Introduction to Optics Design

G. Sterbini, BE-ABP-HSI, CERN

9-21 June 2019, CAS, Slangerup, Denmark

guido.sterbini@cern.ch

Linear Optics Calculations

1. Introduction
2. Single Particle
   - Twiss parameters
   - CS invariant
   - CO, D and ξ
3. Ensembles
   - Beam emittance
   - Matched distribution
4. MAD-X
   - MAD-X syntax
   - “Hello World!”

Linear Optics Calculations

Goal

The aim of the "Linear Optics Calculations" lecture is three-fold:

- to recall the matrix formalism applied to Linear Optics,
- to use the matrix formalism to perform Linear Optics Design,
- to break the ice for the concepts that will be generalised during the next days.

You can find the document associated to this slides http://cern.ch/go/7pKL.

References I

ANNALS OF PHYSICS: **3,** 1–48 (1958)

Theory of the Alternating-Gradient Synchrotron*†

E. D. COURANT AND H. S. SNYDER

Brookhaven National Laboratory, Upton, New York

The equations of motion of the particles in a synchrotron in which the field gradient index

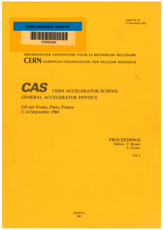
$$n = -(r/B)\partial B/\partial r$$

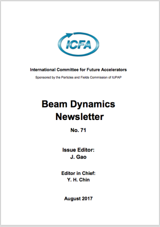
varies along the equilibrium orbit are examined on the basis of the linear approximation. It is shown that if n alternates rapidly between large positive and large negative values, the stability of both radial and vertical oscillations can be greatly increased compared to conventional accelerators in which n is azimuthally constant and must lie between 0 and 1. Thus aperture requirements are reduced. For practical designs, the improvement is limited by the effects of constructional errors; these lead to resonance excitation of oscillations and consequent instability if $2\nu_x$ or $2\nu_z$ or $\nu_x + \nu_z$ is integral, where ν_x and ν_z are the frequencies of horizontal and vertical betatron oscillations, measured in units of the frequency of revolution.

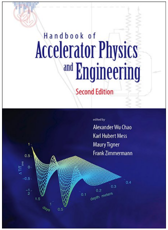
61-years anniversary of the seminal paper of linear optics.

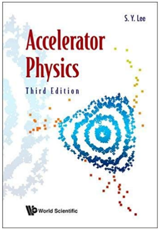
References II

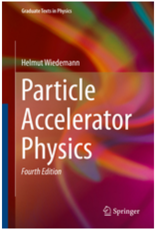
A list[1] of books presenting Linear Optics (and much more).

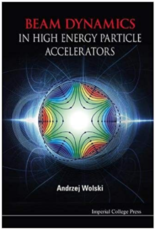








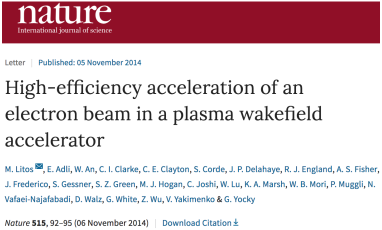




[1] Very incomplete! Apologies for the omissions.

Alternating-gradient as Beam Dynamics foundations

The alternating-gradient was a breakthrough in the history of accelerators based on linear algebra! It is still the very first step for any new technology,

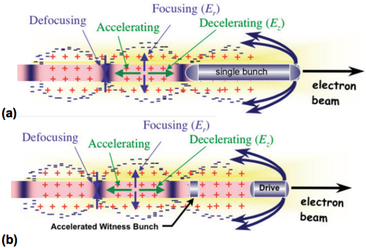

nature
International journal of science

Letter | Published: 05 November 2014

High-efficiency acceleration of an electron beam in a plasma wakefield accelerator

M. Litos, E. Adli, W. An, C. I. Clarke, C. E. Clayton, S. Corde, J. P. Delahaye, R. J. England, A. S. Fisher, J. Frederico, S. Gessner, S. Z. Green, M. J. Hogan, C. Joshi, W. Lu, K. A. Marsh, W. B. Mori, P. Muggli, N. Vafaei-Najafabadi, D. Walz, G. White, Z. Wu, V. Yakimenko & G. Yocky

Nature **515**, 92–95 (06 November 2014) | Download Citation



and for facing the non-linear problems that you will discuss during the following lectures and your professional life.

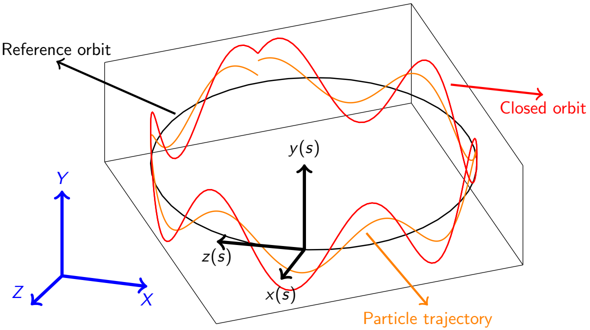
The three ways

One can consider three typical approaches to introduce the linear optics:

- solving the equation of motion (the historical one),
- using Hamiltonian formalism (opening the horizon to the non-linear optics, see later Lectures),
- using the linear matrices (natural choice for the linear optics design).

Our reference system I

To describe the motion of a particle in an optics channel, as usual, we fix a coordinate system to define the status of the particle at a given instant t_1 and a set of laws to transform the coordinates of the system from t_1 to a new instant t_2.



Coordinates

- It is convenient to define the motion along a reference trajectory of the 3D phase space (reference particle trajectory/orbit), so to take into account only the variations along that trajectory (Frenet-Serret frame).
- In addition, it is convenient to replace as independent variable the time, t, with the longitudinal position, s, along the reference trajectory/orbit.
- The natural choice for the variables are $(x, \frac{p_x}{p_0}, y, \frac{p_y}{p_0}, z, \frac{p_z}{p_0})$ (phase-space, see Hamiltonian approach). p_0 is the amplitude of the reference particle momentum.
- Assuming $p_s \approx p_0$ one can consider also the trace-space $(x, x' = \frac{dx}{ds}, y, y' = \frac{dy}{ds}, z, \frac{\Delta p}{p_0})$ (see equation of motion approach).

Linear transformations

We have established phase space $(x, \frac{p_x}{p_0}, y, \frac{p_y}{p_0}, z, \frac{p_z}{p_0})$, now we need to study the particle evolution in there. We assume linear transformation. A system is linear IFF the evolution from the coordinates U to V can be expressed as

$$V = M\ U$$

where M is a square matrix and does not depend on U.

BUT we are interested only on a special set of linear transformation: the so called symplectic linear transformations, that is the ones associated to a simplectic matrix.

Bi-linear transformations

To introduce symplectic matrix we need a short digression on bi-linear transformations.

Let us define the bi-linear transformation F as

$$V^T\ F\ U. \tag{1}$$

This is a function of two vectors (e.g. U and V).

Let consider, for simplicity, the 1D case, that is, $U = (u_a, u_b)^T$ and $V = (v_a, v_b)^T$.

EXAMPLE: orthogonal matrix

Assuming

$$F = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{2}$$

the bilinear transformation I is the dot-product between $V = (v_a, v_b)^T$ and $U = (u_a, u_b)^T$:

$$V^T \underbrace{I}_{F} U = v_a u_a + v_b u_b.$$

A matrix M preserves the bi-linear transformation I (then the projections) IFF

$$\underbrace{V^T M^T}_{(M\ V)^T} I\ M\ U = V^T\ I\ U \rightarrow M^T\ I\ M = I,$$

then M is called orthogonal matrix.

EXAMPLE: symplectic matrix

Assuming

$$F = \boxed{\Omega = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}},$$

the bi-linear transformation Ω is proportional to the amplitude of the cross-product between $V = (v_a, v_b)^T$ and $U = (u_a, u_b)^T$:

$$V^T \underbrace{\Omega}_{F} U = v_a u_b - v_b u_a.$$

that is proportional to the area defined by the vectors. A matrix M preserves the bi-linear transformation Ω (related to the cross-product) IFF

$$V^T M^T \Omega M U = V^T \Omega U \rightarrow \boxed{M^T \Omega M = \Omega},$$

then M is called symplectic matrix.

EXAMPLE: visualise transformations.

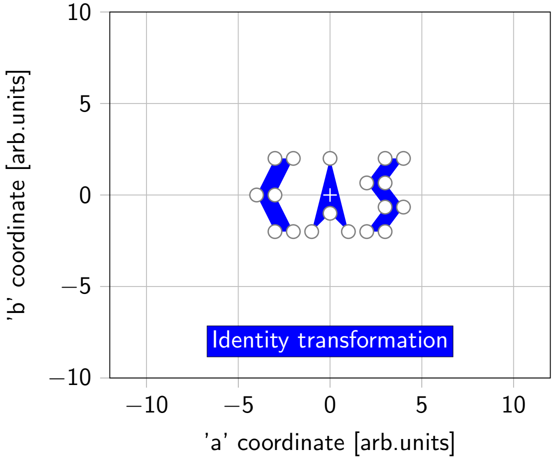


Figure 1: Identity transformation.

EXAMPLE: visualise transformations.

