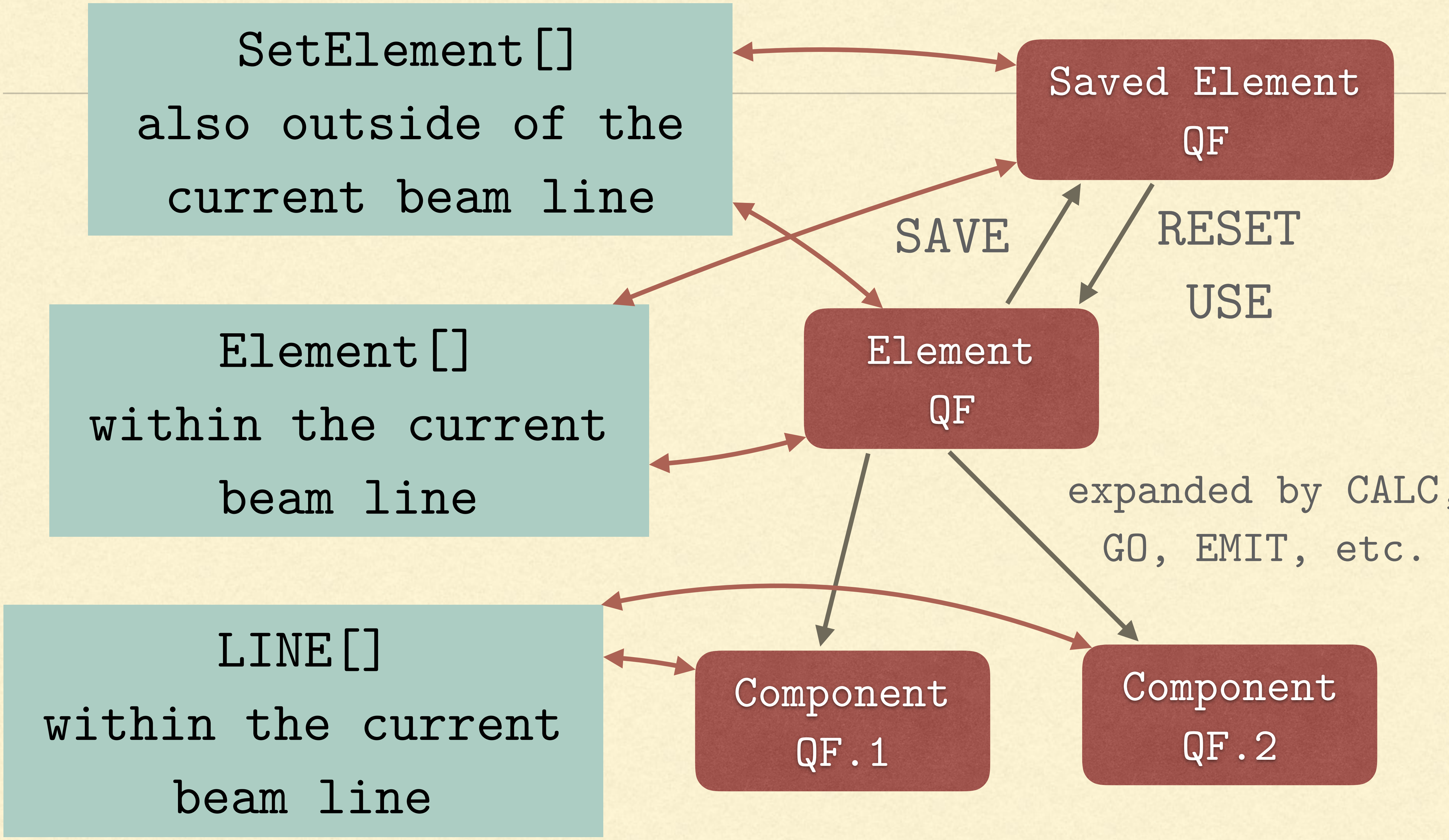
- All programs in SAD Script are expressed by List, similar to LISP, *Mathematica*, etc.

Beam lines



A beam line must start with a MARK element, if it is the object of USE.

Elements and components



Wildcards in FFS

`*`	`matches any number of (including zero) chars`
`%`	`matches any one character`
`{…}`	`matches any one character in {}`
`{^… }`	`matches any one character not in {}`
a|*b*|…	alternative; matches *a* or *b* …

Element[] and LINE[]

- Element[*key**, *name**] and LINE[*key**, *name**] accesses the element and component in the current beam line.

- Element and LINE are bidirectional.

- Arguments *key* and *name* are listable, and can be character strings with wildcards.

- Examples:

```
Element["K1","Q*"]
Element["K1","Q{FD}*"]
LINE[{"K1","L"},{"QF1.*","B1.*"}]
LINE[{"NAME","K1","L"},{"QF1.*|B1.*"}]
p=LINE["POSITION","QF1.*"];k1=LINE["K1",p]
```

Some predefined symbols

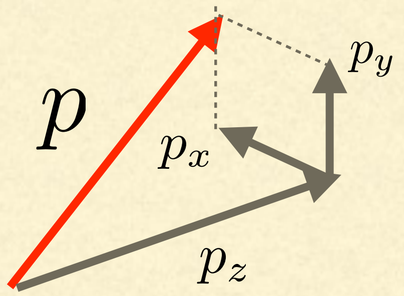
```
True (≡ 1)
False (≡ 0)

Pi    (PI in the MAIN level)
Degree (≡ 180/Pi)

SpeedOfLight (m/s)
ElectronCharge (Coulomb)
ElectronMass (eV)
FineStructureConstant
ElectronRadius (m)
```

DRIFT: Drift Space: the source of chromaticity (and some nonlinearity)

$$x \to x + (p_x/p_z)\ell \approx x + (p_x/p)\ell$$

$$y \to y + (p_y/p_z)\ell \approx y + (p_y/p)\ell$$

$$p_z = \sqrt{p^2 - p_x^2 - p_y^2}$$



- Even if a particle has same p_x and p_y, the displacement after a drift can be different depending on p: the source of chromaticity.
- A quadrupole gives the same $\Delta p_{x,y}$ for any particle, independent on p.

