Mathematical and Numerical Methods for Non-linear Beam Dynamics (an introduction)

Part 1

Werner Herr
CERN

Talk: http://cern.ch/Werner.Herr/CAS2015_LECTURES/Otwock_methods.pdf

Recommended Bibliography:

[AW] A. Wolski, *Beam Dynamics in High Energy Particle Accelerators*, Imperial College Press, 2014.

[AC1] A. Chao, *Lecture Notes on Topics in Accelerator Physics*, SLAC, 2001.

[WH] W. Herr, *Mathematical and Numerical Methods for Non-linear Dynamics* in Proceedings of the CERN Accelerator School: **Advanced Accelerator Physics**, Trondheim, Norway, 18-29 August 2013, edited by W. Herr, CERN-2014-009 (CERN, Geneva, 2014), pp. 157-198.

[AC2] A. Chao and M. Tigner, *Handbook of Accelerator Physics and Engineering*, World Scientific Publishing, 1998.

[EF] E. Forest, *Beam Dynamics - A New Attitude and Framework*, Harwood Academic Publishers, 1998.

[AD] A. Dragt, *Lie Methods for Non-linear Dynamics with Applications to Accelerator Physics*

[MB] M. Berz, *Modern Map Methods in Particle Beam Physics*, Academic Press, 1999.

Additional material also in handout of the lecture and backup slides

Linear Dynamics in particle accelerators

Linear dynamics, optics and lattice concepts are traditionally developped for periodic systems, i.e. synchrotrons and storage rings, but:

- Most machines are not synchrotrons, let alone storage rings
- They are not periodic systems (sources, beam lines, linacs, cyclotrons, FELs, ERLs, ...)
- Most accelerators accelerate particles !!
- Particles in general are not relativistic

→ Tools and concepts should be applicable to all these cases

... and they exist, as we shall see

Menu and Outline of these lectures

1. Motivation and discussion of standard concepts.
   Important: use of correct variables
2. Introduce a modern and contemporary*) approach that automatically leads from simple applications to general and powerful concepts (necessarily brief and incomplete):
3. No attempt is made at mathematical rigor, (much more rigorous, derivations and many examples in [AW][EF])
   - Develop the concepts first for linear beam dynamics
   - Generalize to non-linear dynamics
4. We get a general and consistent formalism (linear and non-linear)

*) the oldest reference is from 1998!

Motivation - why should you listen to me ?

Most introductions to transverse dynamics and beam optics deal with ”linear” (ideal) machines

But there are a few problems:

- Linear (ideal) machines do not exist
- Linear (ideal) machines do not work !
- Non-linearities unavoidable (and needed !)
- Non-linear (real) machines work

What we shall see:

- Concept of linear and non-linear maps
- Methods for analysis of the dynamical behaviour
- Symplectic integrators
- Lie operators and transformations
- Use of Truncated Power Series Algebra

→ (some applications in lectures by Y. Papaphilippou)

First: which variables should be used ?

(I) Start with standard Cartesian coordinates:

$$(X, Y, Z)$$

I use X, Y as transverse coordinates (some use X, Z)
Usually impossible to solve equation of motion (and useless)

Find new coordinate system where transverse variables remain small as the beam moves along Z (see Transverse Dynamics)

$$(X, Y) \implies (x, y)$$

These transverse variables are the transverse distances from reference path.

(II) Independent variable:

The independent variable is the time t and the time increases continuously as we go along

We are much more interested in the distance between particles as we go along (in particular: longitudinal deviation from the reference particle):

We therefore replace time t by:

$s = \beta_0 \cdot c \cdot t \approx c \cdot t$		distance along (curved)*) reference path
$\Delta s = c \cdot \Delta t$		distance from reference particle

So we have for the reference particle:

$$(x, y, \Delta s) = (0, 0, 0)$$

*) for details: other lectures or handout

(III) We also need momenta in the set of variables:

Strictly speaking: one should always use canonical variables (we have to use them later), and in this ordering !

$$x, p_x, y, p_y$$

With an uneasy feeling, to be consistent with other lectures and various textbooks (where it is not crucial*):

$$x, \quad x' = \frac{p_x}{p} = \frac{\partial x}{\partial s}$$

$$y, \quad y' = \frac{p_y}{p} = \frac{\partial y}{\partial s}$$

*) Do not believe there is no difference !!

(IV) Momentum deviation and spread:

Momentum of a particle P may be different from the momentum of the reference particle P_0, we define as the momentum deviation δ:

$$\delta = \frac{P - P_0}{P_0} = \frac{\Delta P}{P_0}$$

An ensemble of particles (let's call it a beam) may have a r.m.s. momentum spread σ_δ

$$\sigma_\delta^2 = < \delta^2 >$$

So we can finally use: $\underline{(x, p_x, y, p_y, \Delta s, \delta)}$ or $\underline{(x, x', y, y', \Delta s, \delta)}$

(For bar discussions: does it make a difference if you have a beam with $\sigma_\delta \neq 0$??)

Terminology: PHASE SPACE

Unfortunately different definitions in textbooks and articles, here I am rigorous

Correct version, generalized coordinates and conjugate generalized momenta:

$$x, p_x, y, p_y, \ldots$$

Wrong version, generalized coordinates and generalized velocities:

$$x, v_x, y, v_y, \ldots$$

Totally wrong version, generalized coordinates and angular deviation:

$$x, x', y, y', \ldots$$

Again: do not believe there is no difference, for a discussion see [AW]

Treatment of linear dynamics in rings

For synchrotrons: often introduction using Hill's equation

For simplicity → in one dimension (as done in most lectures and textbooks !):

$$\frac{d^2x(s)}{ds^2} + K(s)x(s) = 0$$

$K(s)$ periodic function (with period one)

$$K(s) = K(s + C)$$

Use the Courant-Snyder ansatz (only in 1D, x):

$$x(s) = \sqrt{\beta_x(s) \cdot 2J_x} \cdot \cos(\mu_x(s) + \phi)$$

$$x'(s) = \sqrt{\frac{2J_x}{\beta_x(s)}} \cdot (\sin(\mu_x(s) + \phi) + \alpha_x \cdot \cos(\mu(s) + \phi))$$

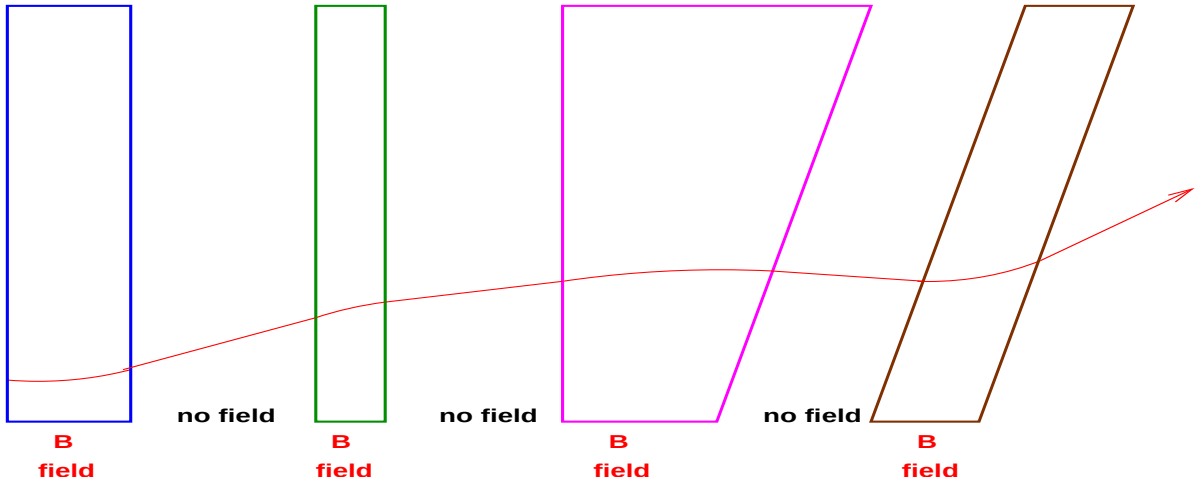
Is the solution to a system that is: linear, confined and periodic !

What about: Beam lines, Linacs, Cyclotrons, ... ?

There are no α-, β-functions !

Question: is a particle with the above solution always stable ?

Trajectories through (arbitrary) magnetic structures



Not a solution of the above (not periodic !) .....

Need something else

But it can be worse

Adding non-linear fields

with distortions, we have to re-write (similar for the other plane):

$$\frac{d^2x(s)}{ds^2} + K(s)x(s) = -\frac{B_y(x, y, s)}{p}$$

or in general as (any order) multipoles:

$$\frac{d^2x(s)}{ds^2} + K(s)x(s) = \sum_{i,j\geq 0} p_{ij}(s)x^i y^j$$

→ For exponents $(i + j) \geq 2$: enter non-linear dynamics

e.g. $(i + j) = 2$: sextupole

→ Very non-linear differential equation to solve

Is it relevant and can we deal with that ?

$$\frac{d^2x(s)}{ds^2} + K(s)x(s) = \sum_{i,j\geq 0} p_{ij}(s)x^i y^j$$

- Bad news:

  For LHC: we go to 20th order ($(i+j) \leq 20$) !

  We have no global solution

  Mostly relevant for hadrons (no damping), e.g. protons, ions

- Good news:

  A global analytical solution is not needed !

Why not ??

We do not want to know (we do not care !):

→ The particle's position and momentum after 12 hours*)

(Remember Thermodynamics !)

We do want to know:

→ Is the beam stable for a long time ?

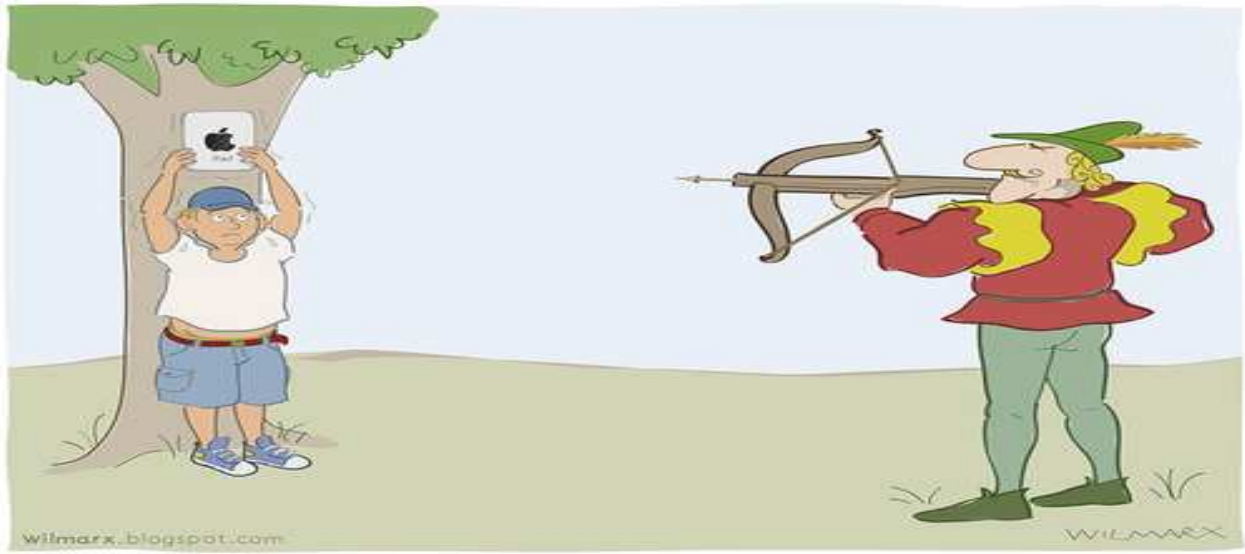
→ Is the motion confined ?