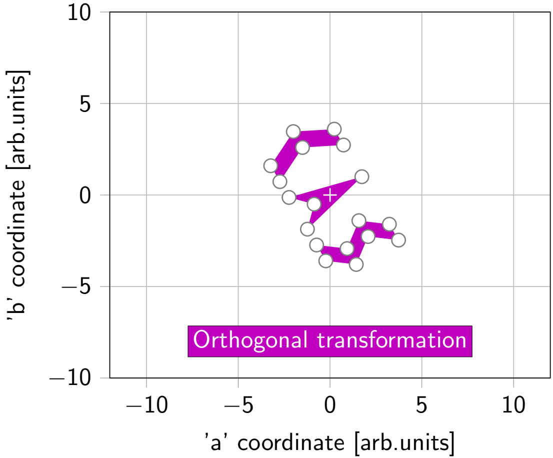


Figure 2: Orthogonal transformation (**dot-product preserved**).

EXAMPLE: visualise transformations.

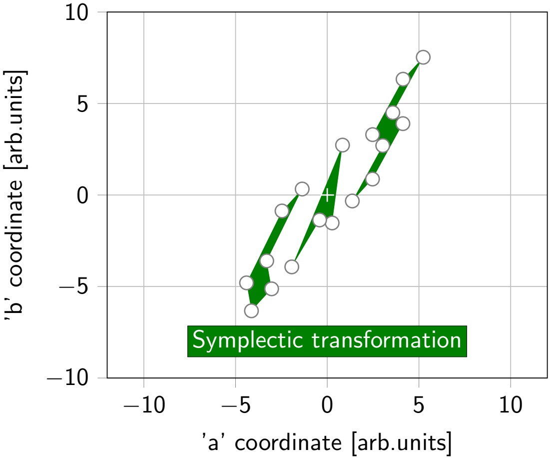


Figure 3: Symplectic transformation (**cross-product preserved**).

Matrix symplecticity in nD

From 1D this can generalized to nD and Ω becomes a $2n \times 2n$ matrix:

$$\Omega = \begin{pmatrix} \begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix} & & 0 \\ & \ddots & \\ 0 & & \begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix} \end{pmatrix}. \tag{3}$$

Example of 2D symplectic matrix:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & \color{red}{1} & 0 \\ 0 & 0 & 1 & 0 \\ \color{red}{1} & 0 & 0 & 1 \end{pmatrix}.$$

Properties of symplectic matrices

- If M_1 and M_2 then $M = M_1 M_2$ is symplectic too.
- If M is symplectic then M^T is symplectic.
- Every symplectic matrix is invertible

$$M^{-1} = \Omega^{-1} M^T \Omega \tag{4}$$

  and M^{-1} is symplectic.
- A necessary condition for M to be symplectic is that $det(M) = +1$. This condition is necessary and sufficient for the 1D case. We will consider 1D case.
- There are symplectic matrices that are defective, that is it cannot be diagonalized, e.g., $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$.

Domino effect

Symplectic matrix and accelerators

Please have a look on this generating set of the symplectic group

$$\underbrace{\begin{pmatrix} G & 0 \\ 0 & \frac{1}{G} \end{pmatrix}}_{\text{thin telescope}}, \underbrace{\begin{pmatrix} 1 & L \\ 0 & 1 \end{pmatrix}}_{\text{drift}}, \underbrace{\begin{pmatrix} 1 & 0 \\ -\frac{1}{f} & 1 \end{pmatrix}}_{\text{thin quad}}.$$

Among the above matrices you can recognise the one of a L-long drift and thin quadrupole with focal length f.

Conveniently combining drifts and thin quadrupole one can find back the well known matrices for the thick elements.

EXAMPLE: a thick quadrupole I

One can derive the transfer matrix of a thick quadrupole of length L by and normalized gradient K_1 by considering the following limit

$$\lim_{n}\left[\begin{pmatrix} 1 & 0 \\ -\frac{K1\ L}{n} & 1 \end{pmatrix}\begin{pmatrix} 1 & \frac{L}{n} \\ 0 & 1 \end{pmatrix}\right]^{n} =$$
$$\begin{pmatrix} \cos\left(\sqrt{\mathrm{K1}}L\right) & \frac{\sin\left(\sqrt{\mathrm{K1}}L\right)}{\sqrt{\mathrm{K1}}} \\ -\sqrt{\mathrm{K1}}\sin\left(\sqrt{\mathrm{K1}}L\right) & \cos\left(\sqrt{\mathrm{K1}}L\right) \end{pmatrix}$$

Therefore we now have a correspondence between elements along our machine (drift, bending, quadrupoles, solenoids,. . .) and symplectic matrices.

EXAMPLE: a thick quadrupole II

To compute the above limit and, in general, for symbolic computations one can profit of the available symbolic computation tools (e.g., Mathematica™).

Code

```
MD[L_] = {{1, L}, {0, 1}}
{{1, L}, {0, 1}}

MQ[KL_] = {{1, 0}, {-KL, 1}}
{{1, 0}, {-KL, 1}}

FullSimplify[Limit[MatrixPower[MQ[K1 L / n].MD[L / n], n], n -> ∞, Assumptions → {K1 > 0, L > 0}]]
```

$$\left\{\left\{\mathrm{Cos}\left[\sqrt{K1}\ L\right],\ \frac{\mathrm{Sin}\left[\sqrt{K1}\ L\right]}{\sqrt{K1}}\right\},\ \left\{-\sqrt{K1}\ \mathrm{Sin}\left[\sqrt{K1}\ L\right],\ \mathrm{Cos}\left[\sqrt{K1}\ L\right]\right\}\right\}$$

```
FullSimplify[Limit[MatrixPower[MQ[-K1 L / n].MD[L / n], n], n -> ∞, Assumptions → {K1 > 0, L > 0}]]
```

$$\left\{\left\{\mathrm{Cosh}\left[\sqrt{K1}\ L\right],\ \frac{\mathrm{Sinh}\left[\sqrt{K1}\ L\right]}{\sqrt{K1}}\right\},\ \left\{\sqrt{K1}\ \mathrm{Sinh}\left[\sqrt{K1}\ L\right],\ \mathrm{Cosh}\left[\sqrt{K1}\ L\right]\right\}\right\}$$

Tracking in a linear system

Given a sequence of elements $M_1, M_2, \ldots M_k$ (the lattice), the evolution of the coordinate, X_n, along the lattice for a given particle can be obtained as

$$X_n = M_n \ldots M_1 \; X_0 \; \mathit{for} \;\; n \geq 1. \tag{5}$$

The transport of the particle along the lattice is called tracking. The tracking on a linear system is trivial and boring. . .

In the following we will decompose the trajectory of the single particle in term of invariant of the motion and properties of the lattice, and via those properties we will describe the statistical evolution of an ensemble of particles.

So instead of tracking an ensemble we will concentrate to solve the properties of the lattice.

Starting a long journey. . .

Voyager 1 is the Man-built object farther away from Earth
$\approx$ 20 light-hours.

Periodic lattice and stability I

We study now the motion of the particles in periodic lattice, that is lattice constituted by a indefinite repetition of the same basic C-long period M_{OTM}, the so-called One-Turn-Map:

$$M_{OTM}(s_0) = M_{OTM}(s_0 + C).$$

From Eq. 5 we get

$$X_n = M_{OTM}^n \, X_0$$

and we study the property of M_{OTM} to have stable motion in the lattice, that is

$$|X_n| < |\hat{X}| \textit{ for all } X_0 \textit{ and } n.$$

In other words, we need to study the if all the elements of the M_{OTM}^n stay bounded.

Periodic lattice and stability II

If M_{OTM} can be expressed as a Diagonal-factorization

$$M_{OTM} = P \underbrace{\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}}_{\text{D}} P^{-1},$$

after m-turns, it yields that

$$M_{OTM}^m = \underbrace{PDP^{-1}}_{1} \times \underbrace{PDP^{-1}}_{2} \times \cdots \times \underbrace{PDP^{-1}}_{m} = PD^m P^{-1}.$$

Therefore the stability depends only on the eigenvalues of M_{OTM}.

Note that the if V is an eigenvector also $kV,\ k \neq 0$ is an eigenvector. Therefore P is not uniquely defined: we chose it such that $det(P) = -i$ and $P_{11} = P_{12}$.

Periodic lattice and stability III

- For a real matrix the eigenvalues, if complex, appear in complex conjugate pairs.
- For a symplectic matrix M_{OTM}

$$\prod_{i}^{2n} \lambda_i = 1$$

where λ_i are the eigenvalues of M_{OTM}.

- Therefore for 2x2 symplectic matrix the eigenvalues can be written as $\lambda_1 = e^{i\mu}$ and $\lambda_2 = e^{-i\mu} \rightarrow D^m = D(m\mu)$.

If μ is real then the motion is stable we can define the fractional tune of the periodic lattice as $\frac{\mu}{2\pi}$.

R-factorization of the M_{OTM} I

The Diagonal-factorization is convenient to check the stability but not to visualize the turn-by-turn phase space evolution of the particle. To do that it is convenient to consider the Rotation-factorization

$$\boxed{M_{OTM} = \bar{P} \underbrace{\begin{pmatrix} \cos\mu & \sin\mu \\ -\sin\mu & \cos\mu \end{pmatrix}}_{\text{R}(\mu)\text{ is orthogonal}} \bar{P}^{-1}}. \tag{6}$$

This is very important since implies that the M_{OTM} is similar to a rotation in phase space (see Yannis and Etienne's lectures).