Kn: Strength of magnets

$$\mathtt{K}n = \frac{B^{(n)}\ell_{\mathrm{s}}}{(B\rho)}$$

$$B\rho \equiv (p_0/e)$$

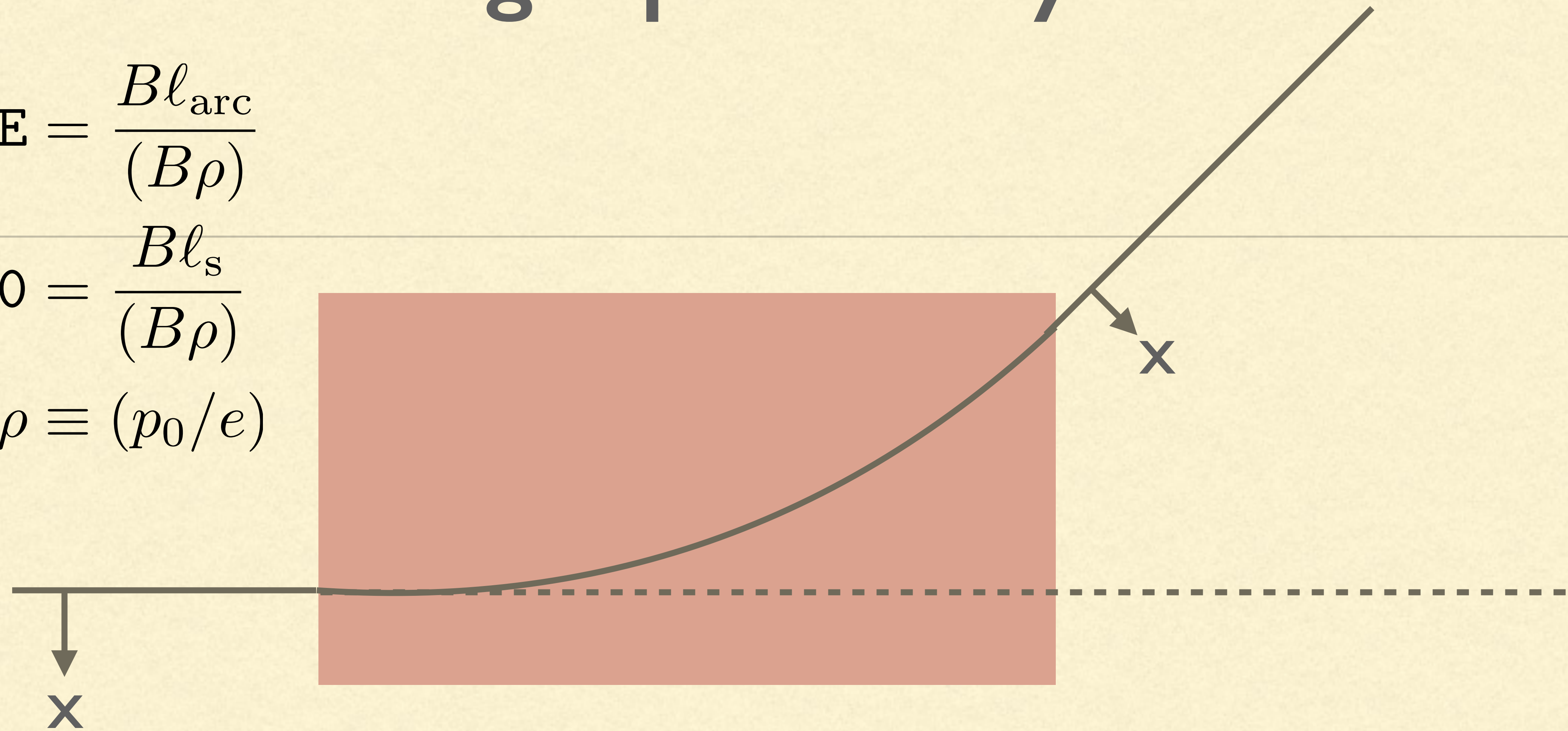
- Kn (n = 0 ..) represents the integrated strength of magnets in SAD.
- The length ℓ_s is the straight effective length of the magnet.
- Positive for horizontal focusing.

BEND: Bending dipole: Why ANGLE?

$$\texttt{ANGLE} = \frac{B\ell_{\mathrm{arc}}}{(B\rho)}$$

$$\texttt{K0} = \frac{B\ell_{\mathrm{s}}}{(B\rho)}$$

$$B\rho \equiv (p_0/e)$$



- If ANGLE $\neq$ 0, the coordinate after the bend follows it. No orbit distortion arises.

- If ANGLE = 0, the beam can be still bent by K0, but an orbit arises.

QUAD: Why K1?

$$\texttt{K1} = \frac{B'\ell_\mathrm{s}}{(B\rho)} = k$$

$$B\rho \equiv (p_0/e)$$

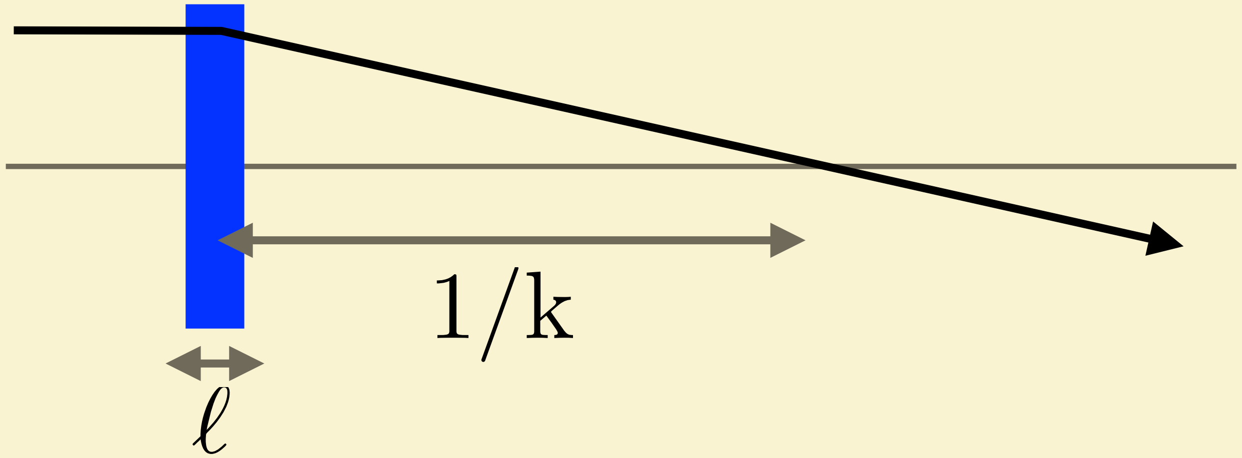
The transfer matrix $\boldsymbol{M}$ of a quadrupole in x-direction with $k > 0$ is approximated as

$$\boldsymbol{M} = \begin{pmatrix} \cos\sqrt{k\ell} & \sqrt{\ell/k}\sin\sqrt{k\ell} \\ -\sqrt{k/\ell}\sin\sqrt{k\ell} & \cos\sqrt{k\ell} \end{pmatrix} , \tag{1}$$

where $\ell = \ell_\mathrm{s}$. In the most cases $\ell \ll 1/k$, *ie*, the thickness of a lens is much smaller than the focal length. Then the matrix $\boldsymbol{M}$ is approximated as

$$\boldsymbol{M} \approx \begin{pmatrix} 1 & 0 \\ -k & 1 \end{pmatrix} + \epsilon \begin{pmatrix} -1/2 & 1/k \\ k/6 & -1/2 \end{pmatrix} + O(\epsilon)^2 , \tag{2}$$

where $\epsilon = k\ell$. Thus $\boldsymbol{M}$ is represented only by k at the lowest order of ϵ. This is the merit of using `K1` instead of something else like $B'/(B\rho)$.



Some important variables in FFS (* common with MAIN)

MOMENTUM*	design momentum at the entrance of the beam line (eV/c).
MASS	mass of the particle (eV). Defaults ElectronMass.
CHARGE	electric charge of the particle (e). Defaults 1.
CONVERGENCE*	goal of matching residual for GO. Defaults 1e-9, less than 1e-20 is recommended.
DP	width of the off-momentum matching.
DP0	Momentum offset of the optics.
MatchingResidual	the residual of matching: if it is less than CONVERGENCE(× # of eps), judged as "Matched".
NetResidual	The sum of square of the residual of eqs (× weight).
StabilityLevel	number of unstable optics
PBUNCH*	particles / bunch
MINCOUP*	minimum y/x emittance ratio assumed in EMIT, etc.
FSHIFT	df/f0 to compensate orbit dilation due to closed orbit distortion, (also chicanes, RADCOD, etc.)