→ What are global properties (e.g. Q, Q', beam size, ...) ?

→ Does the beam hit the target ?

*) ... unlike GPS satellites

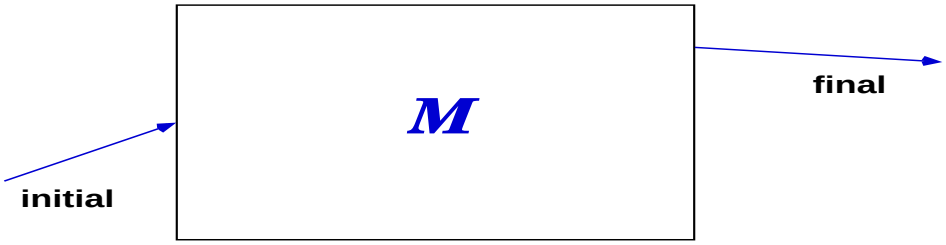
An every day example ...



- Most important to know trajectory at beginning and end of flight, and how to get there
- Not important to know exact trajectory as function of time or distance (unless somebody shoots at you as well ..)
- Can we get a better framework to get that (easily) ?

How can we describe the flight ?

We perform an operation → mapping $\mathcal{M}$:



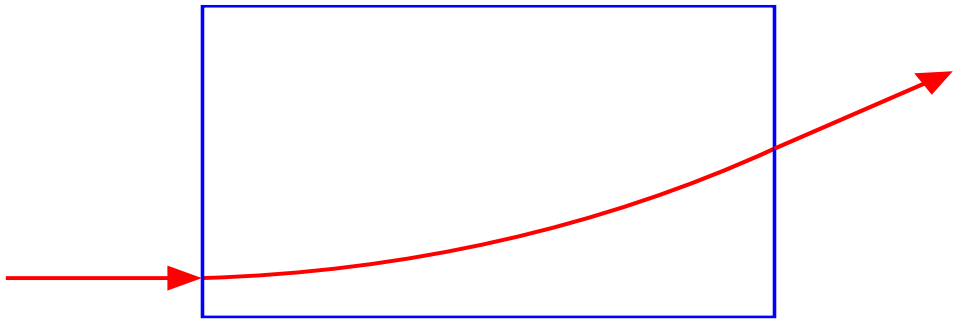
The map $\mathcal{M}$ sends the initial conditions to the final conditions, it is a "generalized" function: Transfer map

What happens inside $\mathcal{M}$ is important only when it is constructed !!

There are many different ways to do it ! (for the same result) see later !!

Applied to accelerators ...

We want to describe what happens to the particle in an element (e.g. an electric or magnetic field)



Assume we have: magnetic field $\vec{B}(\vec{z})$

Newtons law and Lorentz force:

$$\frac{d\vec{z}}{dt} = \vec{v}$$

$$\frac{d}{dt} m_0 \gamma \vec{v} = Q\vec{v} \times \vec{B}$$

We do not need to know: type of element

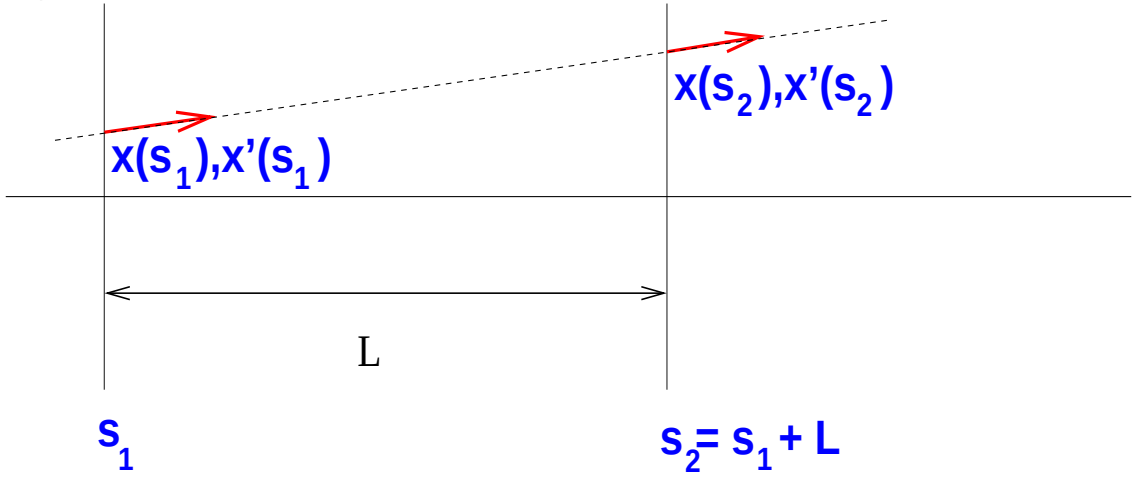
What can $\mathcal{M}$ be ?

- The ”maps” can be:
  - In the simplest (linear) case the ”map” can be written as a ”matrix”
  - A non-linear transformation (Taylor series, Lie Transform ...)
  - High order integration algorithm
  - A computer program, subroutine etc.
- How to get a map ? See later !
- Let us look at the linear theory first !

  Then generalize to non-linear theory

Simple examples - all derivations later

First a drift space (one dimension only) of length L, starting at position s_1 and ending at s_2



The simplest description (1D) is (should be in 3D of course):

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{s_2} = \begin{pmatrix} 1 & L \\ 0 & 1 \end{pmatrix} \circ \begin{pmatrix} x \\ x' \end{pmatrix}_{s_1}$$

This is only an approximation, see later ... !

Another example

Focusing quadrupole of length L and strength k_1 ($k_1 > 0$):

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{s_2} = \begin{pmatrix} \cos(L \cdot \sqrt{k_1}) & \frac{1}{\sqrt{k_1}} \cdot \sin(L \cdot \sqrt{k_1}) \\ -\sqrt{k_1} \cdot \sin(L \cdot \sqrt{k_1}) & \cos(L \cdot \sqrt{k_1}) \end{pmatrix} \circ \begin{pmatrix} x \\ x' \end{pmatrix}_{s_1}$$

for $k_1 < 0$ we have: $\sqrt{k_1} = i \cdot \sqrt{|k_1|}$ it becomes imaginary and re-write

$\sin(x) = -i \cdot \sinh(ix)$ and $\cos(x) = \cosh(ix)$

$$\begin{pmatrix} y \\ y' \end{pmatrix}_{s_2} = \begin{pmatrix} \cosh(L \cdot \sqrt{k_1}) & \frac{1}{\sqrt{k_1}} \cdot \sinh(L \cdot \sqrt{k_1}) \\ \sqrt{k_1} \cdot \sinh(L \cdot \sqrt{k_1}) & \cosh(L \cdot \sqrt{k_1}) \end{pmatrix} \circ \begin{pmatrix} y \\ y' \end{pmatrix}_{s_1}$$

How did we get the sin, cos, sinh, cosh ? See tomorrow ..

Side Note:

In case we have a momentum deviation δ we have to replace:

$$k_1 \implies \frac{k_1}{D}$$

with

$$D = \sqrt{1 + \frac{2\delta}{\beta_r} + \delta^2}$$

we then have a momentum dependent focusing (chromaticity)

Watch out for $\beta_r \ll 1$, usually ignored, but such machines exist !!

Another Side Note:

One can define:

$$\Gamma^2 = k_1 = \frac{1}{B\rho}\frac{dB_y}{dx}$$

Using Γ*) instead of $\sqrt{k}$ → the matrix for focusing and defocusing quadrupoles have the same form !

Frequently used for ”complicated” calculations and to get ”readable” formulae (e.g. [WH, AC1]), so watch out !

*) or whatever you call it ...

Putting the ”pieces” together

We have to deal with many elements in our machines

To make a ring or beam line:

- Combine the maps of all elements together
- Concatenated maps are a map again
- Represents a bigger part of the machine

  (or the whole machine ...)

  We have a lumped map ..

Starting from a position s_0 and applying all maps in sequence of length L to get the overall map for the position $s_0 + L$ (shown for 1D only):

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{s_0 + L} = \mathcal{M}_1 \circ \mathcal{M}_2 \circ \ldots \circ \mathcal{M}_N \circ \begin{pmatrix} x \\ x' \end{pmatrix}_{s_0}$$

$$\Longrightarrow \begin{pmatrix} x \\ x' \end{pmatrix}_{s_0 + L} = \mathcal{M}(s_0, L) \circ \begin{pmatrix} x \\ x' \end{pmatrix}_{s_0}$$

If the elements form a ring with circumference C:

Starting at position s_0 we get the One-Turn-Map (OTM):

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{s_0 + C} = \mathcal{M}_{ring}(s_0) \circ \begin{pmatrix} x \\ x' \end{pmatrix}_{s_0}$$

Please note: $\mathcal{M}_{ring}(s_0)$ depends on s_0 !!

In the simplest case the One-Turn-Map is a One-Turn-Matrix:
Combination becomes a simple matrix multiplication.

How do we extract the wanted information (first for a matrix, general case tomorrow) ?

Analysis of our One-Turn-Map with Normal Forms

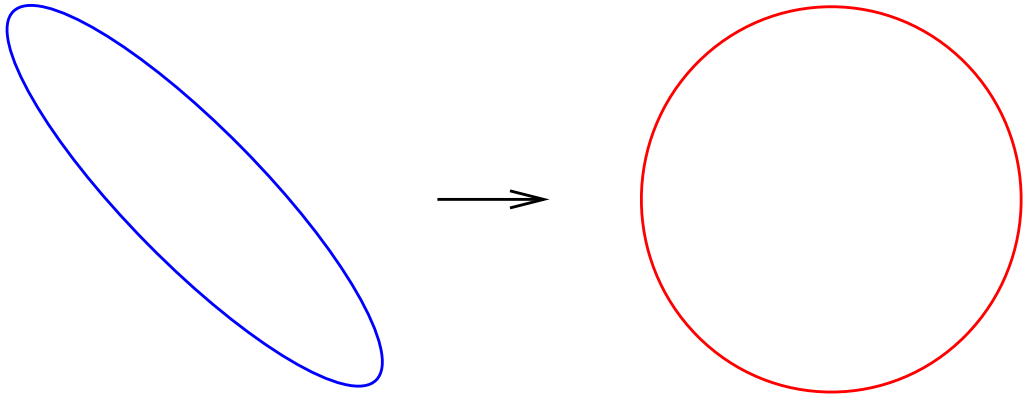
- Matrices can be transformed into (Jordan) Normal Forms
- Original matrix and normal form are equivalent, but ...
- Easier to analyse the matrix to obtain:
  - Stability
  - Get parameters (Q, Q', Twiss function, ..)
  - Study invariants, etc.
  - For resonance analysis (e.g. driving terms)
  - etc. ...
- Idea is to make a transformation to get a simpler form of the map

Assume $\mathcal{M}$ is our One-Turn-Matrix, we try to find a (invertible) transformation $\mathcal{A}$ such that:

$$\mathcal{A}\mathcal{M}\mathcal{A}^{-1} = \mathcal{R}$$

- The matrix $\mathcal{R}$ is:
  - A ”Jordan Normal Form”, (or at least a very simplified form of the matrix)
  - Example: $\mathcal{R}$ becomes a pure rotation
- The matrix $\mathcal{R}$ describes the same dynamics as $\mathcal{M}$, but:
  - All coordinates are transformed
  - The transformation $\mathcal{A}$ ”analyses” the motion

Transformation to Normal Form (pictorial)



$$M = \mathcal{A} \circ \mathcal{R}(\Delta\mu) \circ \mathcal{A}^{-1} \quad \text{or :} \quad \mathcal{R}(\Delta\mu) = \mathcal{A}^{-1} \circ M \circ \mathcal{A}$$

Motion on an ellipse becomes motion on a circle (i.e. a pure rotation)

$\mathcal{R}(\Delta\mu)$ is the ”simple” part of the map

How to get that ? Remember lectures on Linear Algebra (Eigenvectors, Eigenvalues ...)