R-factorization of the M_{OTM} II

To go from Diagonal to Rotation-factorization we note that

$$\underbrace{\begin{pmatrix} \cos\mu & \sin\mu \\ -\sin\mu & \cos\mu \end{pmatrix}}_{R(\mu)} = \underbrace{\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} & -\frac{i}{\sqrt{2}} \end{pmatrix}}_{S^{-1}} \underbrace{\begin{pmatrix} e^{i\mu} & 0 \\ 0 & e^{-i\mu} \end{pmatrix}}_{D(\mu)} \underbrace{\begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{i}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}} \end{pmatrix}}_{S}$$

and therefore

$$R^m = R(m\mu),$$

$$M_{OTM} = \underbrace{P\ S}_{\bar{P}}\ \underbrace{S^{-1}\ D\ S}_{R}\ \underbrace{S^{-1}\ P^{-1}}_{\bar{P}^{-1}}$$

We note that $\det(\bar{P}) = 1$.

Twiss-factorization of M_{OTM} I

We note that

$$R(\mu) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cos\mu + \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \sin\mu,$$

yielding the, so called, Twiss-factorization

$$\boxed{M_{OTM} = \underbrace{\bar{P} I \bar{P}^{-1}}_{I} \cos\mu + \underbrace{\bar{P}\Omega\bar{P}^{-1}}_{J} \sin\mu}$$

Where J has three properties: $\det(J) = 1$, $J_{11} = -J_{22}$, $J_{12} > 0$.

Code: J properties

```
Omega = {{0, 1}, {-1, 0}};
Pbar = {{m11, m12}, {m21, m22}};
Pbar.Omega.Inverse[Pbar] /. {-m12 m21 + m11 m22 -> 1}
{{-m11 m21 - m12 m22, m11^2 + m12^2}, {-m21^2 - m22^2, m11 m21 + m12 m22}}
```

Twiss-factorization of M_{OTM} II

Therefore the following parametric expression has been proposed

$$J = \begin{pmatrix} \alpha & \overbrace{\beta}^{>0} \\ \underbrace{-\frac{1+\alpha^2}{\beta}}_{\gamma>0} & -\alpha \end{pmatrix}$$

defining the Twiss parameters of the lattice at the start of the sequence M_{OTM}. It is very important to not that they are not depending on m since

$$M_{OTM}^m = I\cos(m\mu) + J\sin(m\mu)$$

In other words the Twiss parameters are periodic (compare to Floquet theorem).

Twiss-factorization of M_{OTM} III

From the definition of J follows, $J = \bar{P}\Omega\bar{P}^{-1}$, the one of

$$\bar{P} = \begin{pmatrix} \sqrt{\beta} & 0 \\ -\frac{\alpha}{\sqrt{\beta}} & \frac{1}{\sqrt{\beta}} \end{pmatrix} = \begin{pmatrix} \sqrt{\beta} & 0 \\ 0 & \frac{1}{\sqrt{\beta}} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -\frac{\alpha}{\sqrt{\beta}} & 1 \end{pmatrix}$$

We note that by choosing $\det P = -i$ we got $\det \bar{P} = 1$ that is we expressed M as the product of orthogonal and symplectic matrices.

and

$$\boxed{P = \bar{P}S^{-1} = \begin{pmatrix} \sqrt{\frac{\beta}{2}} & \sqrt{\frac{\beta}{2}} \\ \frac{-\alpha+i}{\sqrt{2\beta}} & \frac{-\alpha-i}{\sqrt{2\beta}} \end{pmatrix}}.$$

Where do we stand?

Given a symplectic $M_{OTM}(s)$, if diagonalizable, we can study three equivalent periodic problems

- $M_{OTM}(s)^m = P\ D(m\mu)\ P^{-1}$,
- $M_{OTM}(s)^m = \bar{P}\ R(m\mu)\ \bar{P}^{-1}$,
- $M_{OTM}(s)^m = I\ \cos(m\mu) + J\ \sin(m\mu)$.

The previous factorizations allow us to reduce the power of a matrix to an algebric multiplication $(m\mu)$. We expressed P, $\bar{P}$ and J as function of β and α parameters.

$\rightarrow$ IMPORTANT FOR LATTICE STABILITY $\leftarrow$

Code

From $M_{OTM}(s)$ compute D (check stability) and P (force $det(P) = -i$, $m_{11} = m_{12}$), then $\bar{P} = PS$ and $J = \bar{P}\Omega\bar{P}^{-1}$. You therefore get the fractional tune and the Twiss parameters at s_0.

$M_{OTM}(s_0)$ and $M_{OTM}(s_1)$

$M_{OTM}(s)$ is a function of s: are μ, β and α all s-function?

Given a C-long periodic lattice and two longitudinal positions s_0 and s_1 ($s_1 > s_0$), the transformation from s_0 to $s_1 + C$ can be expressed as

$$s_0 \longrightarrow s_1 \longrightarrow s_1 + C$$

$$s_0 \longrightarrow s_0 + C \longrightarrow s_1 + C$$

$$M_{OTM}(s_1)\ M = M\ M_{OTM}(s_0)$$

where M is the transformation from s_0 to s_1. This implies

$$M_{OTM}(s_1) = M\ M_{OTM}(s_0)\ M^{-1}$$

$\rightarrow$ the matrices $M_{OTM}(s_1)$ and $M_{OTM}(s_2)$ are similar.

$\rightarrow$ same eigenvalues: the M_{OTM} is s-dependent but the Q is not.

β and α transport I

On the other hand we observe that β and α are s-dependent function and we have:

$$M_{OTM}(s_1) = M\ M_{OTM}(s_0)\ M^{-1} = M\ (I\cos\mu + J(s_0)\sin\mu)\ M^{-1},$$

therefore

$$\underbrace{\begin{pmatrix} \alpha(s_1) & \beta(s_1) \\ -\gamma(s_1) & -\alpha(s_1) \end{pmatrix}}_{J(s_1)} = M \underbrace{\begin{pmatrix} \alpha(s_0) & \beta(s_0) \\ -\gamma(s_0) & -\alpha(s_0) \end{pmatrix}}_{J(s_0)}\ M^{-1}.$$

β and α transport II

To simplify from a computational point of view the Eq. 7 we can use the Eq. 4 (inverse of a symplectic matrix) and this yields

$$\begin{pmatrix} \alpha(s_1) & \beta(s_1) \\ -\gamma(s_1) & -\alpha(s_1) \end{pmatrix} \Omega^{-1} = M \begin{pmatrix} \alpha(s_0) & \beta(s_0) \\ -\gamma(s_0) & -\alpha(s_0) \end{pmatrix} \Omega^{-1} M^T,$$

that is

$$\underbrace{\begin{pmatrix} \beta(s_1) & -\alpha(s_1) \\ -\alpha(s_1) & \gamma(s_1) \end{pmatrix}}_{J(s_1)\ \Omega^{-1}} = M \underbrace{\begin{pmatrix} \beta(s_0) & -\alpha(s_0) \\ -\alpha(s_0) & \gamma(s_0) \end{pmatrix}}_{J(s_0)\ \Omega^{-1}} M^T. \qquad (7)$$

EXAMPLE: the β-function in a drift

To compute the Twiss parameters in a drift we can simply apply the previous equation

$$\begin{pmatrix} \beta(s) & -\alpha(s) \\ -\alpha(s) & \gamma(s) \end{pmatrix} = \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \beta_0 & -\alpha_0 \\ -\alpha_0 & \gamma_0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ s & 1 \end{pmatrix}$$

yielding

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

and

$$\alpha(s) = \alpha_0 - \gamma_0 s.$$

$\rightarrow$ IMPORTANT FOR INSERTIONS $\leftarrow$

The differential relation between α and β I

In order to see differential relation with the matrix formalism we consider the general ΔM matrix for the infinitesimal offset, Δs,

$$\Delta M = \begin{pmatrix} 1 & \Delta s \\ -K(s)\Delta s & 1 \end{pmatrix}.$$

Note that ΔM is symplectic only for $\Delta s \to 0$.