Some flags (common with MAIN)

Flag	antonym	default	effect
CELL	INS	INS	impose a periodic condition on the entire beam optics
RING	TRPT	RING	design momentum is constant through the beam line, etc.
RFSW	NORFSW	RFSW	turns on acceleration
RAD	NORAD	NORAD	turns on synchrotron radiation in the tracking
RADCOD	NORADCOD	NORADCOD	include the radiation loss in the optics calculation
INTRA	NOINTRA	NOINTRA	calculate intrabeam scattering in EMIT or Emittance[]
CONV			True when matched, False otherwise

In FFS, flags can be set by just saying *flag*;.

In expressions, Use FFS["*flag*;"] .

Why matching?

- Even when the optics is linear, the solution for the parameters of elements (length, strength, etc.) to satisfy the condition cannot be written analytically.
  - The number of equations is usually more than 4.
- Thus a numerical solution is inevitable: called matching.
- Equations for matching in FFS:
  - built-in optical functions (fast)
  - any equations with FitFunction (slow).
- Variables for matching:
  - any key of an element
  - dependence between keys can be expressed by ElementValues.

Matching basics

- Location to be matched:

  FIT [*loc* | *loc1 loc2*]

- where *loc* has the form of

  element[*.nth*][+|- *offset*]

- The location specified by FIT is valid through the session, unless the beam line is switched by USE.

- Special locations:

^^^	beginning of the beam line
$$$	end of the beam line

- *loc* defaults $$$.
- If two locs are given it specifies a relative matching or region matching.

Matching basics (2)

- Goal of matching:

  fun[M] *value* [*nopts*];

- where *fun* is the built-in optical function at the current *loc* given by FIT, and *value* the goal value.
- If M is attached, it specifies the maximum value of *fun*.
- *nopts* is the number of optics for off-momentum matching.
- Goals of matching can be rejected by

  REJ *fun**; (* rejects *funs* matching to *fun* at this loc* *)

  REJ *; (* rejects all *funs at this loc* *)

  REJ total; (* rejects all *funs at all locs* *)

FitFunction

- Goals by FitFunction:

```
FitFunction:= a list of expressions
```

- where each *expression* is to be matched to 0:

```
FitFunction:= Module[{e = Emittance[]},
   {Max[0, Emittances[[1]]/emitx0 - 1],
   MomentumCompaction/alpha0 -1}/.e];
```

- above tries to match the maximum horizontal emittance and momentum compaction factor to be emitx0 and alpha0, respectively.

- FitFunction persists even after USE, so it should be cleared by:

```
FitFunction:=.
```

Optical functions for matching

AX, AY	αx. αy
BX, BY	βx, βy
NX, NY	ψx, ψy $(1/2\pi)$
EX, EY	ηx, ηy (in the normalized coordinate)
EPX, EPY	ηpx, ηpy (in the normalized coordinate)
R1, R4	x-y coupling parameter
R2	x-y coupling parameter (m)
R3	x-y coupling parameter (1/m)
DX, DY	Δx, Δy
DPX, DPY	Δpx, Δpy
DZ, DDP	Δz, $\Delta p/p_0$
PEX, PEY	ηx, ηy (in the physical coordinate)
PEPX, PEPY	ηpx, ηpy (in the physical coordinate)

Geometrical functions for matching

LENG	s
GX, GY, GZ	geometrical position ξ_x, ξ_y, ξ_z
CHI1, CHI2, CHI3	rotation of local coordinate χ_1, χ_2, χ_3 (degree).

- The origin of the geometrical functions can be given by the ORG command.

  ORG loc ΔGX, ΔGY, ΔGZ, CHI1, CHI2, CHI3;

- If a flag SORG is on, the origin of s is set to the *loc* given by ORG.

Control the goal value by FitValue

- It is possible to modify the goal value using a function FitValue:

    FitValue[*loc*, *fun*, {*id*_, *dp*_}, *v*_, *x*_]:= *expr*;

- where *v* is the goal value given by *fun* command, and *x* is the current value of *fun*.
- If FitValue returns a real number, it is used as the goal value. If a non-real is returned, the matching condition for *fun* is ignored.
- Example:

```
FitValue["$$$","EX",_,v_,x_]:=
    Which[x < v*0.9, v*0.9,
      x > v*1.1, v*1.1, True, Null ];
```

- The example above matches EX within 0.9*v ≦ x ≦ 1.1*v .

Matching basics (3)

- Matching variables:

```
FREE element [key] [element1 [key1]…];
```

- Wildcards can be used for *element*.

```
FREE Q* B* B* L;
```

- The default keys (can be changed by VARY command):

DRIFT	L
BEND	ANGLE
QUAD	K1
SEXT	K2
MULT	K1

- Variables are fixed by FIX with the same syntax as FREE.

Coupling between variables

- The default variables can be coupled to each other by the COUP command:

  COUP *slave master ratio*;

  where *slave* and *master* must be single elements, no wildcards are allowed. *ratio* is an expression giving a real number to give the ratio *slave/master* .

- For more general relations, use ElementValues:

  ElementValues= {"*key*"["*element*"]:> *expr*, …};

  where expr is a general expression which may also contain "*key1*"["*element1*"]:

```
Bq=1;      (* pole-tip field of a quad *)
aq=0.02;  (* bore radius of a quad *)
ElementValues=With[{q=#}, "L"[q]:> Abs["K1"[q]] * aq * Brho/Bq]&
   /@ Element["NAME", "Q*"];     (* give the lengths of quads, keeping
   their pole-tip field constant. *)
```

- ElementValues is persistent even switched to the other beam line. It should be cleared by

```
ElementValues=.
```

  before USE.

Matching basics (4)

- Some commands for matching

CALC	calculate the current optics
GO	do matching
REC	recover the optics before the previous GO (only one step)
SAVE [*elem** | all]	save the result to MAIN
RESET [*elem**]	restore parameters from MAIN
SHOW	display the current matching conditions
VAR [*elem**]	display variables

GO and CALC

```
 fit nx 0.375 ny 0.125;
 free q*;
 go;
 Iterations  Residual    Method     Reduction  Variables
       2    0.9726      (NEWTON)    1.000          2
       3    6.6733E-03  (NEWTON)    1.000          2
       4    1.2168E-05  (NEWTON)    1.000          2
       5    3.7061E-11  (NEWTON)    1.000          2
 Matched. ( 3.4175E-22) DP = 0.01000  DP0 = 0.00000  ExponentOfResidual = 2.0
 OffMomentumWeight =   1.000
$$$       f AX    #######  #  -3.874436 $$$       f BX    #######  #  12.491610
$$$       f NX   .375       1     .375000 $$$       f AY    #######  #   1.233800
$$$       f BY    #######  #   3.967586 $$$       f NY   .125       1     .125000
$$$       f LENG  #######  #   6.000000
 In[18]:=  CALC
 Matched. ( 3.4175E-22) DP = 0.01000  DP0 = 0.00000  ExponentOfResidual = 2.0
 OffMomentumWeight =   1.000
$$$       f AX    #######  #  -3.874436 $$$       f BX    #######  #  12.491610
$$$       f NX   .375       1     .375000 $$$       f AY    #######  #   1.233800
$$$       f BY    #######  #   3.967586 $$$       f NY   .125       1     .125000
$$$       f LENG  #######  #   6.000000
 In[19]:=
```