We find the two components of the original map:

$$\mathcal{A} = \begin{pmatrix} \sqrt{\beta(s)} & 0 \\ -\frac{\alpha}{\sqrt{\beta}} & \frac{1}{\sqrt{\beta(s)}} \end{pmatrix} \text{ and } \mathcal{R} = \begin{pmatrix} cos(\mu_x) & sin(\mu_x) \\ -sin(\mu_x) & cos(\mu_x) \end{pmatrix}$$

- From the Normal Form transformation we get plenty of information:
  - We get $\beta, \alpha, \ldots$ etc., μ_x is the tune $Q_x \cdot 2\pi$!
  - The machine is stable when μ_x (and μ_y etc.) are real (i.e. not complex)
  - We get the closed orbit

No need to make any assumptions, ansatz, approximations, ...

Here demonstrated for one dimension, but it works for many dimensions

A key: it does not matter how we got the map ! (see later)

Side note - normalized variables:

Also the variables are transformed:

$$\begin{pmatrix} x_n \\ x'_n \end{pmatrix} = \mathcal{A}^{-1} \circ \begin{pmatrix} x \\ x' \end{pmatrix}$$

is just a variable transformation to new, normalized variables:

$$\gamma x^2 + 2\alpha_x x x' + \beta_x x'^2 \quad \rightarrow \quad x_n^2 + x_n'^2$$

Another example: linear coupling (2D)

Assume a one-turn-matrix in 2D (4×4 matrix):

$$T = \begin{pmatrix} M & n \\ m & N \end{pmatrix}$$

M,m,N,n are 2-by-2 matrices.

In case of coupling: $m \neq 0, n \neq 0$ we can try to re-write as:

$$T = \begin{pmatrix} M & n \\ m & N \end{pmatrix} = VRV^{-1}$$

with (same procedure as before):

$$R = \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix} \quad \text{and} \quad V = \begin{pmatrix} \gamma I & C \\ -C^t & \gamma I \end{pmatrix}$$

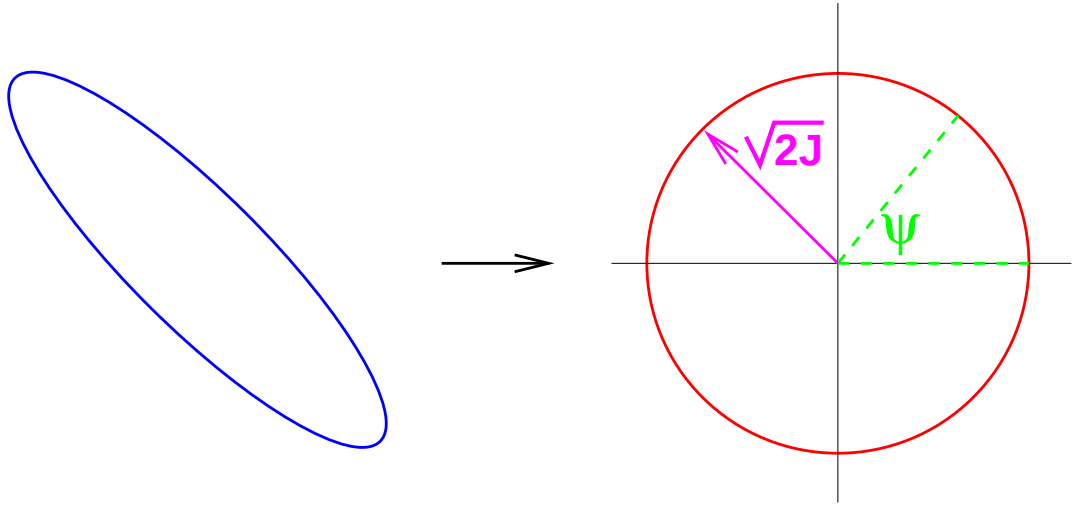
What have we obtained ?

- The matrix R is our simple rotation, now in 2D:
  - A and B are the one-turn-matrices for the ”normal modes”, each in 1D
  - The matrix C contains the ”coupling coefficients”
  - The matrix V transforms from the coordinates (x, x', y, y') into the ”normal mode” coordinates (w, w', v, v') via the expression:

    $$(x, x', y, y') = V(w, w', v, v')$$

- One simple operation !

Description using Action - Angle variables

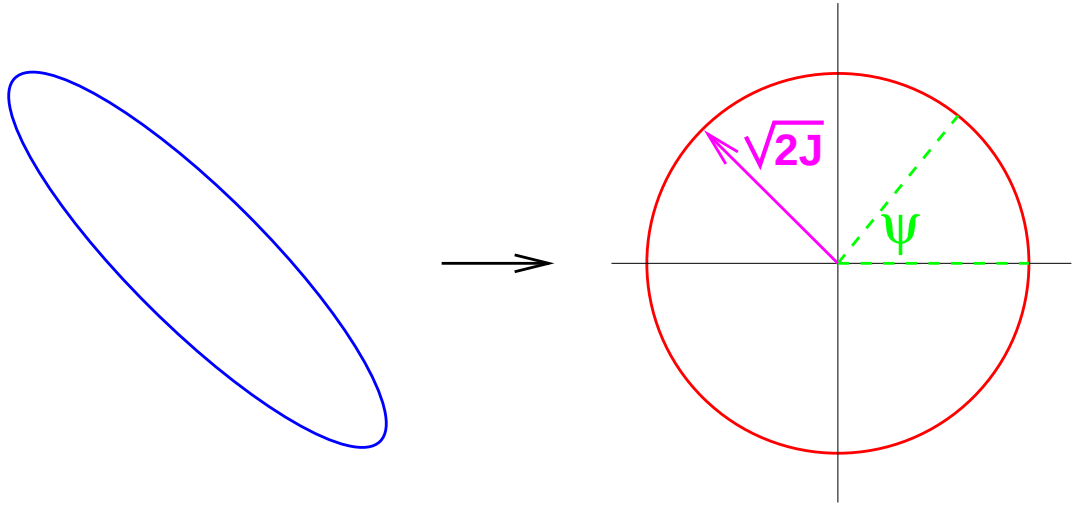


Once the particles ”travel” on a circle:

- Radius $\sqrt{2J}$ is constant (invariant of motion): action J
- Phase advances by constant amount: angle Ψ

  i.e. the angle Ψ is the only independent variable !

Description using Action - Angle variables



Related to phase space variables x and p_x:

$$x = \sqrt{2J_x\beta_x}\, cos(\Psi_x)$$

$$p_x = -\sqrt{\frac{2J_x}{\beta_x}}\, (sin(\Psi_x) + \alpha_x cos(\Psi_x))$$

$$J_x = \frac{1}{2}(\gamma_x x^2 + 2\alpha_x x p_x + \beta_x p_x^2) = \frac{1}{2}(x_n^2 + p_{xn}^2)$$

What are they good for ?

$$\int p \, \mathrm{d}q \;=\; const. \;=\; 2\pi J$$

for canonical variables*) q, p then J is a constant as the system evolves (Poincaré invariant).

We define an important quantity for a system with many particles:
beam emittance

$$\epsilon_q \;:=\; < J_q > \;=\; \sqrt{< q^2 >< p^2 > - < qp >^2} \qquad (q \;=\; x, y)$$

It is the average of the action over all particles (other definitions exist, but not generally valid)

*) This is a necessary condition, what it means see later ..

The general philosophy (linear systems):

- Describe your elements by a linear map
- Combine all maps into a ring or beam line to get the linear one turn matrix
- Normal form analysis of the linear one turn matrix will give all the information

No need for any assumptions !

No need for any approximations !

Works in more than 1D and with coupling !

Try now: A general philosophy (non-linear systems):

- Describe your elements by a non-linear map
- Combine all maps into a ring or beam line to get the non-linear one turn map
- Normal form analysis of the non-linear one turn map will give all the information

No need for any assumptions !

No need for any approximations !

Works in more than 1D and non-linearities !

A small complication ...

Non-linear maps are not matrices !

Various types of non-linear maps

Choice depends on the application, some examples:

- Taylor maps
- Symplectic integration techniques
- Lie transformations
- Truncated power series algebra (TPSA), can also generate maps from tracking

➢ Not all maps are allowed !

➢ Key concept: Symplecticity most relevant for rings !

Requires for a matrix $\mathcal{M} \rightarrow \mathcal{M}^T \cdot S \cdot \mathcal{M} = \mathcal{S}$ with:

$$S = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

It basically means: $\mathcal{M}$ is area preserving and

$$\lim_{n\to\infty} \mathcal{M}^n = \text{finite} \quad \Longrightarrow \quad \det \mathcal{M} = 1$$

To be rigorous: only for canonical variables

This form of S only for our standard ordering of variables (a general definition exists)

Introducing non-linear elements (e.g. sextupole)

Effect of a (thin) sextupole with strength k_2 is:

$$\vec{z}(s_2) = \begin{pmatrix} x \\ x' \\ y \\ y' \end{pmatrix}_{s_2} = \begin{pmatrix} x \\ x' \\ y \\ y' \end{pmatrix}_{s_1} + \begin{pmatrix} 0 \\ \frac{1}{2} k_2 L \cdot (x_{s_1}^2 - y_{s_1}^2) \\ 0 \\ k_2 L \cdot (x_{s_1} \cdot y_{s_1}) \end{pmatrix}$$

→ $\vec{z}(s_2) = \mathcal{M} \circ \vec{z}(s_1)$

→ $\mathcal{M}$ is not a matrix, i.e. cannot be expressed by matrix multiplication

We need something like [with: $\vec{z} = (z_1, z_2, z_3, z_4) = (x, x', y, y')$]:

$$
\begin{aligned}
z_j(s_2) \quad = \quad & R_{j1} \cdot x \quad + R_{j2} \cdot x' \quad + R_{j3} \cdot y + ... \\
& + T_{j11} \cdot x^2 + T_{j12} \cdot xx' + T_{j22} \cdot x'^2 + \\
& + T_{j13} \cdot xy + T_{j14} \cdot xy' + ... \\
& + U_{j111} \cdot x^3 + U_{j112} \cdot x^2 x' + ...
\end{aligned}
$$

and the equivalent for all other variables ...