Then we have

$$\underbrace{\begin{pmatrix} \beta(s+\Delta s) & -\alpha(s+\Delta s) \\ -\alpha(s+\Delta s) & \gamma(s+\Delta s) \end{pmatrix}}_{J(s+\Delta s)\Omega^{-1}} = \Delta M \underbrace{\begin{pmatrix} \beta(s) & -\alpha(s) \\ -\alpha(s) & \gamma(s) \end{pmatrix}}_{J(s)\Omega^{-1}} \Delta M^T.$$

The differential relation between α and β II

From that we have that

$$\lim_{\Delta s \to 0} \frac{J(s+\Delta s) - J(s)}{\Delta s}\Omega^{-1} = \begin{pmatrix} \beta'(s) & -\alpha'(s) \\ -\alpha'(s) & \gamma'(s) \end{pmatrix}$$

where we used standard notation $\frac{d\cdot}{ds} = \cdot'$. One gets

$$\begin{aligned} \beta'(s) &= -2\alpha(s) \\ \alpha'(s) &= -\gamma + K(s)\beta(s). \end{aligned}$$

Replacing α and γ in the latter equation with functions of β we get the non-linear differential equation:

$$\boxed{\frac{\beta''\beta}{2} - \frac{\beta'^2}{4} + K\beta^2 = 1}.$$

EXAMPLE: from matrices to Hill's equation

Following the notation already introduced

$$X(s+\Delta s) = \Delta M\ X(s)$$

with $X(s) = (x(s), \frac{p_x(s)}{p_0})^T \underset{p_0 \approx p_s}{\approx} (x(s), x'(s))^T$, therefore

$$X'(s) = \begin{pmatrix} x'(s) \\ x''(s) \end{pmatrix} = \lim_{\Delta s \to 0} \frac{X(s+\Delta s) - X(s)}{\Delta s} = \begin{pmatrix} x'(s) \\ -K(s)x(s) \end{pmatrix}$$

we find back the Hill's equation

$$\boxed{x''(s) + K(s)x(s) = 0}.$$

Where do we stand?

- We learnt how to propagate via linear matrices the initial Twiss parameters along the machine.
- We also retrieved several differential relations between α and β, β and K, and X and K: these are, in general, not practical for computations.
- The next question is, moving from the lattice to the particle, is there an invariant of the motion?

Courant-Snyder invariant I

Given a particle with coordinate X we can observe that the quantity

$$\boxed{X^T \Omega\, J^{-1}\, X}$$

is an invariant of the motion: it is called the Courant-Snyder invariant, J_{CS}. In fact from Eq. 7

$$X_1^T \Omega\, J_1^{-1}\, X_1 = X_0^T M^T (M\, J_0 \Omega^{-1}\, M^T)^{-1} M\, X_0 = X_0^T \Omega\, J_0^{-1}\, X_0$$

Code: find back the CS invariant in the trace-space

```
J = {{α, β}, {-γ, -α}};
FullSimplify[{{x, x'}}.{{0, 1}, {-1, 0}}.Inverse[J].{{x}, {x'}}] /. βγ - α^2 → 1
{{x^2 γ + 2 x α x' + β (x')^2}}
```

Courant-Snyder invariant II

In the normalized phase-space, remembering that $X = \bar{P}\ \tilde{X}$, we have

$$X^T\Omega\ J^{-1}\ X = \tilde{X}^T \underbrace{\bar{P}^T\Omega\ J^{-1}\bar{P}}_{I}\ \tilde{X} = \tilde{X}^T\ \tilde{X}$$

that is the J_{CS} is the square of the circle radius defined by the particle initial condition.

This normalized phase-space is also called action-angle phase space. The particle action is defined as $J_{CS}/2$.

What about the phase $\mu(s)$? I

What is the $\Delta\mu$ introduced by a linear matrix $M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}$?

In normalized space the transport from s to $s + \Delta s$ does not change J_{CS} but the angle by $\Delta\mu = \mu(s + \Delta s) - \mu(s)$.

To compute it we move to the normalized phase-space

$$X(s) = \bar{P}(s)\ \tilde{X}(s) \text{ and } X(s + \Delta s) = \bar{P}(s + \Delta s)\ \tilde{X}(s)$$

and from

$$X(s + \Delta s) = M\ X(s),$$

it yields

$$\tilde{X}(s{+}\Delta s) = \bar{P}(s{+}\Delta s)^{-1}\ M\ \bar{P}(s)\tilde{X}(s) = \begin{pmatrix} \cos\Delta\mu & \sin\Delta\mu \\ -\sin\Delta\mu & \cos\Delta\mu \end{pmatrix}\ \tilde{X}(s).$$

What about the phase $\mu(s)$? II

That is

$$\tan \Delta\mu = \underbrace{\frac{\sin \Delta\mu}{\cos \Delta\mu}}_{\text{It does depend only on } \beta \text{ and } \alpha \text{ in } s!} = \frac{m_{12}}{m_{11}\ \beta(s) - m_{12}\ \alpha(s)}.$$

Code: derivation of $\Delta\mu$

```
Pbar0 = {{√β0, 0}, {-α0/√β0, 1/√β0}};
Pbar1 = {{√β1, 0}, {-α1/√β1, 1/√β1}};
M = {{m11, m12}, {m21, m22}};
FullSimplify[Inverse[Pbar1].M.Pbar0]
```

$$\left\{\left\{\frac{-\mathrm{m12}\ \alpha0 + \mathrm{m11}\ \beta0}{\sqrt{\beta0}\ \sqrt{\beta1}}, \frac{\mathrm{m12}}{\sqrt{\beta0}\ \sqrt{\beta1}}\right\}, \left\{\frac{-\mathrm{m12}\ \alpha0\ \alpha1 + \mathrm{m11}\ \alpha1\ \beta0 - \mathrm{m22}\ \alpha0\ \beta1 + \mathrm{m21}\ \beta0\ \beta1}{\sqrt{\beta0}\ \sqrt{\beta1}}, \frac{\mathrm{m12}\ \alpha1 + \mathrm{m22}\ \beta1}{\sqrt{\beta0}\ \sqrt{\beta1}}\right\}\right\}$$

EXAMPLE 1: $\mu(s)$ differential equation

If $M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} = \Delta M = \begin{pmatrix} 1 & \Delta s \\ -K(s)\Delta s & 1 \end{pmatrix}$ then one gets

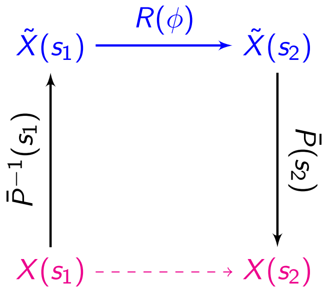
$$\mu' = \lim_{\Delta s \to 0} \frac{\tan \Delta\mu}{\Delta s} = \lim_{\Delta s \to 0} \frac{1}{\beta(s) - \alpha(s)\ \Delta s} = \frac{1}{\beta(s)},$$

that is the well know expression

$$\boxed{\mu(s) = \int_{s_0}^{s} \frac{1}{\beta(\sigma)} d\sigma + \mu(s_0)}.$$

EXAMPLE 2: Betatron oscillation I

How we describe a betatronic oscillation from s_1 to s_2 in terms of Twiss parameters and initial conditions?



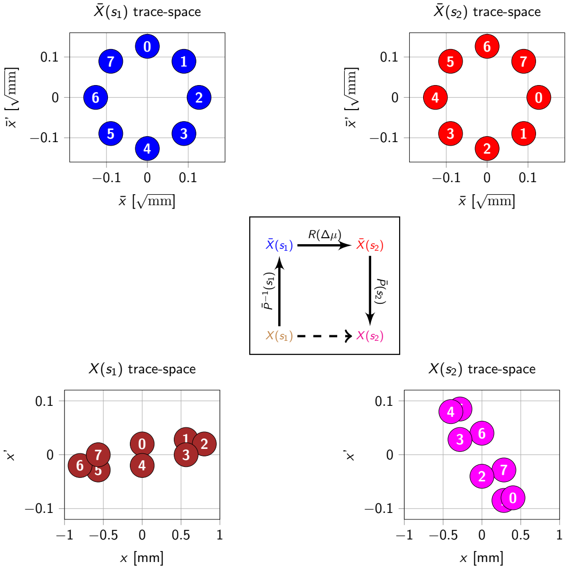
It is easy by transforming the vector X in the normalized phase space in s_1, moving it from s_1 to s_2 in the normalized space (pure rotation of the phase ϕ) and back transform it in the original phase space.

EXAMPLE 2: Betatron oscillation II

Code

```
Pbar1 = {{√β1, 0}, {-α1/√β1, 1/√β1}};

Pbar2 = {{√β2, 0}, {-α2/√β2, 1/√β2}};

R = {{Cos[ϕ], Sin[ϕ]}, {-Sin[ϕ], Cos[ϕ]}};

FullSimplify[Pbar2.R.Inverse[Pbar1]]

{{√β2 (Cos[ϕ] + α1 Sin[ϕ])/√β1, √β1 √β2 Sin[ϕ]}, {-((-α1 + α2) Cos[ϕ] + Sin[ϕ] + α1 α2 Sin[ϕ])/(√β1 √β2), √β1 (Cos[ϕ] - α2 Sin[ϕ])/√β2}}
```

$$
\begin{aligned}
M &= \bar{P}(s_2)\ R(\phi)\bar{P}(s_1)^{-1} = \\
&= \begin{pmatrix} \sqrt{\frac{\beta_2}{\beta_1}}(\cos\phi + \alpha_1 \sin\phi) & \sqrt{\beta_1\beta_2}\sin\phi \\ \frac{\alpha_1-\alpha_2}{\sqrt{\beta_1\beta_2}}\cos\phi - \frac{1+\alpha_1\alpha_2}{\sqrt{\beta_1\beta_2}}\sin\phi & \sqrt{\frac{\beta_1}{\beta_2}}(\cos\phi - \alpha_2\sin\phi) \end{pmatrix}
\end{aligned}
$$