GO and CALC

```
fit nx 0.375 ny 0.125;
free q*;
go;
Iterations  Residual     Method      Reduction  Variables
      2    0.9726       (NEWTON)     1.000          2
      3    6.6733E-03  (NEWTON)     1.000          2
      4    1.2168E-05  (NEWTON)     1.000          2
      5    3.7061E-11  (NEWTON)     1.000          2
Matched. ( 3.4175E-22) DP = 0.01000  DP0 = 0.00000  ExponentOfResidual = 2.0
OffMomentumWeight =   1.000
$$$        f AX     #######  #  -3.874436 $$$        f BX     #######  #  12.491610
$$$        f NX    .375      1     .375000 $$$        f AY     #######  #   1.233800
$$$        f BY     #######  #   3.967586 $$$        f NY    .125      1     .125000
$$$        f LENG  #######  #   6.000000
In[18]:=  CALC
Matched. ( 3.4175E-22) DP = 0.01000  DP0 = 0.00000  ExponentOfResidual = 2.0
OffMomentumWeight =   1.000
$$$        f AX     #######  #  -3.874436 $$$        f BX     #######  #  12.491610
$$$        f NX    .375      1     .375000 $$$        f AY     #######  #   1.233800
$$$        f BY     #######  #   3.967586 $$$        f NY    .125      1     .125000
$$$        f LENG  #######  #   6.000000
In[19]:=
```

result with MatchingResidual

GO and CALC

```
fit nx 0.375 ny 0.125;
free q*;
go;
Iterations  Residual    Method     Reduction  Variables
      2    0.9726      (NEWTON)    1.0
      3    6.6733E-03  (NEWTON)    1.0
      4    1.2168E-05  (NEWTON)    1.000          2
      5    3.7061E-11  (NEWTON)    1.000          2
Matched. ( 3.4175E-22) DP = 0.01000  DP0 = 0.00000  ExponentOfResidual = 2.0
OffMomentumWeight =   1.000
$$$       f AX     #######  #  -3.874436 $$$       f BX     #######  #  12.491610
$$$       f NX    .375      1     .375000 $$$       f AY     #######  #   1.233800
$$$       f BY     #######  #   3.967586 $$$       f NY    .125      1     .125000
$$$       f LENG   #######  #   6.000000
In[18]:=  CALC
Matched. ( 3.4175E-22) DP = 0.01000  DP0 = 0.00000  ExponentOfResidual = 2.0
OffMomentumWeight =   1.000
$$$       f AX     #######  #  -3.874436 $$$       f BX     #######  #  12.491610
$$$       f NX    .375      1     .375000 $$$       f AY     #######  #   1.233800
$$$       f BY     #######  #   3.967586 $$$       f NY    .125      1     .125000
$$$       f LENG   #######  #   6.000000
In[19]:=
```

result with MatchingResidual

location to fit

GO and CALC

```
fit nx 0.375 ny 0.125;
free q*;
go;
Iterations  Residual    Method     Reduction  Variables
      2    0.9726      (NEWTON)    1.
      3    6.6733E-03  (NEWTON)    1.
      4    1.2168E-05  (NEWTON)    1.000          2
      5    3.7061E-11  (NEWTON)    1.000
Matched. ( 3.4175E-22) DP = 0.01000  DP0 = 0.000
OffMomentumWeight =   1.000
$$$         f AX     #######  #  -3.874436 $$$
$$$         f NX    .375      1    .375000 $$$        f AY     #######  #   1.233800
$$$         f BY     #######  #   3.967586 $$$        f NY    .125      1    .125000
$$$         f LENG   #######  #   6.000000
In[18]:=  CALC
Matched. ( 3.4175E-22) DP = 0.01000  DP0 = 0.00000  ExponentOfResidual = 2.0
OffMomentumWeight =   1.000
$$$         f AX     #######  #  -3.874436 $$$        f BX     #######  #  12.491610
$$$         f NX    .375      1    .375000 $$$        f AY     #######  #   1.233800
$$$         f BY     #######  #   3.967586 $$$        f NY    .125      1    .125000
$$$         f LENG   #######  #   6.000000
In[19]:=
```

result with MatchingResidual

location to fit

indicating current fit location

GO and CALC

```
fit nx 0.375 ny 0.125;
free q*;
go;
Iterations  Residual    Method     Reduction  Variables
      2    0.9726       (NEWTON)    1.
      3    6.6733E-03   (NEWTON)    1.
      4    1.2168E-05   (NEWTON)    1.000          2
      5    3.7061E-11   (NEWTON)    1.000
Matched. ( 3.4175E-22) DP = 0.01000  DP0 = 0.000
OffMomentumWeight =   1.000
$$$       f AX    #######  #  -3.874436 $$$
$$$       f NX    .375     1    .375000 $$$       f AY    #######  #   1.233800
$$$       f BY    #######  #   3.967586 $$$       f NY    .125     1    .125000
$$$       f LENG  #######  #   6.000000
In[18]:=  CALC
Matched. ( 3.4175E-22) DP = 0.01000  D                ExponentOfResidual = 2.0
OffMomentumWeight =   1.000
$$$       f AX    #######  #  -3.874436 $$$       f BX    #######  #  12.491610
$$$       f NX    .375     1    .375000 $$$       f AY    #######  #   1.233800
$$$       f BY    #######  #   3.967586 $$$       f NY    .125     1    .125000
$$$       f LENG  #######  #   6.000000
In[19]:=
```

result with MatchingResidual

location to fit

indicating current fit location

function

GO and CALC

```
fit nx 0.375 ny 0.125;
free q*;
go;
Iterations  Residual     Method      Reduction  Variables
      2     0.9726       (NEWTON)    1.
      3     6.6733E-03   (NEWTON)    1.
      4     1.2168E-05   (NEWTON)    1.000          2
      5     3.7061E-11   (NEWTON)    1.000
Matched. ( 3.4175E-22) DP = 0.01000  DPO = 0.000
OffMomentumWeight =   1.000
$$$        f AX    #######  #  -3.874436 $$$
$$$        f NX    .375     1    .375000 $$$       f AY    #######  #   1.233800
$$$        f BY    #######  #   3.967586 $$$       f NY    .125     1    .125000
$$$        f LENG  #######  #   6.000000
In[18]:=  CALC
Matched. ( 3.4175E-22) DP = 0.01000  I                ExponentOfResidual = 2.0
OffMomentumWeight =   1.000
$$$        f AX    #######  #  -3.8744                     #######  #  12.491610
$$$        f NX    .375     1    .3750                     #######  #   1.233800
$$$        f BY    #######  #   3.967586 $$$       f NY    .125     1    .125000
$$$        f LENG  #######  #   6.000000
In[19]:=
```