Compact as higher order Taylor - Maps:

We have (for: j = 1...4):

$$z_j(s_2) = \sum_{k=1}^{4} R_{jk} z_k(s_1) + \sum_{k=1}^{4}\sum_{l=1}^{4} T_{jkl} z_k(s_1) z_l(s_1) + \sum_{k=1}^{4}\sum_{l=1}^{4}\sum_{m=1}^{4} U_{jklm} z_k(s_1) z_l(s_1) z_m(s_1) + \ldots\ldots$$

Has to be truncated at some order (e.g. order 2 for a sextupole)

Second order map for a sextupole with length L and strength k_2 (in 2D):

$$
\begin{aligned}
x_2 &= x_1 + Lx_1' && -k_2\left(\tfrac{L^2}{4}(x_1^2 - y_1^2) + \tfrac{L^3}{6}(x_1x_1' - y_1y_1') + \tfrac{L^4}{24}(x_1'^2 - y_1'^2)\right)\\
x_2' &= x_1' && -k_2\left(\tfrac{L}{2}(x_1^2 - y_1^2) + \tfrac{L^2}{2}(x_1x_1' - y_1y_1') + \tfrac{L^3}{6}(x_1'^2 - y_1'^2)\right)\\
y_2 &= y_1 + Ly_1' && +k_2\left(\tfrac{L^2}{4}x_1y_1 + \tfrac{L^3}{6}(x_1y_1' + y_1x_1') + \tfrac{L^4}{24}(x_1'y_1')\right)\\
y_2' &= y_1' && +k_2\left(\tfrac{L}{2}x_1y_1 + \tfrac{L^2}{2}(x_1y_1' + y_1x_1') + \tfrac{L^3}{6}(x_1'y_1')\right)
\end{aligned}
$$

Symplecticity for higher order MAPS

- Truncated Taylor expansions are not matrices !!
- It is the associated Jacobian matrix $\mathcal{J}$ which must fulfil the symplecticity condition:

$$\mathcal{J}_{ik} = \frac{\partial z_2^i}{\partial z_1^k} \quad \left(\text{e.g. } \mathcal{J}_{xy} = \frac{\partial z_2^x}{\partial z_1^y} \right)$$

$$\mathcal{J} \quad \text{must fulfil:} \quad \mathcal{J}^t \cdot \mathcal{S} \cdot \mathcal{J} = \mathcal{S}$$

- In general: $\mathcal{J}_{ik} \neq \text{const}$ → for truncated Taylor map can be difficult to fulfil for all z

Example: take the sextupole Taylor map (for simplicity in one dimension):

$$\begin{aligned} x_2 &= x_1 + Lx_1' \quad -k_2\left(\tfrac{L^2}{4}x_1^2 + \tfrac{L^3}{6}x_1x_1' + \tfrac{L^4}{24}x_1'^2 + O(3)\right) \\ x_2' &= x_1' \quad\quad\quad\quad -k_2\left(\tfrac{L}{2}x_1^2 + \tfrac{L^2}{2}x_1x_1' + \tfrac{L^3}{6}x_1'^2 + O(3)\right) \end{aligned}$$

we compute:

$$\mathcal{J}^T \cdot S \cdot \mathcal{J} = \begin{pmatrix} 0 & 1+\Delta S \\ -1-\Delta S & 0 \end{pmatrix} \neq S$$

is non-symplectic with error:

$$\Delta S = \frac{k_2^2}{72} L^4 (L^2 x'^2 + 6Lxx' + 6x^2)$$

Thick magnets

Taylor maps do not correctly describe non-linear magnets (not symplectic !)

(position and momentum change inside the magnet)

Is this a dead end ? Do we have to wave the white flag ?

Not yet: thin magnets →

In previous example: $\Delta S \propto L^4$

Small error for small L, no error for $L = 0$!

In a magnet with zero length (thin) the position cannot change

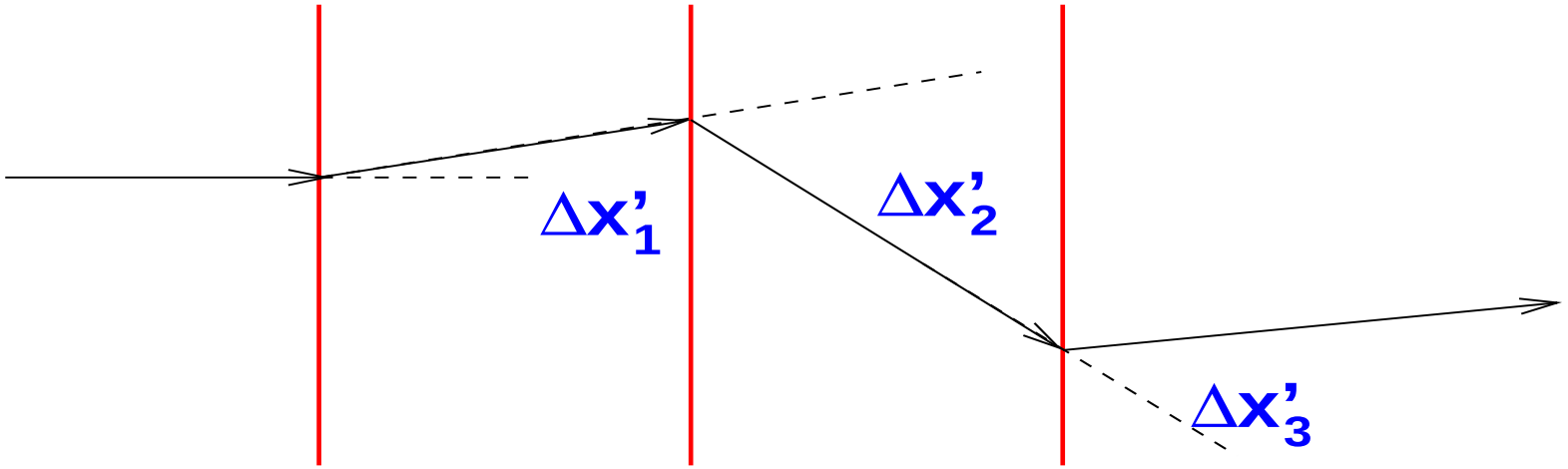
Thick ”magnet”: Length and Field used to compute effect

Thin ”magnet”: let the length go to zero, but keep field integral finite (constant):

$$L \cdot k_1, \quad L \cdot k_2, \quad L \cdot k_3, ...$$

Thin (zero length) elements are technically difficult, but much easier to use (how ?) ...

Moving through thin elements (shown for 1D)



No change of amplitude x

→ $\Delta x' = f(x)$ (polynomials of some order)

The ”momentum” x' receives an amplitude dependent deflection (kick)

$x' \rightarrow x' + \Delta x'$

Always symplectic (proof as tutorial exercise)

Using thin elements

Can we approximate a thick element by one or more thin element(s) ?

- Yes, when the length is small or does not matter
- Symplecticity o.k.
- What about accuracy, what have we lost ??
- Demonstrate with some simple examples

  (What follows is valid for all elements !!!)

Accuracy of thin lenses

- Start with exact map, compare with thin quadrupole

$$\mathcal{M}_{s\to s+L} = \begin{pmatrix} \cos(L\cdot\sqrt{K}) & \frac{1}{\sqrt{K}}\cdot\sin(L\cdot\sqrt{K}) \\ -\sqrt{K}\cdot\sin(L\cdot\sqrt{K}) & \cos(L\cdot\sqrt{K}) \end{pmatrix}$$

- Taylor expansion in ”small” length L:

$$L^0\cdot\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + L^1\cdot\begin{pmatrix} 0 & 1 \\ -K & 0 \end{pmatrix} + L^2\cdot\begin{pmatrix} -\frac{K}{2} & 0 \\ 0 & -\frac{K}{2} \end{pmatrix} + \ldots$$

- Keep up to first order term in L

$$\mathcal{M}_{s\to s+L} = L^0 \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + L^1 \cdot \begin{pmatrix} 0 & 1 \\ -K & 0 \end{pmatrix}$$

$$\mathcal{M}_{s\to s+L} = \begin{pmatrix} 1 & L \\ -K\cdot L & 1 \end{pmatrix} + \mathcal{O}(L^2)$$

- Precise to first order $\mathcal{O}(L^1)$
- $\det \mathcal{M} \neq 1$, non-symplectic (of course, this is obvious !)

$$\mathcal{M}_{s\to s+L} = \begin{pmatrix} 1 & L \\ -K\cdot L & 1 \end{pmatrix} + \mathcal{O}(L^2)$$

If we add a term $-KL^2$ the matrix becomes symplectic:
”symplectification” (it is wrong to $\mathcal{O}(L^2)$ anyway ...)

$$\rightarrow \quad \mathcal{M}_{s\to s+L} = \begin{pmatrix} 1 & L \\ -K\cdot L & 1-KL^2 \end{pmatrix}$$

Keep up to second order term in L

$$\mathcal{M}_{s\to s+L} = \begin{pmatrix} 1-\frac{1}{2}KL^2 & L \\ -K\cdot L & 1-\frac{1}{2}KL^2 \end{pmatrix} + \mathcal{O}(L^3)$$

- Precise to second order $\mathcal{O}(L^2)$
- More accurate than before, but again not symplectic

Made it symplectic by adding $-\frac{1}{4}KL^3$

$$\mathcal{M}_{s\to s+L} = \begin{pmatrix} 1-\frac{1}{2}KL^2 & L-\frac{1}{4}KL^3 \\ -K\cdot L & 1-\frac{1}{2}KL^2 \end{pmatrix} + O(L^3)$$

Precise to second order $O(L^2)$

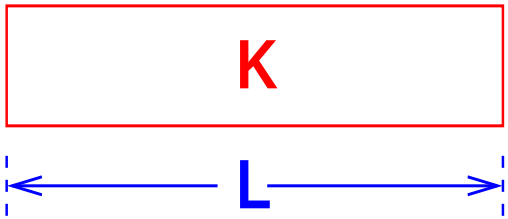
Looks like we made some arbitrary changes and called it ”symplectification”

Is there a physical picture behind the approximations ?

Yes, geometry of thin lens kicks ...