$\bar{X}(s_1)$ trace-space
$\bar{X}(s_2)$ trace-space
$\bar{x}'$ [$\sqrt{\text{mm}}$]
$\bar{x}$ [$\sqrt{\text{mm}}$]
$R(\Delta\mu)$
$\bar{X}(s_1)$
$\bar{X}(s_2)$
$\bar{P}^{-1}(s_1)$
$\bar{P}(s_2)$
$X(s_1)$
$X(s_2)$
$X(s_1)$ trace-space
$X(s_2)$ trace-space
x'
x [mm]

EXAMPLE 3: Solution of Hill's equation

How we describe a betatronic oscillation in machine considering a J_{CS} and phase μ_0? This is a special case of the previous one. With the J_{CS} and phase μ_0 we are already in the normalized phase space, therefore we need only to rotate by $\mu(s)$ and back transform it in the original phase space.

$$\begin{aligned} X(s) &= \bar{P}(s) \begin{pmatrix} \sqrt{J_{CS}} \cos(\mu + \mu_0) \\ -\sqrt{J_{CS}} \sin(\mu + \mu_0) \end{pmatrix} = \\ &= \begin{pmatrix} \sqrt{J_{CS} \beta(s)} \cos(\mu + \mu_0) \\ -\sqrt{\frac{J_{CS}}{\beta(s)}} [\alpha(s) \cos(\mu + \mu_0) + \sin(\mu + \mu_0)] \end{pmatrix} \end{aligned}$$

where one recognizes the solutions of the Hill's equation.

Computing the closed orbit

Up to now we assumed that the closed orbit (CO) corresponded to the reference orbit. This is not always true.

Assuming a $M_{OTM}(s_0)$ and a single thin kick Θ at s_0 (independent from X_n) we can write

$$X_{n+1}(s_0) = M_{OTM}(s_0)\ X_n(s_0) + \Theta.$$

In the 1D case Θ can represent a kick of a dipole correction or misalignment of a quadrupole ($\Theta = (0, \theta)^T$). The closed orbit solution can be retrieved imposing $V_{n+1} = V_n$ (fixed point), yielding

$$X_n(s_0) = (I - M_{OTM}(s_0))^{-1}\Theta(s_0).$$

Please note that the CO is discontinuous at s_0 so the previous formula refers to the CO after the kick. In presence of multiple $\Theta(s_i)$ one can sum the single contributions along s.

EXAMPLE: from the CO matrix to the CO formula

Code: closed orbit formula

```
J = {{α1, β1}, {-(1 + α1^2)/β1, -α1}};

MCO = FullSimplify[Inverse[IdentityMatrix[2] - (IdentityMatrix[2] Cos[2 π Q] + J Sin[2 π Q])]]
{{1/2 (1 + α1 Cot[π Q]), 1/2 β1 Cot[π Q]}, {-((1 + α1^2) Cot[π Q])/(2 β1), 1/2 (1 - α1 Cot[π Q])}}

x0 = FullSimplify[MCO.{{0}, {θ1}}]
{{1/2 β1 θ1 Cot[π Q]}, {1/2 (θ1 - α1 θ1 Cot[π Q])}}

Transport = {{(Sqrt[β2] (Cos[ϕ] + α1 Sin[ϕ]))/Sqrt[β1], Sqrt[β1] Sqrt[β2] Sin[ϕ]}, {-((-α1 + α2) Cos[ϕ] + Sin[ϕ] + α1 α2 Sin[ϕ])/(Sqrt[β1] Sqrt[β2]), (Sqrt[β1] (Cos[ϕ] - α2 Sin[ϕ]))/Sqrt[β2]}};

FullSimplify[Transport.x0]
{{1/2 Sqrt[β1] Sqrt[β2] θ1 (Cos[ϕ] Cot[π Q] + Sin[ϕ])}, {-(Sqrt[β1] θ1 (Cos[ϕ] (-1 + α2 Cot[π Q]) + (α2 + Cot[π Q]) Sin[ϕ]))/(2 Sqrt[β2])}}

TrigReduce[Cos[ϕ] Cot[π Q] + Sin[ϕ]]
Cos[π Q - ϕ] Csc[π Q]
```

We found back the known equation

$$x_{CO}(s) = \frac{\sqrt{\beta(s)\beta(s_0)}}{2\sin(\pi Q)}\theta_{s_0}\cos(\phi - \pi Q) \qquad (8)$$

where ϕ is the phase advance (> 0) from s_0 to s.

Computing dispersion and chromaticity I

Up to now we considered all the optics parameters for the on-momentum particle. To evaluate the off-momentum effect of the closed orbit and the tune we introduce the dispersion, $D_{x,y}(s, \frac{\Delta p}{p_0})$, and chromaticity, $\xi_{x,y}(\frac{\Delta p}{p_0})$, respectively, as

$$\Delta CO_{x,y}(s) = D_{x,y}\left(s, \frac{\Delta p}{p_0}\right) \times \frac{\Delta p}{p_0}, \quad D_{x,y}(s+C) = D(s)$$

and

$$\Delta Q_{x,y} = \xi_{x,y}\left(\frac{\Delta p}{p_0}\right) \times \frac{\Delta p}{p_0}.$$

Computing dispersion and chromaticity II

In order to compute numerically the $D_{x,y}$ and $\xi_{x,y}$ one can compute first the $CO_{x,y}$ and the $Q_{x,y}$ as function of of $\frac{\Delta p}{p_0}$.

To do that one has to compute $M_{OTM}(s, \frac{\Delta p}{p_0})$, that is evaluate the property of the element of the lattice as function of $\frac{\Delta p}{p_0}$.

- In a thin quadrupole the focal length linearly scales with the beam rigidity:

$$\begin{pmatrix} 1 & 0 \\ -\frac{1}{f(\frac{\Delta p}{p_0})} & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 \\ -\frac{1}{f_0 \times (1+\frac{\Delta p}{p_0})} & 1 \end{pmatrix}.$$

- A dipolar kick θ, scales with the inverse of the beam rigidity:

$$\begin{pmatrix} 0 \\ \theta(\frac{\Delta p}{p_0}) \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ \frac{\theta_0}{1+\frac{\Delta p}{p_0}} \end{pmatrix}.$$

Where do we stand?

We learnt how to compute

- the invariant of the motion J_{CS},
- the betatronic phase, $\mu(s)$, along the lattice,
- the CO given a set of kicks,
- the dispersion and chromaticity.

We will consider in the following an ensemble of non-interacting particle and we will introduce the concept of beam emittance and beam matching.

The Beam distribution, a set of N particles

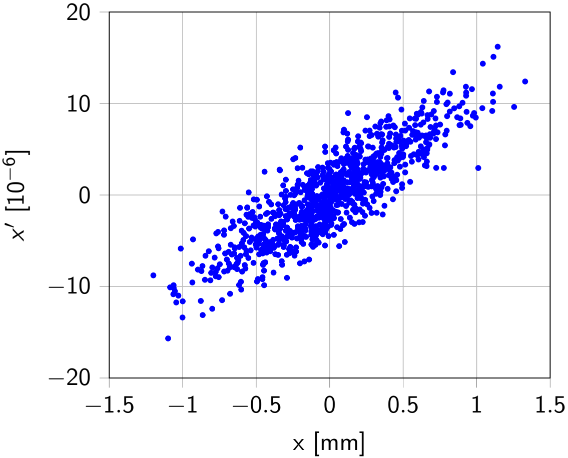


Figure 5: From single particle to particle ensembles.

The Beam distribution, a set of N particles I

To track N particles is possible by using the same approach of the single particle tracking were X becomes X_{Beam}, a $2n \times N$ matrix:

$$X_{Beam} = (X_1, X_2, \ldots, X_n)$$

We will restrict ourself to the 1D case (n=1).
We are looking for one or more statistical quantities that represents this ensemble and its evolution in the lattice.
A natural one is the average J_{CS} over the ensemble:

$$\frac{1}{N}\sum_{i=1}^{N} J_{CS,i} = \langle J_{CS} \rangle$$

From the definition it follows that the quantity is preserved during the beam evolution along the lattice.

Beam emittance

We will see in the hands-on that $\langle J_{CS} \rangle$ converges, under specific assumptions, to twice the rms emittance of the beam, ϵ_{rms}

$$\epsilon_{rms} = \sqrt{\underbrace{\det(\frac{1}{N} X_B X_B^T)}_{\sigma \text{ matrix}}}.$$

One can see that the ϵ_{rms} is preserved for the symplectic linear transformation M from s_0 to s_1 (see Cauchy-Binet theorem):

$$\epsilon_{rms}^2(s_0) = \det(\frac{1}{N} X_B X_B^T)$$

$$\epsilon_{rms}^2(s_1) = \det(M \underbrace{\frac{1}{N} X_B X_B^T}_{\sigma(s_0)} M^T) = \underbrace{\det M}_{=1} \det(\frac{1}{N} X_B X_B^T) \underbrace{\det M^T}_{=1}$$

where X_B denotes $X_B(s_0)$. Note that $\sigma(s_1) = M\ \sigma(s_0)\ M^T$.