result with MatchingResidual

location to fit

indicating current fit location

function

goal value

GO and CALC

```
fit nx 0.375 ny 0.125;
free q*;
go;
Iterations  Residual     Method      Reduction  Variables
      2     0.9726       (NEWTON)    1.
      3     6.6733E-03   (NEWTON)    1.
      4     1.2168E-05   (NEWTON)    1.000           2
      5     3.7061E-11   (NEWTON)    1.000
Matched. ( 3.4175E-22) DP = 0.01000  DP0 = 0.000
OffMomentumWeight =   1.000
$$$        f AX     #######  #  -3.874436 $$$
$$$        f NX    .375      1    .375000 $$$       f AY    #######  #   1.233800
$$$        f BY     #######  #   3.967586 $$$       f NY    .125     1    .125000
$$$        f LENG   #######  #   6.000000
In[18]:=  CALC
Matched. ( 3.4175E-22) DP = 0.01000  D                Expon               = 2.0
OffMomentumWeight =   1.000
$$$        f AX     #######  #  -3.8744                                 2.491610
$$$        f NX    .375      1    .3750                                 1.233800
$$$        f BY     #######  #   3.967586 $$$       f NY    .125     1    .125000
$$$        f LENG   #######  #   6.000000
In[19]:=
```

result with MatchingResidual

location to fit

indicating current fit location

function

goal value

# of optics

GO and CALC

```
fit nx 0.375 ny 0.125;
free q*;
go;
Iterations  Residual    Method     Reduction  Variables
      2     0.9726      (NEWTON)   1.
      3     6.6733E-03  (NEWTON)   1.
      4     1.2168E-05  (NEWTON)   1.000           2
      5     3.7061E-11  (NEWTON)   1.000
Matched. ( 3.4175E-22) DP = 0.01000  DP0 = 0.000
OffMomentumWeight =   1.000
$$$        f AX    #######  #  -3.874436 $$$
$$$        f NX    .375     1    .375000 $$$        f AY    #######  #   1.233800
$$$        f BY    #######  #   3.967586 $$$        f NY    .125     1    .125000
$$$        f LENG  #######  #   6.000000
In[18]:=  CALC
Matched. ( 3.4175E-22) DP = 0.01000  D                Expon          = 2.0
OffMomentumWeight =   1.000
$$$        f AX    #######  #  -3.8744
$$$        f NX    .375     1    .3750
$$$        f BY    #######  #   3.967586 $$$        f NY    .125     1
$$$        f LENG  #######  #   6.000000
In[19]:=
```

result with MatchingResidual

location to fit

indicating current fit location

function

goal value

\# of optics

current value

DISP

DISP [R|O|A|G|B] [*elem**] [*range1* [*range2*]];

none: standard functions with length and values of elements

R: x-y coupling parameters   O: orbits

A: acceleration with emittances   G: geometry   B: beam sizes

disp

AX	BX	NX	EX	EPX	Element	Length	Value	s(m)	AY	BY	NY	EY	EPY	DetR	#
-3.8744	12.4916	.00000	.33846	.10519	P1	.00000	0	.000000	1.23380	3.96759	.00000	.00000	.00000	.0000	1
-3.8744	12.4916	.00000	.33846	.10519	QF	.20000	.6151954	.000000	1.23380	3.96759	.00000	.00000	.00000	.0000	2
3.87444	12.4916	.00250	.33846	-.10519	LXSF.1	.50000	.5000000	.200000	-1.2338	3.96759	.00819	.00000	.00000	.0000	3
3.23356	8.93761	.01003	.28587	-.10519	B1.1	2.00000	.0785398	.700000	-1.5516	5.36032	.02548	.00000	.00000	.0000	4
.66821	1.11801	.11892	.15332	-.02729	LX03.1	.30000	.3000000	2.700000	-2.8230	14.1098	.06241	.00000	.00000	.0000	5
.28006	.83353	.16922	.14513	-.02729	QD	.20000	-.3784105	3.000000	-3.0138	15.8608	.06561	.00000	.00000	.0000	6
-.28006	.83353	.20828	.14513	.02729	LX03.2	.30000	.3000000	3.200000	3.01381	15.8608	.06759	.00000	.00000	.0000	7
-.66821	1.11801	.25857	.15332	.02729	B1.2	2.00000	.0785398	3.500000	2.82309	14.1098	.07078	.00000	.00000	.0000	8
-3.2335	8.93761	.36747	.28587	.10519	LXSF.2	.50000	.5000000	5.500000	1.55166	5.36032	.10771	.00000	.00000	.0000	9
-3.8744	12.4916	.37500	.33846	.10519	$$$	.00000	0	6.000000	1.23380	3.96759	.12500	.00000	.00000	.0000	10

disp o

AX	BX	NX	EX	EPX	Element	DX	DPX	DY	DPY	AY	BY	NY	EY	EPY	DetR	#
-3.8744	12.4916	.00000	.33846	.10519	P1	.0000	.0000	.0000	.0000	1.23380	3.96759	.00000	.00000	.00000	.0000	1
-3.8744	12.4916	.00000	.33846	.10519	QF	.0000	.0000	.0000	.0000	1.23380	3.96759	.00000	.00000	.00000	.0000	2
3.87444	12.4916	.00250	.33846	-.10519	LXSF.1	.0000	.0000	.0000	.0000	-1.2338	3.96759	.00819	.00000	.00000	.0000	3
3.23356	8.93761	.01003	.28587	-.10519	B1.1	.0000	.0000	.0000	.0000	-1.5516	5.36032	.02548	.00000	.00000	.0000	4
.66821	1.11801	.11892	.15332	-.02729	LX03.1	.0000	.0000	.0000	.0000	-2.8230	14.1098	.06241	.00000	.00000	.0000	5
.28006	.83353	.16922	.14513	-.02729	QD	.0000	.0000	.0000	.0000	-3.0138	15.8608	.06561	.00000	.00000	.0000	6
-.28006	.83353	.20828	.14513	.02729	LX03.2	.0000	.0000	.0000	.0000	3.01381	15.8608	.06759	.00000	.00000	.0000	7
-.66821	1.11801	.25857	.15332	.02729	B1.2	.0000	.0000	.0000	.0000	2.82309	14.1098	.07078	.00000	.00000	.0000	8
-3.2335	8.93761	.36747	.28587	.10519	LXSF.2	.0000	.0000	.0000	.0000	1.55166	5.36032	.10771	.00000	.00000	.0000	9
-3.8744	12.4916	.37500	.33846	.10519	$$$	.0000	.0000	.0000	.0000	1.23380	3.96759	.12500	.00000	.00000	.0000	10

Movable range of variables

- The movable range of a default key can be limited by MIN MAX, and MINMAX:

```
QF* MIN 0; (* 0 ≦ x *)
QD* MAX 0; (* x ≦ 0 *)
B*  MINMAX 0.1 (* -0.1 ≦ x ≦ 0.1 *)
```

- For any variable, a function VariableRange sets the range:

```
Bmax=1.5;
VariableRange["B1","L",v_]:=
   {0, Brho / Bmax * Abs[LINE["ANGLE","B1"]]};
```

Execute FFS commands in expressions

- FFS[*commands_string*] excuses character-storing *commands_string* as FFS commands. It may return the results for some commands such as CALC, GO, VAR, SHOW, etc.

```
 In[10]:= FFS["CALC"]
Out[11]:= {{0},{0},{{3.417528475389843e-22,1,1}},{{"$$$","","AX",
{-3.874435819115132}},{"$$$","","BX",{12.491609914888326}},{"$$$","","NX",
{2.356194490192527 4}},{"$$$","","AY",{1.2338003736307515}},{"$$$","","BY",
{3.967585589568553 7}},{"$$$","","NY",{.7853981633789626}},{"$$$","","LENG",{6}}}}
```

- Multiple commands are accepted by a single FFS[]:

```
FFS[" cell;\
 fit nx 0.375 ny 0.125;\
 free q*; go;"]
```

- FFS[*commands_string*, *out*] redirect the output to unit number *out*. If *out* is $Output or -1, it redirects to the current output unit.