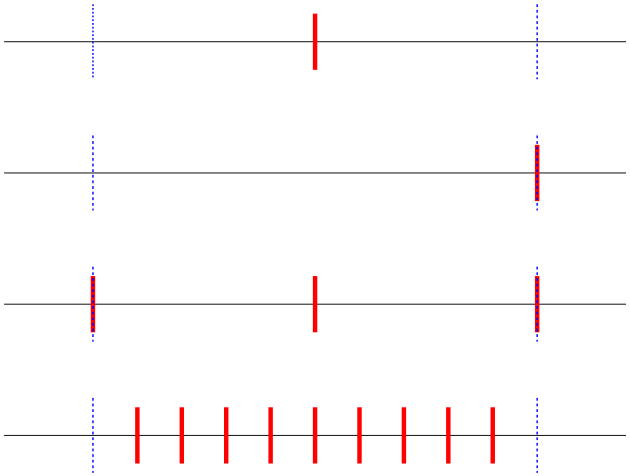
A thick element is split into thin elements with drifts between them

Thick → thin quadrupole



quadrupole of finite length

options:

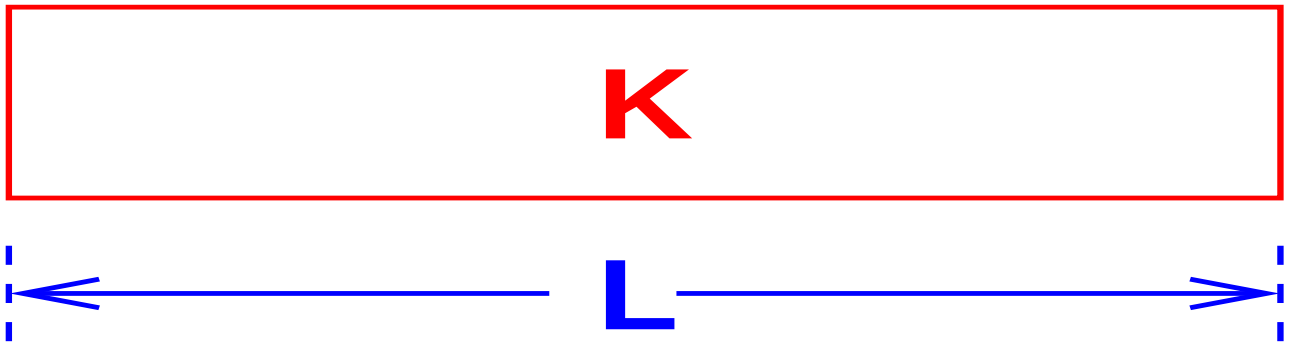


Represented by one or more ”thin” lenses (kicks)

How many and where ?

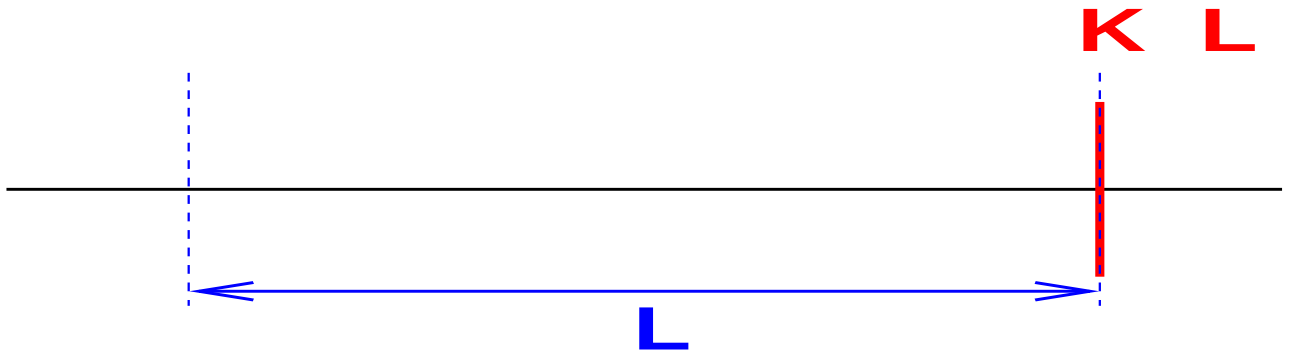
Which is a good strategy ? → accuracy and speed

Thick quadrupole ..



Go to thin quadrupoles → various options

Option 1



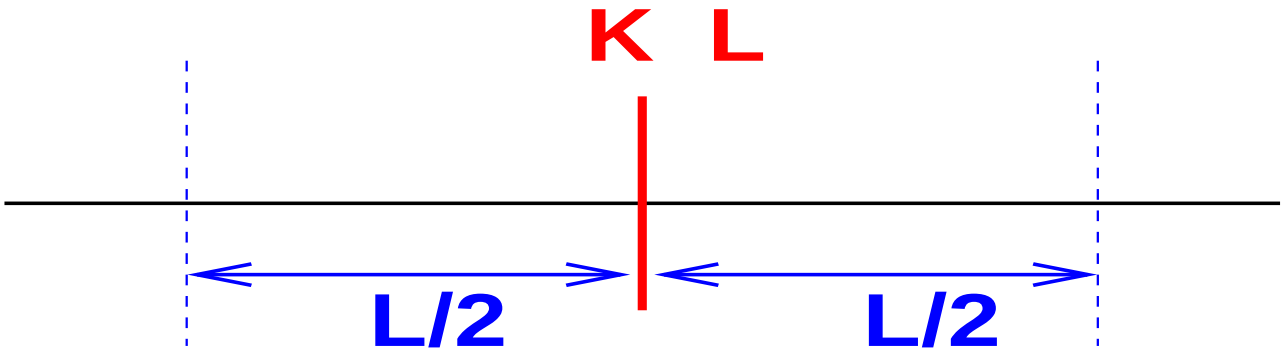
Separate L and K become $K \cdot L$

One thin quadrupole ”kick” and one drift combined (lumped)

$$\mathcal{M}_{lumped} = \begin{pmatrix} 1 & 0 \\ -K \cdot L & 1 \end{pmatrix} \begin{pmatrix} 1 & L \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & L \\ -K \cdot L & 1 - KL^2 \end{pmatrix}$$

Resembles ”symplectification” of order O(1)

Option 2



One thin quadrupole ”kick” between two drifts

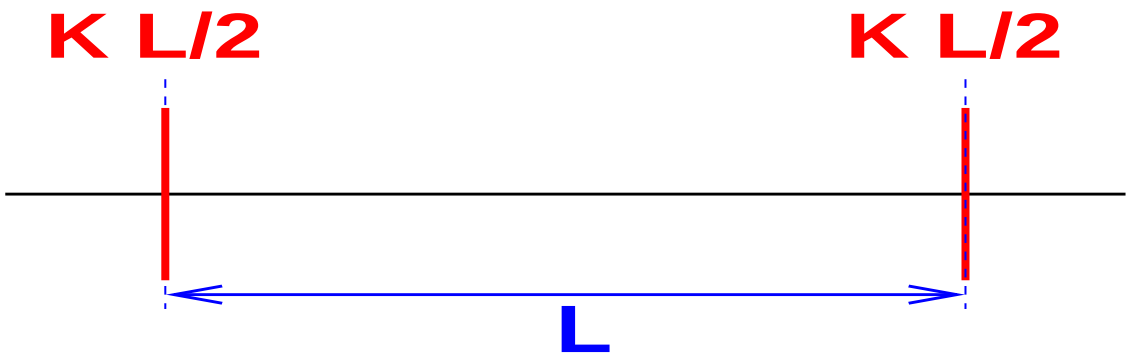
$$\mathcal{M}_{lumped} = \begin{pmatrix} 1 & \frac{1}{2}L \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -K \cdot L & 1 \end{pmatrix} \begin{pmatrix} 1 & \frac{1}{2}L \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 - \frac{1}{2}KL^2 & L - \frac{1}{4}KL^3 \\ -K \cdot L & 1 - \frac{1}{2}KL^2 \end{pmatrix}$$

Resembles more accurate ”symplectification” of order O(2)

Accuracy of thin lenses

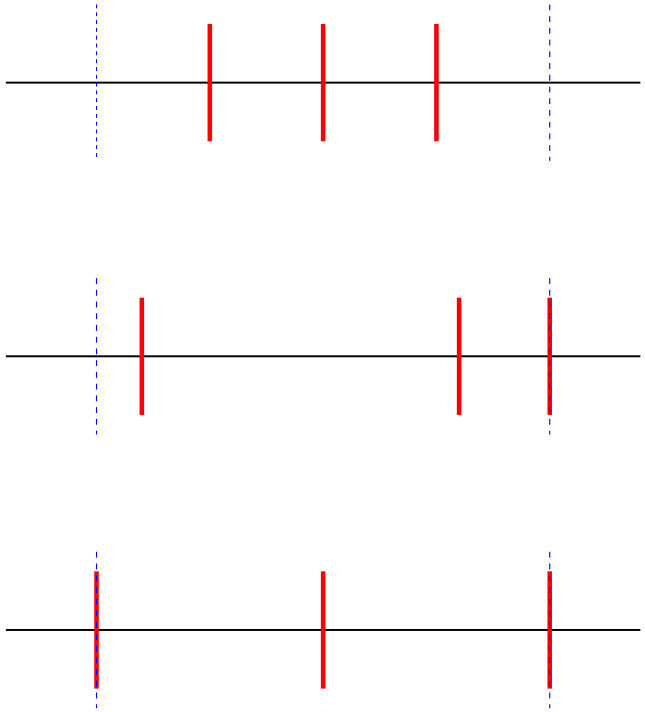
- One kick at the end (or beginning):
  - → Error (inaccuracy) of first order $\mathcal{O}(L^2)$
- One kick in the centre:
  - → Error (inaccuracy) of second order $\mathcal{O}(L^3)$
- It is very relevant how to apply thin lenses
- Aim should be to be precise and fast (and simple to implement)

What about this option ?



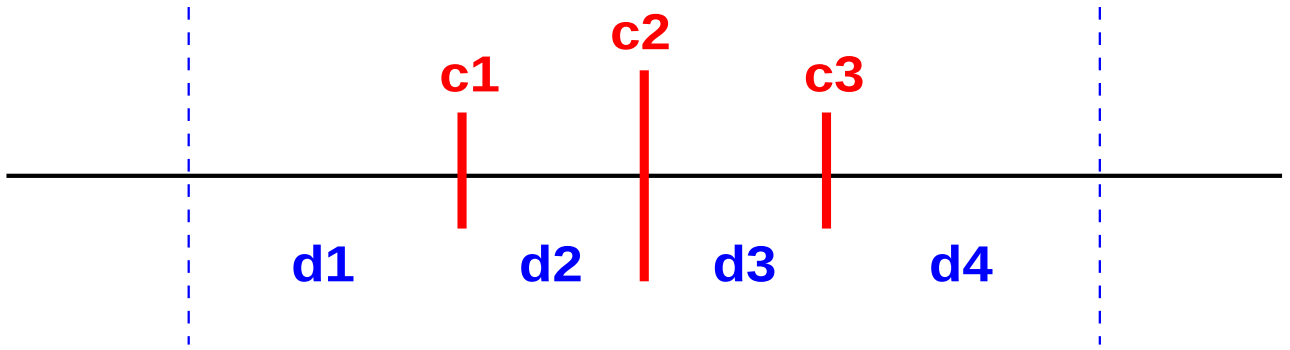
→ Exercises in tutorial

Can we do better ?