The σ matrix

By its definition we have (e.g., 1D trace-space) that

$$\sigma = \begin{pmatrix} \frac{1}{N}\sum_{i=1}^{N} x_i x_i & \frac{1}{N}\sum_{i=1}^{N} x_i x_i' \\ \frac{1}{N}\sum_{i=1}^{N} x_i' x_i & \frac{1}{N}\sum_{i=1}^{N} x_i' x_i' \end{pmatrix} = \begin{pmatrix} \overbrace{\langle \bar{x}^2 \rangle}^{x_{rms}^2} & \langle xx' \rangle \\ \langle xx' \rangle & \underbrace{\langle \bar{x}'^2 \rangle}_{x_{rms}'^2} \end{pmatrix}$$

and therefore we can write

$$\epsilon_{rms} = \sqrt{\langle x^2 \rangle \langle x'^2 \rangle - \langle xx' \rangle^2}.$$

So we show how to numerically transport the second-order moments of the beam distribution.

Matched beam distribution I

A beam distribution is matched to the specific optics functions $\bar{\alpha}$ and $\bar{\beta}$ if the corresponding normalized distribution is statistically invariant by rotation in the normalized space. In other words it has an azimuthal symmetry.

It is worth noting that since $\bar{P}^{-1}$ is a symplectic matrix and defining $\bar{X}_B = \bar{P}^{-1} X_B$ we have that $\bar{\epsilon}_{rms} = \epsilon_{rms}$ and for a matched beam we have

$$\bar{\sigma} = \frac{1}{N} \bar{X}_B \bar{X}_B^T = \bar{P}^{-1} \sigma \; \bar{P} = \begin{pmatrix} \overbrace{\langle \bar{x}^2 \rangle}^{\bar{x}_{rms}^2} & \langle \bar{x}\bar{x}' \rangle \\ \langle \tilde{x}\bar{x}' \rangle & \underbrace{\langle \bar{x}'^2 \rangle}_{\bar{x}'^2_{rms}} \end{pmatrix} = \begin{pmatrix} \epsilon_{rms} & 0 \\ 0 & \epsilon_{rms} \end{pmatrix}.$$

Therefore $\bar{\sigma}$ is diagonal.

Matched beam distribution II

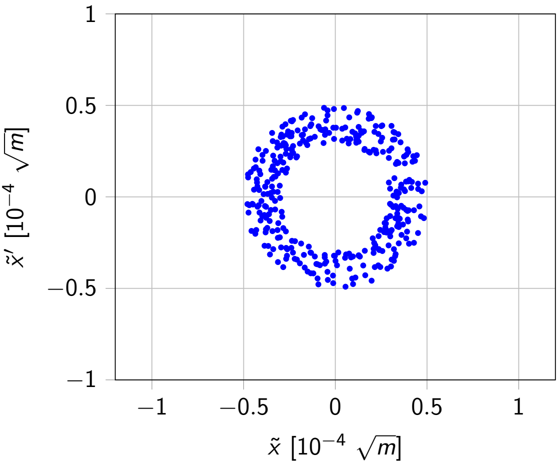


Figure 6: A **matched** beam distribution in normalized trace-space.

Matched beam distribution III

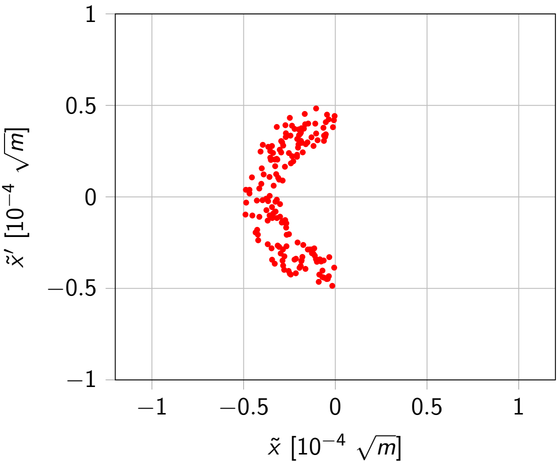


Figure 7: A **mismatched** beam distribution in normalized trace-space.

Matched beam distribution IV

For a beam distribution matched to the specific optics functions $\bar{\alpha}$ and $\bar{\beta}$ the we have

$$\sigma = \bar{P}\bar{\sigma}\ \bar{P}^{-1} = \begin{pmatrix} \bar{\beta}\epsilon_{rms} & -\bar{\alpha}\epsilon_{rms} \\ -\bar{\alpha}\epsilon_{rms} & \bar{\gamma}\epsilon_{rms} \end{pmatrix} \tag{9}$$

where we found back the rms beam size and divergence formulas, $\boxed{\sqrt{\bar{\beta}\epsilon_{rms}}}$ and $\boxed{\sqrt{\bar{\gamma}\epsilon_{rms}}}$, respectively.

The rms size of a matched beam in a periodic stable lattice and at given position s_0 is a turn-by-turn invariant.

About ensembles

- We extended the single particle computation method to transport ensembles of particles.
- We introduced the concepts of beam σ matrix, the ϵ_{rms}, its relation with the $\langle J_{CS} \rangle$ and the concept of beam matching.

MAD-X in 20 min...

DISCLAIMER

- This material is intended to be an short introduction to MAD-X: a large part of the code capabilities are not discussed in details or are not discussed at all!
- Please refer to MAD-X web site http://madx.web.cern.ch/ to learn more.

What is MAD-X?

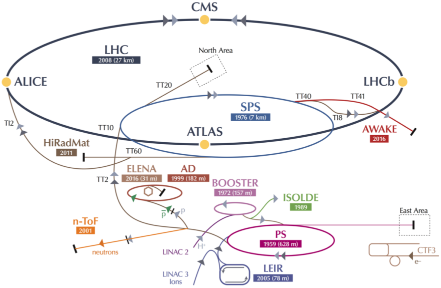
Methodical Accelerator Design version X

- A general purpose (free) beam optics and lattice program.
- It is used since more than 30 years.
- MAD-X is written in C/C++/Fortran77/Fortran90 (source code is available under CERN copyright).

sterbini — -bash — 51×18

```
Last login: Wed Nov  7 10:53:39 on ttys000
macbe16107:~ sterbini$ madx

  ++++++++++++++++++++++++++++++++++++++++++++
  +     MAD-X 5.02.13  (64 bit, Darwin)      +
  + Support: mad@cern.ch, http://cern.ch/mad +
  + Release    date: 2016.12.20              +
  + Execution date: 2018.11.09 16:35:34      +
  ++++++++++++++++++++++++++++++++++++++++++++

X:> quit;

  Number of warnings: 0

  ++++++++++++++++++++++++++++++++++++++++++++
  +          MAD-X finished normally         +
  ++++++++++++++++++++++++++++++++++++++++++++
macbe16107:~ sterbini$ ¢
```

A general purpose beam optics code



For circular machines, beam lines and linacs. . .

- Describe/document parameters from machine description.
- Design a lattice for getting the desired properties (matching).
- Simulate beam dynamics, imperfections and operation.

A general purpose beam optics code

MAD-X is

- multiplatforms (Linux/OSX/WIN...),
- very flexible and easy to extend,
- made for complicated applications, powerful and rather complete,
- mainly designed for large projects (LHC, CLIC, FCC...).

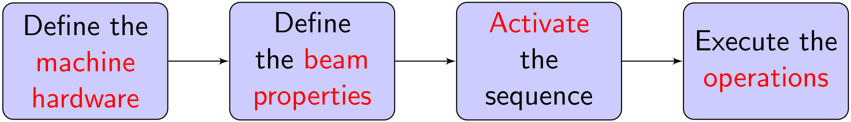
In large projects (e.g., LHC):

- Must be able to handle machines with $\geq\ 10^4$ elements,
- many simultaneous MAD-X users (LHC: more than 400 around the world): need consistent database,
- if you have many machines: ideally use only one design program.

Describe an accelerator in MAD-X

Goals. . .

- Describe, optimize and simulate a machine with several thousand elements eventually with magnetic elements shared by different beams, like in colliders.



MAD-X language

How does MAD-X get this info? Via text (interpreter).

- It accepts and executes statements, expressions. . . ,
- it can be used interactively (input from command line) or in batch (input from file),
- many features of a programming language (loops, if's,. . .).

All input statements are analysed by a parser and checked.

- E.g. assignments: properties of machine elements, set up of the lattice, definition of beam properties, errors. . .
- E.g. actions: compute lattice functions, optimize and correct the machine. . .

MAD-X input language

- Strong resemblance to "C" language (but NO need for declarations and NOT case sensitive apart in expressions in inverted commas),
- free format, all statements are terminated with **;** (do not forget!),
- comment lines start with: **//** or **!** or is between **/***. . . ***/**,
- Arithmetic expressions, including basic functions (**exp**, **log**, **sin**, **cosh**. . .), built-in random number generators and predefined constants (speed of the light, e, π, m_p, m_e. . .).

In particular it is possible to use deferred assignments

- regular assignment: **a = b**, if **b** changes **a** does not,
- deferred assignment: **a := b**, if **b** changes **a** is updated too.