VAR and SHOW

```
 In[21]:= var
!Variable      Keyword     Now            !   Previous     Saved          Minimum       Maximum       Couple       Coefficient
 QF            K1         .615195427585 !  .100000000  .100000000  -1.00000E10   1.000000E10 <--              1.00000000
 QD            K1        -.378410496646 ! -.100000000 -.100000000  -1.00000E10   1.000000E10 <--              1.00000000
 In[22]:= FFS["VAR"]
Out[232]:= {{"QF","K1",.6151954275846839,.1,.1,-1e+10,1e+10,"QF",1},
{"QD","K1",-.3784104966460243,-.1,-.1,-1e+10,1e+10,"QD",1}}

 In[23]:= show
!    component1   component2   fun        goal-value  np     scale
 FIT $$$                       NX           .375000000  1 ! * 6.283185307
 FIT $$$                       NY           .125000000  1 ! * 6.283185307
Out[23]:= FFS["SHOW"]
Out[24]:= {{"$$$","","NX",.375,1,6.283185307179586},{"$$$","","NY",.
125,1,6.283185307179586}}
In[23]:=
```

EMIT

The `EMIT` command calculates the equilibrium beam matrix:

$$\boldsymbol{B} \equiv \langle q_i q_j \rangle, \qquad q_i = (x, p_x, y, p_y, z, \delta) \tag{1}$$

using

$$\boldsymbol{B} = \boldsymbol{M}\boldsymbol{B}\boldsymbol{M}^T + \boldsymbol{b}\ , \tag{2}$$

where $\boldsymbol{M}$ is the one-turn transfer matrix including damping, and $\boldsymbol{b}$ the excitation matrix due to synchrotron radiation and intrabeam scattering. This is a simple but powerful method especially when $\boldsymbol{M}$ has x-y or x-y-z coupling terms.

```
    Closed orbit:                                                      
                  x      px/p0         y      py/p0         z      dp/p0          Imag.tune:-0.0000000            0.0000000            -0.0000000
  Entrance :    .000000    .000000    .000000    .000000    .000000    .000000    Real tune: 0.3750000            0.1250000            -0.0000000
      Exit :    .000000    .000000    .000000    .000000    .000000    .000000
                                                                                    Damping per one revolution:
    Extended Twiss Parameters:                                                            X : -1.162935E-06    Y : -1.172449E-06    Z : -2.354406E-06
AX: -3.87443 BX: 12.49161                            ZX: 5.86E-16   EX:   .338461     Damping time (sec):
           PSIX: -1.6E-17                           ZPX: 2.93E-16 EPX:   .105191            X :   1.720978E-02    Y :   1.707013E-02    Z :   8.500592E-03
R1:   .000000 R2:   .000000 AY: 1.233800 BY: 3.967586  ZY:   .000000   EY:   .000000    Tune shift due to radiation:
R3:   .000000 R4:   .000000            PSIY:   .000000 ZPY:   .000000 EPY:   .000000            X : -3.098860E-13    Y : -1.460987E-13    Z :   6.080434E-09
                                                      AZ:   .000000   BZ: 1.000000      Damping partition number:
                                                                  PSIZ:   .000000            X :     0.9919         Y :     1.0000         Z :     2.0081
    Units: B(X,Y,Z), E(X,Y), R2: m | PSI(X,Y,Z): radian | ZP(X,Y), R3: 1/m
                                                                                  Emittance X            = 8.64279E-9 m   Emittance Y              =  .00000000 m
Design momentum      P0 = 3.0000000 GeV Revolution freq.     f0 = 49965408. Hz   Emittance Z            =  .00000000 m   Energy spread            = 5.08214E-4
Energy loss per turn U0 =  .0070347 MV  Effective voltage    Vc =  .0000000 MV   Bunch Length           =  .00000000 mm  Beam tilt                =  .00000000 rad
Equilibrium position dz =  .0000000 mm  Momentum compact. alpha =  .0054089      Beam size xi           =  .37087720 mm  Beam size eta            =  .00000000 mm
Orbit dilation       dl =  .0000000 mm  Effective harmonic #  h =  .0000000
Bucket height     dV/P0 =  .0000000
```

Emittance[]

- Emittance[] does the same thing as EMIT, but returns the result as a list of Rules (->):

```
 In[11]:= Emittance[]
Out[11]:= {Stable->1,Tunes->{.3750000000002826,.1249999999705737,-0},
EnergyLossU0->7034.697060646969,RfVoltageVc->0, EquilibriumPosition>0,
MomentumCompaction->.005408933078228795,OrbitDilation->0,
BucketHeight->0,HarmonicNumber->0,OrbitAtExit->{0,0,0,0,0,0},DampingRate->
{-1.162934551369965 3e-06,-1.172448675392321e-06,-2.354406166 2267986e-06},
Emittances->{8.642785032172451e-09,0,0},MomentumSpread->000508213708964793,
BunchLength->0,TuneShiftByRadiation->
{-3.098860468979787 4e-13,-1.460986955594973e-13,6.080433798344304e-09}}
```

- The result can be obtained by using ReplaceAll (/.):

```
 In[12]:= e=Emittance[];
 In[12]:= Emittances[[1]]/.e
Out[12]:= 8.642785032172451e-09
```

Operations on Lists

- Map (/@):

```
 In[14]:= Map[f, {1,2,3}]
Out[14]:= {f[1],f[2],f[3]}

 In[15]:= f/@{1,2,3}
Out[15]:= {f[1],f[2],f[3]}

 In[16]:= f/@[{{1,2},{3,4},{5,6}}, {2}]
Out[16]:= {{f[1],f[2]},{f[3],f[4]},{f[5],f[6]}}
```

- Apply (@@):

```
 In[17]:= Apply[f, {1,2,3}]
Out[17]:= f[1,2,3]

 In[18]:= f@@{1,2,3}
Out[18]:= f[1,2,3]

 In[19]:= f@@[{{1,2},{3,4},{5,6}}, {1}]
Out[19]:= {f[1,2],f[3,4],f[5,6]}
```

More functions for list operations

<table>
<tr><th>function</th><th>description</th></tr>
<tr><td>Thread</td><td>Thread[{{1,2,3},{4,5,6}}]
→ {{1,4},{2,5},{3,6}}</td></tr>
<tr><td>MapThread</td><td>MapThread[f, {{1,2,3},{4,5,6}}]
→ {f[1,4],f[2,5],f[3,6]}</td></tr>
<tr><td>Scan</td><td>Map without output; faster & less memory</td></tr>
<tr><td>Position</td><td>Position[{a,b,c,d,a,e}, a] → {{1},{5}}</td></tr>
<tr><td>Cases</td><td>Cases[{a,b,c,d,a,e}, a|d] → {a,d,a}</td></tr>
<tr><td>DeleteCases</td><td>DeleteCases[{a,b,c,d,a,e}, a|d] → {b,c,e}</td></tr>
<tr><td>Select</td><td>Select[{1,4,-1,1,2}, (#>1)&] → {4,2}</td></tr>
<tr><td>Sort</td><td>Sort[{1,4,-1,1,2}] → {-1,1,1,2,4}</td></tr>
<tr><td>Union</td><td>Union[{1,4,-1,1,2},{3,-1}] → {-1,1,2,3,4}</td></tr>
<tr><td>Intersection</td><td>Intersection[{1,4,-1,1,2}, {3,-1}] → {-1}</td></tr>
</table>