- Try more slices, e.g. 3 kicks:
- How to put them ?
- Allow that they are at different positions and have different strengths
- Minimize the inaccuracy

- Try a model with 3 kicks:

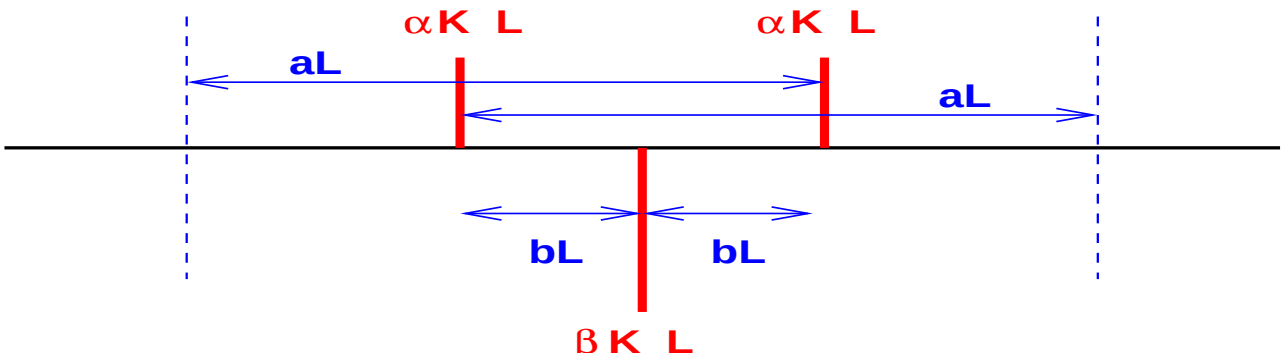


→ To get best accuracy (i.e. deviation from exact solution):

→ You have 7 free parameters to minimize deviation:

- Kicks c1, c2, c3
- Drifts d1, d2, d3, d4

- The optimization gives us:



with:

a ≈ 0.6756, b ≈ -0.1756, $\alpha \approx 1.3512$, $\beta \approx -1.7024$

- We have a $\mathcal{O}(4)$ integrator ...
- (a $\mathcal{O}(6)$ integrator would require 9 kicks (!) ...)

(for the derivation: [AC1])

The lumped matrix M becomes now:

$$\mathcal{M}(O4)_{lumped} = \begin{pmatrix} 1 - \frac{1}{2}k^2L^2 + \frac{1}{24}k^4L^4 & L - \frac{1}{6}k^2L^3 + \frac{1-2^{1/3}}{24(2-2^{1/3})^2}k^4L^5 \\ + \frac{2^{1/3}}{48(2-2^{1/3})^3}k^6L^6 & + \frac{2^{1/3}}{96(2-2^{1/3})^4}k^6L^7 \\ -k^2L + \frac{1}{6}k^4L^3 & 1 - \frac{1}{2}k^2L^2 + \frac{1}{24}k^4L^4 \\ + \frac{2^{1/3}}{24(2-2^{1/3})^2}k^6L^5 & + \frac{2^{1/3}}{48(2-2^{1/3})^3}k^6L^6 \end{pmatrix}$$

Prove that it is symplectic ..

Why all that ? (answer in a few minutes)

Symplectic integration

- What we do is Symplectic Integration
- From a lower order integration scheme (1 kick), construct higher order scheme
- Formally (for the formulation of $S_k(t)$ see later):
  - From a 2nd order scheme (1 kick) $S_2(t)$ we construct a 4th order scheme (3 kicks = 3 x 1 kick) like:

$$S_4(t) = S_2(x_1 t) \circ S_2(x_0 t) \circ S_2(x_1 t) \quad \text{with:}$$

$$x_0 = \frac{-2^{1/3}}{2 - 2^{1/3}} \approx -1.7024 \qquad x_1 = \frac{1}{2 - 2^{1/3}} \approx 1.3512$$

- Can be considered an iterative scheme (see e.g. H. Yoshida, 1990, E. Forest, 1998[2)]):
  - If $S_{2k}(t)$ is a symmetric integrator of order $2k$, then:

    $S_{2k+2}(t) = S_{2k}(x_1 t) \circ S_{2k}(x_0 t) \circ S_{2k}(x_1 t)$ with:

$$x_0 = \frac{-\sqrt[2k+1]{2}}{2 - \sqrt[2k+1]{2}} \qquad x_1 = \frac{1}{2 - \sqrt[2k+1]{2}}$$

- Higher order integrators can be obtained in a similar way

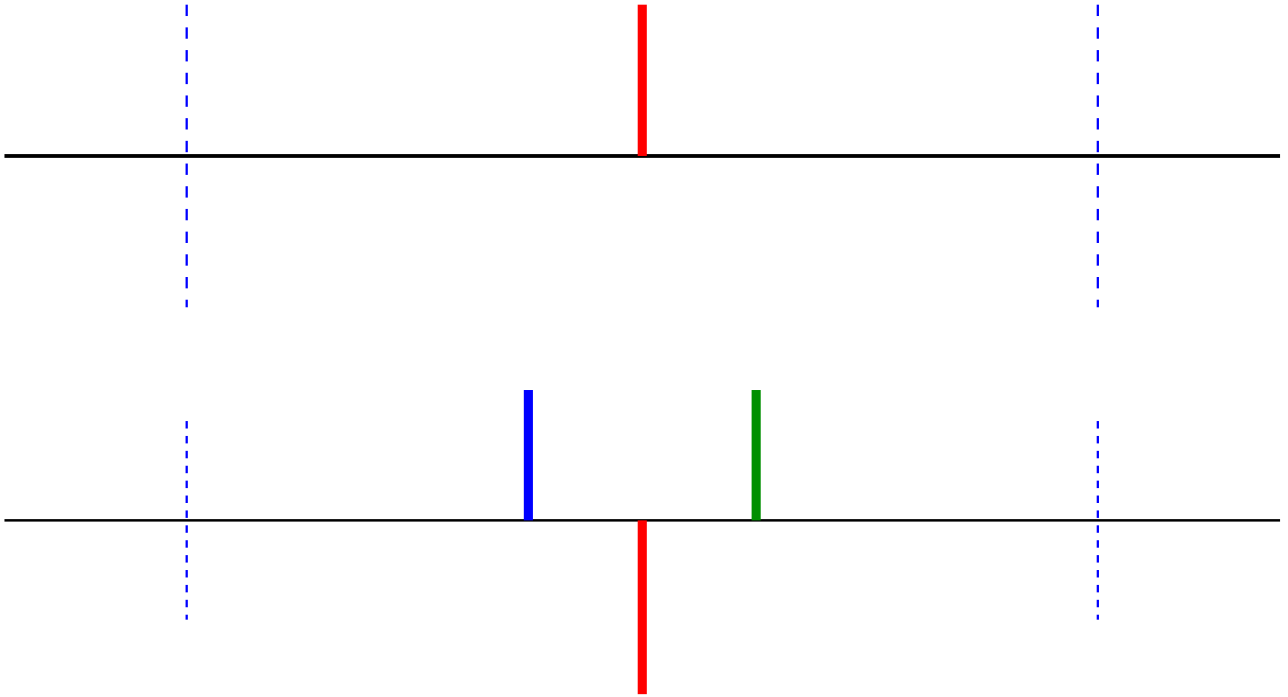
Symplectic integration

- Example: From a 4th order to 6th order

$$S_6(t) = S_4(x_1 t) \circ S_4(x_0 t) \circ S_4(x_1 t)$$

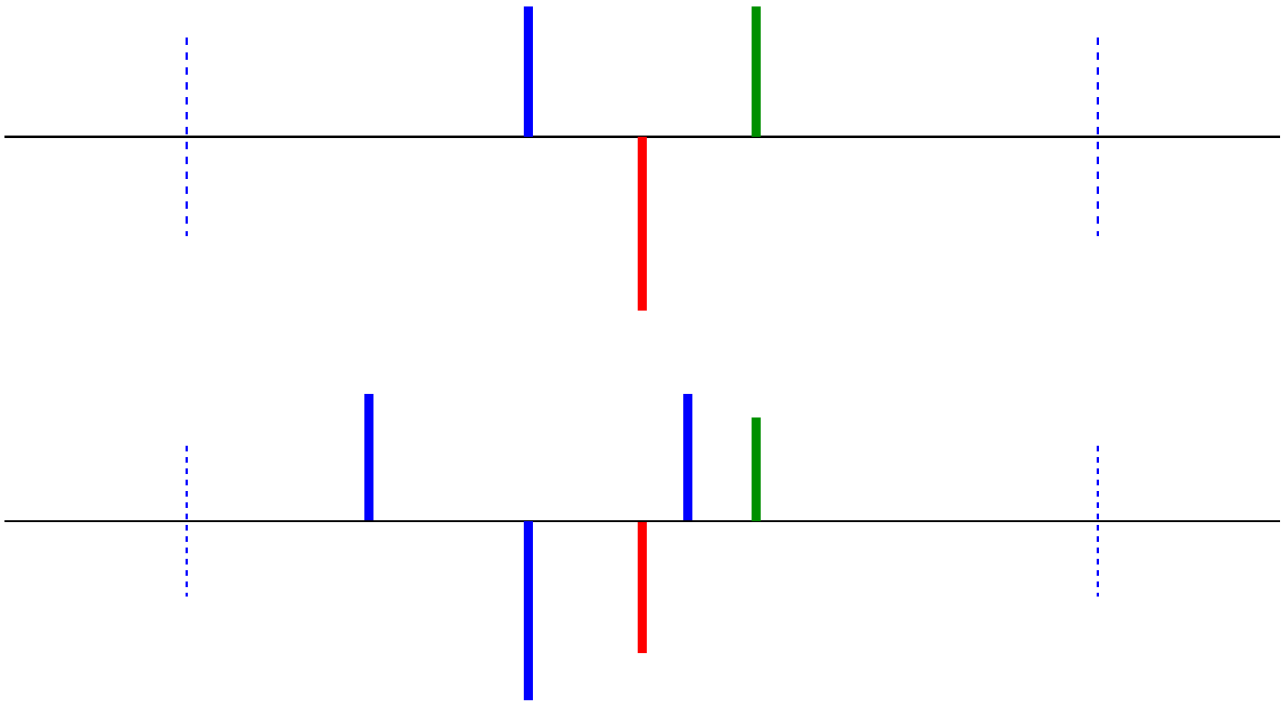
- Replace each kick of a 4th order integrator by a 4th order integrator, using the same scaling factors
- We get 3 times 4th order with 3 kicks each, we have the 9 kick, 6th order integrator mentioned earlier