Example: deferred assignments

```
sterbini — -bash — 54×19
  ++++++++++++++++++++++++++++++++++++++++++++
  +     MAD-X 5.02.13  (64 bit, Darwin)      +
  + Support: mad@cern.ch, http://cern.ch/mad +
  + Release   date: 2016.12.20               +
  + Execution date: 2018.11.10 10:16:13      +
  ++++++++++++++++++++++++++++++++++++++++++++

X:> a=1;
X:> b=a;
X:> c:=a;
X:> a=2;
++++++ info: a redefined
X:> value a;
a                  =                  2 ;
X:> value b;
b                  =                  1 ;
X:> value c;
c                  =                  2 ;
X:> quit;
```

We use the **value** command to print the variables content.

Definitions of the lattice elements

Generic pattern to define an element:

label: *keyword*, properties. . . ;

- For a dipole magnet:
  MBL: SBEND, L=10.0;
- For a quadrupole magnet:
  MQ: QUADRUPOLE, L=3.3;
- For a sextupole magnet:
  MSF: SEXTUPOLE, L=1.0;

In the previous examples we considered only the L property, that is the length in meters of the element.

The strength of the elements

The name of the parameter that define the normalized magnetic strength of the element depends on the element type.

- For dipole (horizontal bending) magnet is k_0:

$$k_0 = \frac{1}{B\rho} B_y \ [\mathrm{in\ m}^{-1}]$$

- For quadrupole magnet is k_1:

$$k_1 = \frac{1}{B\rho} \frac{\partial B_y}{\partial x} \ [\mathrm{in\ m}^{-2}]$$

- For sextupole magnet is k_2:

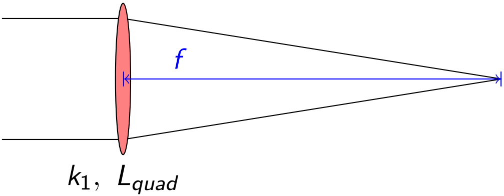
$$k_2 = \frac{1}{B\rho} \frac{\partial^2 B_y}{\partial x^2} \ [\mathrm{in\ m}^{-3}]$$

Interlude

What does k_1 mean? It is related to the quad focal length [2].

$$\frac{1}{k_1 \ L_{quad}} = f \qquad (10)$$

Assuming $k_1 = 10^{-1}$ m^{-2} and $L_{quad} = 10^{-1}$ m the $f = 10^2$ m.



[2] thin lens approximation

Example: definitions of elements

- Kicker magnet:

  theta = 1e-6;

  KICK: HKICKER, L=0, HKICK=theta;

- Multipole magnet "thin" element:

  MMQ: MULTIPOLE, KNL $= \{k0 \cdot l, k1 \cdot l, k2 \cdot l, k3 \cdot l, \ldots\}$;

- LHC dipole magnet as thick element:

  length = 14.3;

  p = 7000;

  angleLHC = 8.33 * clight * length/p;

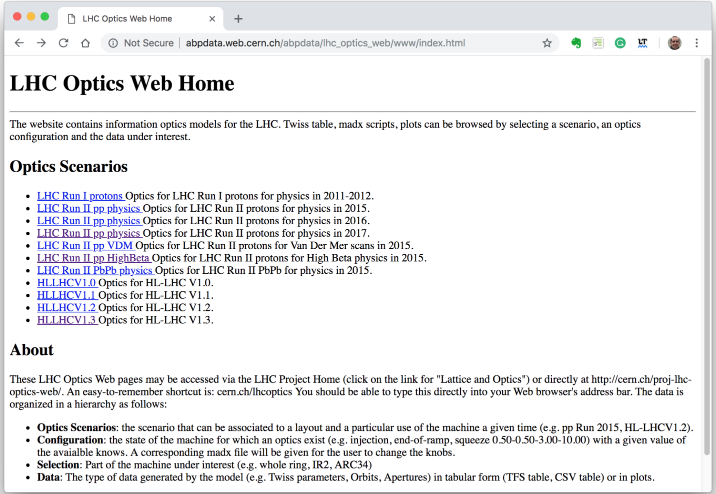
  MBL: SBEND, ANGLE = angleLHC;

The lattice sequence

A lattice sequence is an ordered collection of machine elements.
Each element has a position in the sequence that can be defined wrt the CENTRE, EXIT or ENTRY of the element and wrt the sequence start or the position of an other element:

```
label: SEQUENCE, REFER=CENTRE, L=length;
...;
...;
...here specify position of all elements...;
...;
...;
ENDSEQUENCE;
```

EXAMPLE: www.cern.ch/lhcoptics



EXAMPLE: the LHC sequence

sterbini — sterbini@lxplus101:/eos/user/s/sterbini/_First/2018/LHC MD Optics/injection/db5 — ssh lxplus.cern.ch — 129×39

```
//---------------------- VCORRECTOR     ----------------------------------------------
MCBCV : VCORRECTOR, L := l.MCBCV, Kmax := Kmax_MCBCV, Kmin := Kmin_MCBCV, Calib := Kmax_MCBCV / Imax_MCBCV;
MCBV : VCORRECTOR, L := l.MCBV, Kmax := Kmax_MCBV, Kmin := Kmin_MCBV, Calib := Kmax_MCBV / Imax_MCBV;
MCBWV : VCORRECTOR, L := l.MCBWV, Kmax := Kmax_MCBWV, Kmin := Kmin_MCBWV, Calib := Kmax_MCBWV / Imax_MCBWV;
MCBXV : VCORRECTOR, Lrad := l.MCBXV, Kmax := Kmax_MCBXV, Kmin := Kmin_MCBXV, Calib := Kmax_MCBXV / Imax_MCBXV;
MCBYV : VCORRECTOR, L := l.MCBYV, Calib := Kmax_MCBYV_4.5K / Imax_MCBYV_4.5K;
//---------------------- VKICKER       ----------------------------------------------
MBAW : VKICKER, L := l.MBAW, Kmax := Kmax_MBAW, Kmin := Kmin_MBAW, Calib := Kmax_MBAW / Imax_MBAW;
MBWMD : VKICKER, L := l.MBWMD, Kmax := Kmax_MBWMD, Kmin := Kmin_MBWMD, Calib := Kmax_MBWMD / Imax_MBWMD;
MBXWT : VKICKER, L := l.MBXWT, Kmax := Kmax_MBXWT, Kmin := Kmin_MBXWT, Calib := Kmax_MBXWT / Imax_MBXWT;

/************************************************************************************/
/*                        LHC SEQUENCE                                              */
/************************************************************************************/

LHCB1 : SEQUENCE, refer = CENTRE, L = LHCLENGTH;
IP1:OMK,                           at= pIP1+IP1OFS.B1*DS;
  MBAS2.1R1:MBAS2,                 at= 1.5+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 2209454,
  TAS.1R1:TAS,                     at= 19.95+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 102103,
  BPMSW.1R1.B1:BPMSW002,           at= 21.564+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 6080259, assembly_id= 6080224,
  BPMSW.1R1.B1_DOROS:BPMSW002,     at= 21.564+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 10429420, assembly_id= 6080224,
  BPMWK.1R1:BPMWK,                 at= 21.62+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 6080224,
  BPMWF.A1R1.B1:BPMWF,             at= 21.724+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 6080267, assembly_id= 6080224,
  MQXA.1R1:MQXA,                   at= 26.15+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 282126, assembly_id= 102104,
  MCBXH.1R1:MCBXH,                 at= 29.842+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 282213, assembly_id= 102104,
  MCBXV.1R1:MCBXV,                 at= 29.842+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 282212, assembly_id= 102104,
  BPMS.2R1.B1:BPMS,                at= 31.529+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 241889, assembly_id= 102105,
  MQXB.A2R1:MQXB,                  at= 34.8+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 241890, assembly_id= 102105,
  MCBXH.2R1:MCBXH,                 at= 38.019+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 249450, assembly_id= 102105,
  MCBXV.2R1:MCBXV,                 at= 38.019+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 249451, assembly_id= 102105,
  MQXB.B2R1:MQXB,                  at= 41.3+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 241892, assembly_id= 102105,
  TASB.3R1:TASB,                   at= 45.342+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 241893, assembly_id= 102106,
  MQSX.3R1:MQSX,                   at= 46.608+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 282127, assembly_id= 102106,
  MQXA.3R1:MQXA,                   at= 50.15+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 241895, assembly_id= 102106,
  MCBXH.3R1:MCBXH,                 at= 53.814+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 249456, assembly_id= 102106,
  MCBXV.3R1:MCBXV,                 at= 53.814+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 249457, assembly_id= 102106,
  MCSX.3R1:MCSX,                   at= 53.814+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 249458, assembly_id= 102106,
  MCTX.3R1:MCTX,                   at= 53.814+(0-IP1OFS.B1)*DS, mech_sep= 0, slot_id= 249459, assembly_id= 102106,
                                                                                          808,1          2%
```

Beam definition & sequence activation

Generic pattern to define the beam:

label: BEAM, PARTICLE=x, ENERGY[a]=y,. . . ;

e.g., BEAM, PARTICLE=proton, ENERGY=7000;//in GeV

[a] It is the TOTAL energy!

After a sequence has been read, it can be activated:

USE, SEQUENCE=sequence_label;

e.g., USE, SEQUENCE=lhc1;

The USE command expands the specified sequence, inserts the drift spaces and makes it active.

Definition of operations

Once the sequence is activated we can perform operations on it.