More functions for list operations (2)

function	description
Length	number of elements in a list
Dimensions	dimensions of a matrix or a tensor
Part	{{1,2,3},{4,5,6}}[[2]] → {4,5,6} {{1,2,3},{4,5,6}}[[-2]] → {1,2,3} {{1,2,3},{4,5,6}}[[1,2]] → 2 {{1,2,3},{4,5,6}}[[2,{1,3}]] → {4,6}
Take	Take[{1,4,-1,1,2},3] → {1,4,-1} Take[{1,4,-1,1,2},-3] → {-1,1,2} Take[{1,4,-1,1,2},{3,-2}] → {-1,1}
Drop	Drop[{1,4,-1,1,2},3] → {1,2} Drop[{1,4,-1,1,2},-3] → {1,4} Drop[{1,4,-1,1,2},{3,-2}] → {1,4,2}

… and more

Listable operations

- Many arithmetic operations and functions are "listable", *i.e.*, they operate parts by parts:

```
{a, b, c} * {d, e, f} → {a d, b e, c f}
r * {a, b, c} → {r a, r b, r c}
Sin[{a, b, c}] → {Sin[a], Sin[b], Sin[c]}
```

- A listable operation is faster than an operation using Part and index, so is encouraged to use:

```
 In[21]:= u=Range[1000]
Out[21]:={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,
 ...
 In[22]:= Timing[u*2]
Out[22]:= {2.7000904083251953e-05,
{2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,46,
...
 In[23]:= Timing[Table[u[[i]]*2,{i,Length[u]}]]
Out[23]:= {.00054502487182617 19,
{2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,46,
...
```

Pure function

- A pure function is a function without name:

```
(# + #2)&[a, b] → a+b

ave = ((Plus@@#) /Length[#])&[list]
rms = Sqrt[(Plus@@[(# - ave)^2]) /Length[#]]&[list]
```

#	the first arg
#*n*	the *n*th arg
##	sequence of all args
##*n*	sequence of args from #*n* to end

- Another form of pure function:

```
Function[{a,b,c}, a+b+c][1,2,3] → 6
```

Module

- Module[{*var1*, …}, *body*] defines local symbols *var1*,…, which are usable within *body*, then evaluate *body*. Local symbols are abandoned after the exit of Module.

```
a=1; Module[{a}, a=2]; a  → 1
```

- The local symbols must explicitly appear in *body*.

```
Clear[a,f]; f=a; Module[{a}, a=2; {f,a}] → {a, 2}
```

- The initial values of local symbols can be given as:

Module[{*var1*=*value*, {*value2*, *value3*, …}=*list*, …}, *body*]

Defining a function

- A function *f* with arguments *arg1*,… can be defined as:

  `f[arg1_, …]:= body;`

- where *arg1_* is a pattern to match any actual argument when *f*[…] is evaluated. Every symbol *arg1* appearing in *body* is replaced by with the actual argument *before* the evaluation of *body*.

- The symbol *arg1* must explicitly appear in body.

*x*_	matches anything, replacing symbol *x*
*x*__	matches any non-null sequence, replacing symbol *x*
*x*___	matches any sequence, can be null, replacing symbol *x*
*x*_*h*	matches anything having head *h*
*x*_?(*test*)	matches anything *test*[*x*] gives True

Defining a function (2)

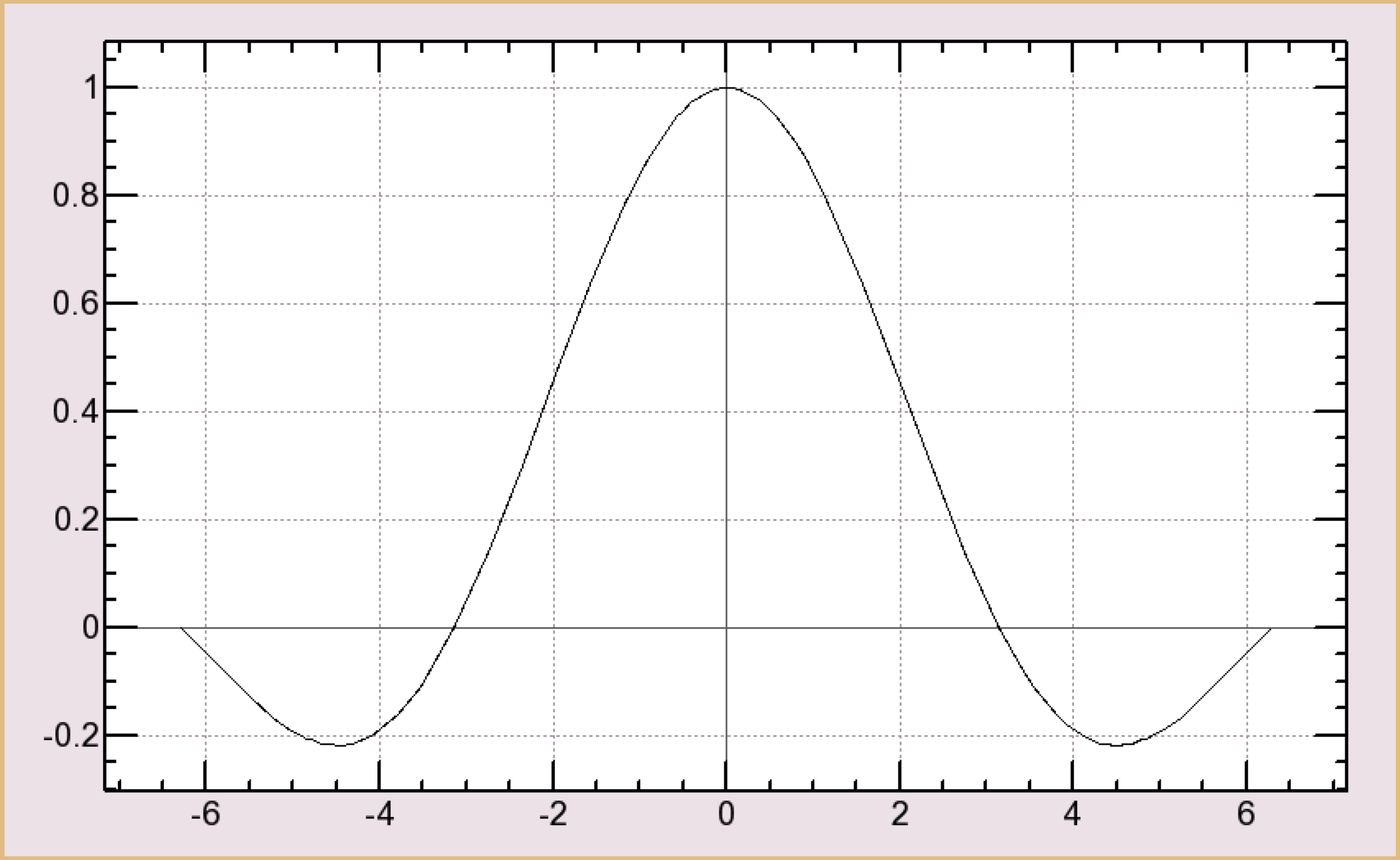
- Example:

```
f[x_]:=Sin[x]/x;
f[0]=1;
Plot[f[x], {x, -2Pi, 2Pi}];
Update[];
```