Integrator of order 2 $\Longrightarrow$ 4



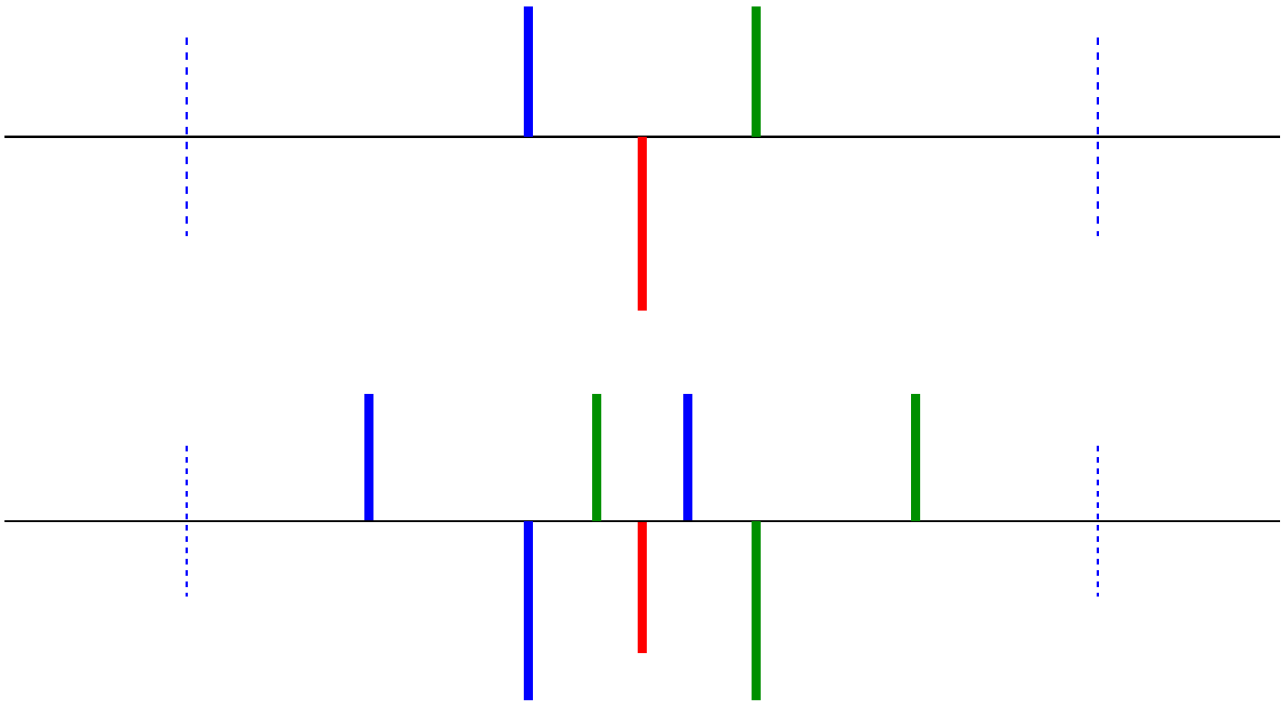
- Replace kick by 4th order integrator

Integrator of order 4 $\Longrightarrow$ 6 - step by step



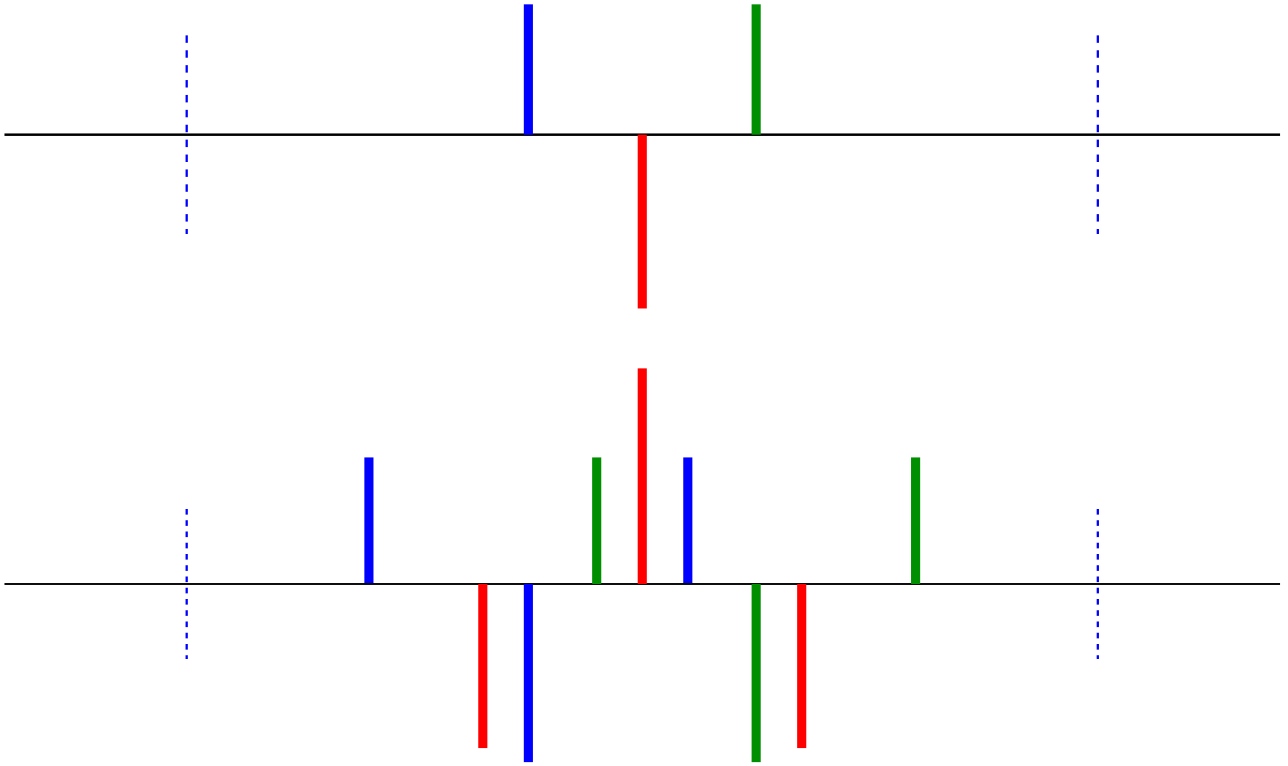
- Replace each kick by 4th order integrator

Integrator of order 4 $\Longrightarrow$ 6 - step by step



- Replace each kick by 4th order integrator

Integrator of order 4 $\Longrightarrow$ 6 - step by step



- Replace each kick by 4th order integrator, requires 9 kicks
- We have 3 interleaved 4th order integrators (compute $\mathcal{M}(\mathcal{O}6)_{lumped}$), Repeat the procedure to go to higher orders

Some remarks:

- We have used a linear map (quadrupole) to demonstrate the integration
- Can that be applied for other maps (solenoids, higher order, non-linear maps) ?
  - Yes (this is the answer) !!
  - We get the same integrators !
  - Proof and systematic (and easy) extension in the form of Lie operators[2)] (see later)

→ Best accuracy for thin lenses !

Accuracy of (non-linear) thin lenses

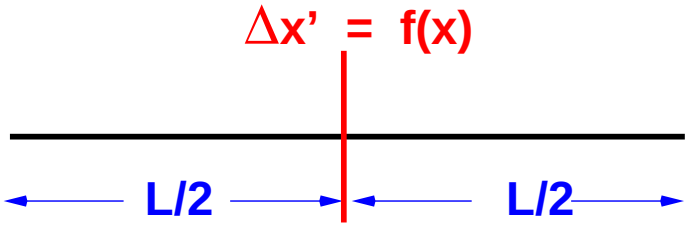
Now the general case with the function f(x):

$$\Delta x' \quad = \quad f(x)$$

- Non-linear elements are usually thin (thinner than dipoles, quadrupoles ...)
  - Dipoles: $\approx$ 14.3 m
  - Quadrupole: $\approx$ 2 - 5 m
  - Sextupoles, Octupoles: $\approx$ 0.30 m

→ Can try our simplest thin lens approximation O(2) first ...

Drift - Kick - Drift



$$1.Step \begin{pmatrix} x \\ x' \end{pmatrix}_{s_1+L/2} = \begin{pmatrix} 1 & \frac{L}{2} \\ 0 & 1 \end{pmatrix} \circ \begin{pmatrix} x_0 \\ x'_0 \end{pmatrix}_{s_1}$$

$$2.Step \begin{pmatrix} x \\ x' \end{pmatrix}_{s_1+L/2} = \begin{pmatrix} x_0 \\ x'_0 + \Delta x' \end{pmatrix}_{s_1+L/2}$$

$$3.Step \begin{pmatrix} x \\ x' \end{pmatrix}_{s_1+L} = \begin{pmatrix} 1 & \frac{L}{2} \\ 0 & 1 \end{pmatrix} \circ \begin{pmatrix} x_0 \\ x'_0 \end{pmatrix}_{s_1+L/2}$$

Putting it together:

$$\begin{aligned} x'(L) &\approx x'_0 + L \cdot f(x_0 + \frac{L}{2} x'_0) \\ x(L) &\approx x_0 + \frac{L}{2} \cdot (x'_0 + x'(L)) \end{aligned}$$

It is symplectic (... and time reversible) !!

Accuracy of thin lenses

the (exact) Taylor expansion of $f(x)$ gives:

$$x(L) \quad = \quad x_0 + x_0' L + \frac{1}{2} f(x_0) L^2 + \frac{1}{6} x_0' f'(x_0) L^3 + \ldots$$

the (approximate) algorithm gives:

$$x(L) \quad = \quad x_0 + x_0' L + \frac{1}{2} f(x_0) L^2 + \frac{1}{4} x_0' f'(x_0) L^3 + \ldots$$

- Errors are $O(L^3)$ (of course, because $O(L^2)$ is correct by construction)
- For small L acceptable, and symplectic

An application, a (1D) sextupole with:

$$f(x) \quad = \quad k \cdot x^2$$

using the thin lens approximation gives:

$$\begin{aligned} x(L) &= x_0 + x_0' L + \frac{1}{2} k x_0^2 L^2 + \frac{1}{2} k x_0 x_0' L^3 + \frac{1}{8} k x_0'^2 L^4 \\ x'(L) &= x_0' + k x_0^2 L + k x_0 x_0' L^2 + \frac{1}{4} k x_0'^2 L^3 \end{aligned}$$

Map for thick sextupole of length L in thin lens approximation, accurate to $O(L^2)$

For bar/coffee discussions (or a question tomorrow):

why did I write:

$$\Delta x' \quad = \quad f(x)$$

and not:

$$\Delta x' \quad = \quad f(x, x')$$

Simulation and tracking

All these tools can be used in a simulation code

- Main purpose of such a code: Propagate particles around a ring or along a beam line. It is the most precise treatment.
- Results (amongst others):
  - Phase space topology (Poincare sections,..)
  - Global properties (after some analysis), e.g. stability, detuning, invariants, frequency map analysis ....
- In our terminology: Tracking codes produce maps (i.e. relate output to the input), it is the most precise map we can get !
- Can we extract more ”analytical” maps ? → tomorrow

”You should remember” in a nutshell...

- Accelerator elements are described by symplectic maps
- Allow computation and analysis of One-Turn-Maps or Single-Pass-Maps
- Normal Form analysis provides all what we need to know
- Thin and thick lenses: when, why, pros and cons
- Symplectic integration provides best accuracy with thin lenses, in particular when a closed expression is not possible

  An integrator can be constructed to high order and accuracy

Next ...