- Calculation of Twiss parameters around the machine (very important) in order to know, for stable sequences, their main optical parameters.
  TWISS, SEQUENCE=sequence_label;//periodic solution
  TWISS, SEQUENCE=sequence_label, betx=1;//IC solution

- Production of graphical output of the main optical function (e.g., β-functions):
  PLOT, HAXIS=s, VAXIS=betx,bety;

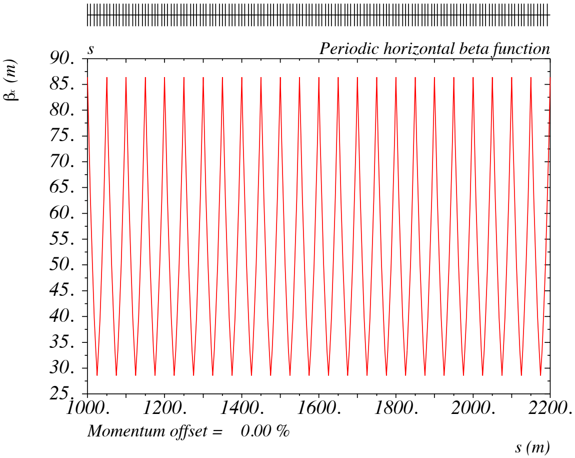
Example

TWISS, SEQUENCE=juaseq, FILE=twiss.out;
PLOT, HAXIS=s, VAXIS=betx, bety, COLOUR=100;

EXAMPLE of a the TWISS file

```
* 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
  ....
  ....
```

EXAMPLE of the graphical output (ps format)



Matching global parameters

It is possible to modify the optical parameters of the machine using the MATCHING module of MAD-X.

- Adjust magnetic strengths to get desired properties (e.g., tune Q, chromaticity dQ),
- Define the properties to match and the **parameters** to vary.

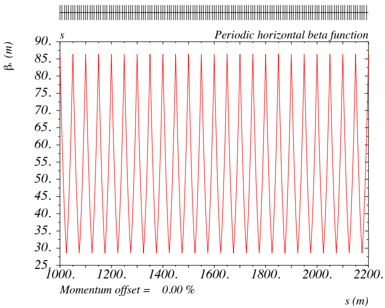
Example:

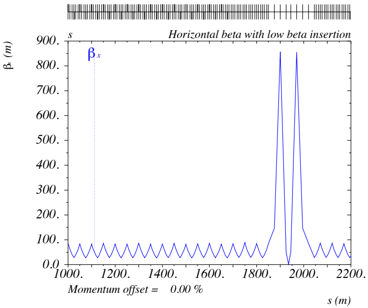
```
MATCH, SEQUENCE=sequence_name;
    GLOBAL, Q1=26.58;//H-tune
    GLOBAL, Q2=26.62;//V-tune
    VARY, NAME= kqf, STEP=0.00001;
    VARY, NAME = kqd, STEP=0.00001;
    LMDIF, CALLS=50, TOLERANCE=1e-6;//method adopted
ENDMATCH;
```

Other types of matching I

Local matching and performance matching:

- Local optical functions (insertions, local optics change),
- any user defined variable.





Other types of matching II

Local matching and performance matching:

- Local optical functions (insertions, local optics change),
- any user defined variable.

Example:

```
MATCH, SEQUENCE=sequence_name;
    CONSTRAINT, range=#e, BETX=50;
    CONSTRAINT, range=#e, ALFX=-2;
     VARY, NAME= kqf, STEP=0.00001;
    VARY, NAME = kqd, STEP=0.00001;
    JACOBIAN, CALLS=50, TOLERANCE=1e-6;
ENDMATCH;
```

“Hello World!” input file

LectureExample — sterbini@lxplus101:/eos/user/s/sterbini/_First/2018/LHC MD Optics/injection — vi fodo.mad — 92×38

```
/****Definition of elements****/
qfType:QUADRUPOLE, L=1.5, K1:=kf;
qdType:QUADRUPOLE, L=1.5, K1:=kd;

/****Definition of the sequence****/
fodo:SEQUENCE, REFER=exit, L=10;
qf: qfType, at=5;
qd: qdType, at=10;
ENDSEQUENCE;

/****Definition of the strength****/
kf=+0.25;
kd:=-kf;

/****Definition of the beam****/
beam, particle=proton, energy=7001;

/****Activation of the sequence****/
use, sequence=fodo;

/****Operations****/
twiss, file=beforeMatching.twiss;
plot, HAXIS=s, VAXIS=betx, bety, title='Before matching';

/****Matching****/
MATCH, sequence=fodo;
  GLOBAL, Q1=.25;
  GLOBAL, Q2=.25;
  VARY, NAME=kf, STEP=0.00001;
  VARY, NAME=kd, STEP=0.00001;
  LMDIF, CALLS=50, TOLERANCE=1e-8;
ENDMATCH;

/****Operations****/
twiss, file=afterMatching.twiss;
plot, HAXIS=s, VAXIS=betx, bety, title='after matching', interpolate=true;
QUIT;
"fodo.mad" 37L, 842C
```

"Hello World!" output (1)

LectureExample — sterbini@lxplus101:/eos/user/s/sterbini/_First/2018/LHC MD Optics/injection — -bash — 92×38

```
/****Definition of elements****/

qfType:QUADRUPOLE, L=1.5, K1:=kf;

qdType:QUADRUPOLE, L=1.5, K1:=kd;


/****Definition of the sequence****/

fodo:SEQUENCE, REFER=exit, L=10;

qf: qfType, at=5;

qd: qdType, at=10;

ENDSEQUENCE;


/****Definition of the strength****/

kf=+0.25;

kd:=-kf;


/****Definition of the beam****/

beam, particle=proton, energy=7001;


/****Activation of the sequence****/

use, sequence=fodo;
```

"Hello World!" output (2)

LectureExample — sterbini@lxplus101:/eos/user/s/sterbini/_First/2018/LHC MD Optics/injection — -bash — 92×38

```
/****Operations****/

twiss, file=beforeMatching.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

            nflips
                 0
plot, HAXIS=s, VAXIS=betx, bety, title='Before matching';

Plot - default table plotted: twiss

 GXPLOT-X11  1.50 initialized

 plot number =             1
```

"Hello World!" output (3)

LectureExample — sterbini@lxplus101:/eos/user/s/sterbini/_First/2018/LHC MD Optics/injection — -bash — 92×38

```
START LMDIF:

Initial Penalty Function =   0.86906699E+00


call:       4   Penalty function =   0.12041476E-01
call:       7   Penalty function =   0.18270348E-05
call:      10   Penalty function =   0.40829956E-13
 ++++++++++ LMDIF ended: converged successfully
call:      10   Penalty function =   0.40829956E-13
ENDMATCH;


MATCH SUMMARY

Node_Name                  Constraint   Type  Target Value       Final Value        Penalty
--------------------------------------------------------------------------------------------
------
Global constraint:         q1           4     2.50000000E-01     2.50000014E-01     1.836786
89E-14
Global constraint:         q2           4     2.50000000E-01     2.50000015E-01     2.246208
74E-14


Final Penalty Function =   4.08299562e-14




Variable                 Final Value  Initial Value Lower Limit  Upper Limit
--------------------------------------------------------------------------------
kf                        2.11022e-01  2.50000e-01 -1.00000e+20  1.00000e+20
kd                       -2.11022e-01 -2.50000e-01 -1.00000e+20  1.00000e+20

END MATCH SUMMARY
```

"Hello World!" output (4)

LectureExample — sterbini@lxplus101:/eos/user/s/sterbini/_First/2018/LHC MD Optics/injection — -bash — 92×38

```
/****Operations****/

twiss, file=afterMatching.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
      0.2500000136      -0.3176945739        14.60761389                  0

             dxrms             xcomax             xcorms                 q2
                 0                  0                  0        0.250000015

               dq2            betymax              dymax              dyrms
     -0.3176945752        14.60761386                  0                  0

            ycomax             ycorms             deltap            synch_1
                 0                  0                  0                  0

           synch_2            synch_3            synch_4            synch_5
                 0                  0                  0                  0

            nflips
                 0
plot, HAXIS=s, VAXIS=betx, bety, title='after matching', interpolate=true;

Plot - default table plotted: twiss
 plot number =            2
QUIT;
```

MAD-X and *python*

There is a very convenient python interface to MAD-X via the cpymadx package

- setup the environment instructions: http://cern.ch/go/7bZZ
- a simple example: http://cern.ch/go/DKZ6.

MAD-X test

[1]:
```
from cpymad.madx import Madx
from matplotlib import pyplot as plt
```

Launching MAD-X

[2]:
```
myMad = Madx()
```

```
++++++++++++++++++++++++++++++++++++++++++++
+     MAD-X 5.05.00  (64 bit, Linux)       +
+ Support: mad@cern.ch, http://cern.ch/mad +
+ Release   date: 2019.05.10               +
+ Execution date: 2019.06.03 12:48:59      +
++++++++++++++++++++++++++++++++++++++++++++
```

Definition of parameters

[3]:
```
myString='''
! *********************************************************************
! Definition of parameters
! *********************************************************************

l_cell=100;
quadrupoleLenght=5;
f=200;
myK:=1/f/quadrupoleLenght;// m^-2
'''
myMad.input(myString);
```