Defining a function (2)

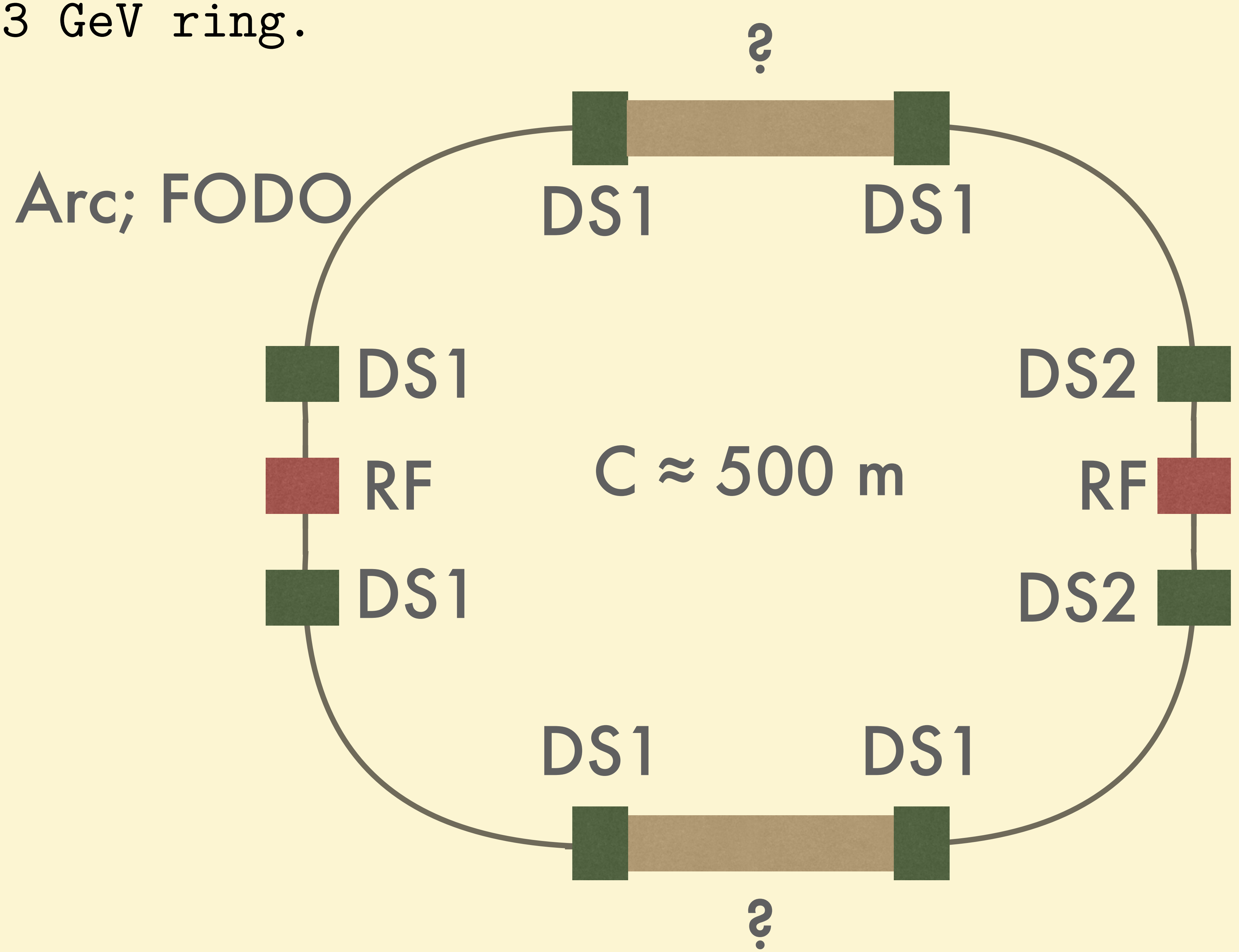
- Example:

```
f[x_]:=Sin[x]/x;
f[0]=1;
Plot[f[x], {x, -2Pi, 2Pi}];
Update[];
```

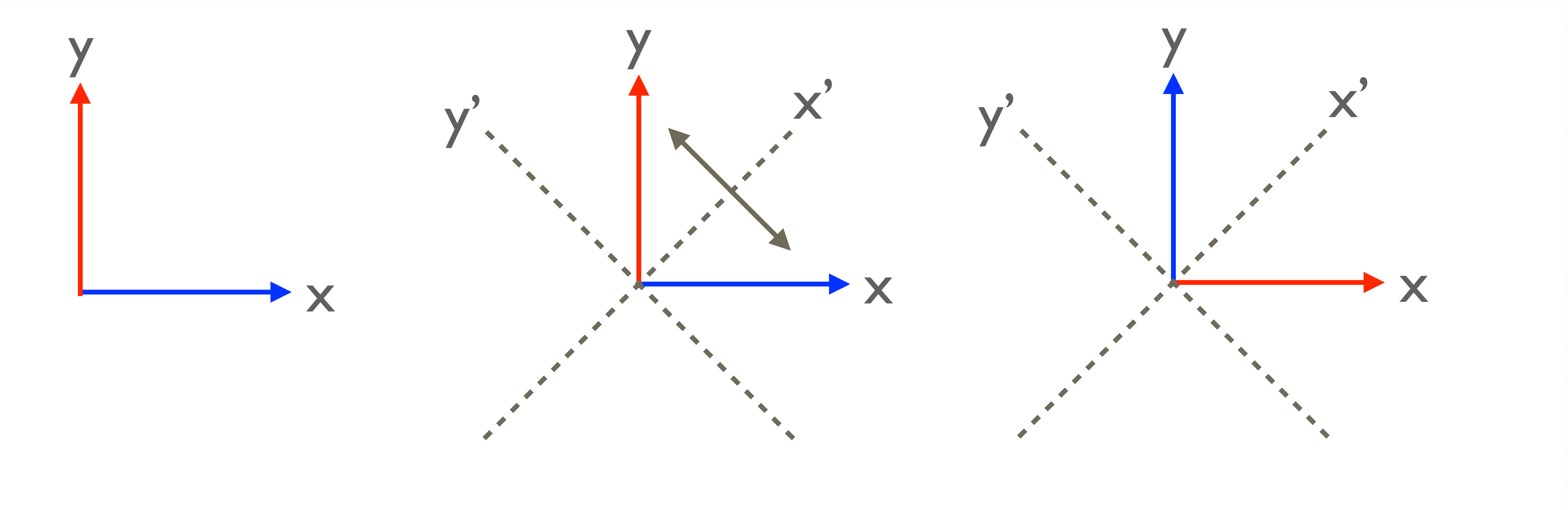


What are we building in this seminar?

- A 3 GeV ring.



Exchange x & y



- x and y coordinates interchange, if they are inverted along the y' axis. The coordinates (x', y') are rotated by 45 deg from (x, y): the axes of a skew quadrupole.