- Extend to non-linear dynamics and introduce modern concepts

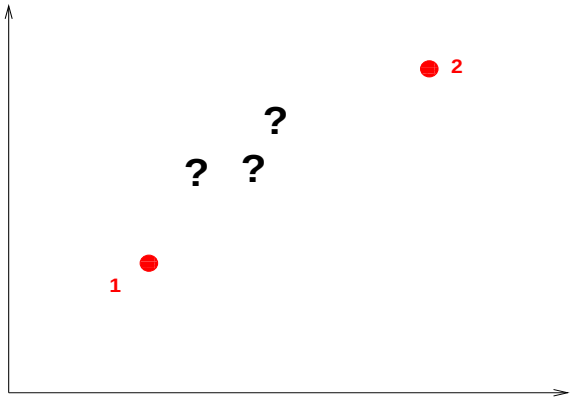
Mathematical and Numerical Methods for Non-linear Beam Dynamics (an introduction)

Part 2

Werner Herr, CERN

- Road map:
  - Lagrangian and Hamiltonian dynamics
  - How to use that → Lie transforms
  - How to analyse that → Non-linear normal forms
  - How to analyse that easier → TPSA
- Avoid abstract definitions and formulation, but:
  - Intuitive (but correct !) treatment
  - Useful formulae and examples
  - Real life examples and demonstration (TPSA)

Problem: what is the motion of a system described by generalized coordinates $\mathbf{q}(t)$ between two specific states $\mathbf{q}_1 = \mathbf{q}(t_1)$ and $\mathbf{q}_2 = \mathbf{q}(t_2)$



coordinates: $\mathbf{q}(t) = q_i(t) \quad (i = 1, n)$

velocities: $\dot{\mathbf{q}}(t) = \dot{q}_i(t) \quad (i = 1, n)$

(Goldstein convention)

n are degrees of freedom of the system, for N particles: $n = 3 \cdot N$

- Describe the particle's motion by a function L (Lagrange function)

$$L(\, q_1(t), \ldots q_n(t),\ \dot{q}_1(t), \ldots \dot{q}_n(t), t\,)$$

$q_1(t), \ldots q_n(t)$ ... generalized coordinates

$\dot{q}_1(t), \ldots \dot{q}_n(t)$ ... generalized velocities

The integral $S = \int L(\, q_i(t), \dot{q}_i(t), t\,)\, dt$ *) defines the action

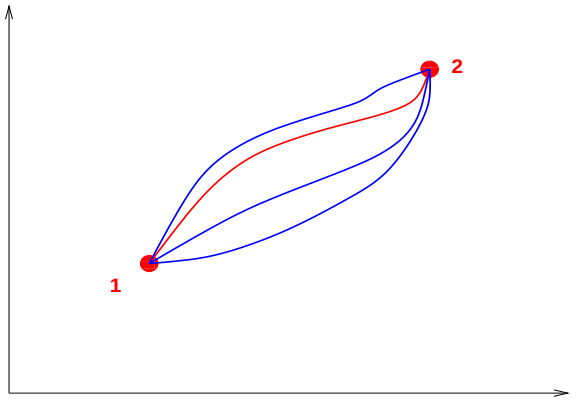
Without proof or derivation:

$S = T - V$ = kinetic energy - potential energy

*) to save typing: $q_i(t),\ \dot{q}_i(t) \implies (q_i, \dot{q}_i)$

Hamilton principle (least action)

$$S = \int_1^2 L(q_i, \dot{q}_i, t)dt = extremum$$



- Hamiltonian principle: system moves such that the action S becomes an extremum

Lagrange formalism

Without proof:

$$S = \int_1^2 L(q_i, \dot{q}_i, t)\, dt = extremum$$

is fulfilled when:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = 0$$

(Euler - Lagrange equation)

From Lagrangian to Hamiltonian ..

- Lagrangian $L(q_1, ...q_n, \dot{q}_1, ...\dot{q}_n, t)$ in generalized coordinates and velocities
- Provides (n) second order differential equations
- Lagrangian is more general, for our purpose we use Hamiltonian (if you are interested why, ask a lecturer or your tutors)
  - First order differential equations (always solvable)
  - Generalized momenta instead of velocities
  - q_j and p_j are independent and on equal footing, q_j and $\dot{q}_j$ are not

The generalized momenta p_j we derive from L as:

$$p_j = \frac{\partial L}{\partial \dot{q}_j}$$

Once we know what the canonical momenta p_i are: the Hamiltonian is a transformation of the Lagrangian:

$$H(q_j, p_j, t) = \sum_i \dot{q}_i p_i - L(q_j, \dot{q}_j, t)$$

without proof:

$H = T + V$ = kinetic energy + potential energy

we obtain 2 first order equation of motion:

$$\frac{\partial H}{\partial q_j} = -\dot{p}_j = -\frac{dp_j}{dt}, \qquad \frac{\partial H}{\partial p_j} = \dot{q}_j = \frac{dq_j}{dt}$$

Now back to beam dynamics .....

Hamiltonian of particle in EM fields

For the Hamiltonian of a (relativistic) particle in an electro-magnetic field we have ($q \rightarrow x$):

$$H(\vec{x}, \vec{p}, t) = c\sqrt{(\vec{p} - e\vec{A}(\vec{x}, t))^2 + m_0^2 c^2} + e\Phi(\vec{x}, t)$$

where $\vec{A}(\vec{x}, t)$, $\Phi(\vec{x}, t)$ the vector and scalar scalar potential

Using canonical variables and the design path length s as independent variable (bending in x-plane) and no electric fields:

$$H = -(1 + \frac{x}{\rho}) \cdot \sqrt{(1+\delta)^2 - p_x^2 - p_y^2} + \frac{x}{\rho} + \frac{x^2}{2\rho^2} - \frac{A_s(x, y)}{B_0\rho}$$

where $\delta = (p_{old} - p_0)/p_0$ is relative momentum deviation and $A_s(x, y)$ longitudinal component of the vector potential [MB].

The magnetic fields can be described with the multipole expansion:

$$B_y + iB_x = \sum_{n=1} (b_n + ia_n)(x + iy)^{n-1}$$

and since $\vec{B} = \nabla \times \vec{A}$:

$$A_s = \sum_{n=1} \frac{1}{n}[(b_n + ia_n)(x + iy)^n]$$

$n = 1$ refers to dipole (not always true, other conventions exist !)

- Looks like what we had before .. what have we gained ?
- For a large machine ($x \ll \rho$) we expand the root (to 2nd order) and sort the variables →

Hamiltonian (for large machine) ..

$$H = \overbrace{\frac{p_x^2 + p_y^2}{2(1+\delta)}}^{kinematic} - \overbrace{\underbrace{\frac{x\delta}{\rho}}_{dispersive} + \underbrace{\frac{x^2}{2\rho^2}}_{focusing}}^{dipole} + \overbrace{\frac{k_1}{2}(x^2 - y^2)}^{quadrupole} + \overbrace{\frac{k_2}{6}(x^3 - 3xy^2)}^{sextupole}$$

$$\left(\text{using (MAD convention)} : \quad k_n = \frac{1}{B\rho}\frac{\partial^n B_y}{\partial x^n} \right)$$

- The Hamiltonian describes exactly the motion of a particle through a magnet
- Basis to extend the linear to a non-linear formalism

But how do we use it ??

Poisson brackets

Introduce Poisson bracket for a differential operator:

$$[f, g] = \sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i} \frac{\partial g}{\partial p_i} - \frac{\partial f}{\partial p_i} \frac{\partial g}{\partial x_i} \right)$$

Here the variables x_i, p_i are canonical variables, f and g are functions of x_i and p_i.

We can now write (using the Hamiltonian H for g):

$$f(x_i, p_i) = x_i \Rightarrow \quad [x_i, H] = \frac{\partial H}{\partial p_i} = \frac{dx_i}{dt}$$

$$f(x_i, p_i) = p_i \Rightarrow \quad [p_i, H] = -\frac{\partial H}{\partial x_i} = \frac{dp_i}{dt}$$

Poisson brackets encode Hamiltons's equations

Poisson bracket $[f, H]$ describes the time evolution of the function f

It is a special case of:

$$\frac{df}{dt} = [f, H] + \frac{\partial f}{\partial t}$$

If f does not explicitly depend on time ($\frac{\partial f}{\partial t} = 0$) and:

$$[f, H] = 0 \quad \Longrightarrow \quad \frac{df}{dt} = 0$$

implies that f is an invariant (constant) of the motion !

Poisson brackets with H determine invariants

We continue now again with s as the independent parameter.

We can define: $: f : g = [f, g]$

where $: f :$ is an operator acting on the function g:

$$: f := [f,]$$

The operator $: f :$ is called a Lie Operator

Lie operators are Poisson brackets in waiting

It acts on functions $g(x, p)$, look at special cases:

$$g(x, p) = x \quad \rightarrow \quad [f, \ x] \ = \ : f : x$$

$$g(x, p) = p \quad \rightarrow \quad [f, \ p] \ = \ : f : p$$

Useful formulae for calculations (and exercises)

With x coordinate, p momentum, try special cases for f:

$$: x : \;=\; \frac{\partial}{\partial p} \qquad : p : \;=\; -\frac{\partial}{\partial x}$$

$$: x :^2 \;=\; \frac{\partial^2}{\partial p^2} \qquad : p :^2 \;=\; \frac{\partial^2}{\partial x^2}$$

$$: x^2 : \;=\; 2x\frac{\partial}{\partial p} \qquad : p^2 : \;=\; -2p\frac{\partial}{\partial x}$$

$$: xp : \;=\; p\frac{\partial}{\partial p} - x\frac{\partial}{\partial x} \qquad : x :: p : \;=\; : p :: x : \;=\; -\frac{\partial^2}{\partial x \partial p}$$

More useful formulae for calculations

With x coordinate, p momentum:

$$: p^2 : x \;=\; \frac{\partial p^2}{\partial x}\frac{\partial x}{\partial p} - \frac{\partial p^2}{\partial p}\frac{\partial x}{\partial x} \;=\; -2p$$

$$: p^2 : p \;=\; \frac{\partial p^2}{\partial x}\frac{\partial p}{\partial p} - \frac{\partial p^2}{\partial p}\frac{\partial p}{\partial x} \;=\; 0$$

$$(: p^2 :)^2 x =: p^2 : (: p^2 : x) =: p^2 : (-2p) = 0$$

$$(: p^2 :)^2 p =: p^2 : (: p^2 : p) =: p^2 : (0) = 0$$

Next step → Lie transformations

We can define powers as:

$$(: f :)^2 g =: f : (: f : g) = [f, [f, g]] \quad \text{etc.}$$

in particular:

$$e^{:f:} = \sum_{i=0}^{\infty} \frac{1}{i!} (: f :)^i$$

$$e^{:f:} = 1 + : f : + \frac{1}{2!}(: f :)^2 + \frac{1}{3!}(: f :)^3 + \ldots$$

The operator $e^{:f:}$ is called a Lie Transformation

TO REMEMBER (if anything): what's the difference ?

- Very F.A.Q. : difference between Lie Operator and Lie Transformation ?

  $: f : g$ describes time evolution of g over an infinitesimal distance

  $e^{:f:}g$ describes time evolution of g over a finite distance

Lie transformations are symplectic for thick elements !

- For coffee/bar discussion (and ask your tutors):

What is the meaning of this expression ?

$$[\rho, H] = 0$$

(ρ is particle density in phase space)

Lie transformations - general

Acting on the phase space coordinates:

$$e^{:f:}(x,p)_0 = (x,p)_1$$

that is:

$$e^{:f:}x_0 = x_1$$

$$e^{:f:}p_0 = p_1$$

- Lie transforms describe how to go from one point to another [AC1, AD].
- Through a machine element (drift, magnet ...) fully described by f
- But what is f ?

Lie transformations

- The generator f is the Hamiltonian H of the element !
- We use the Hamiltonian to describe the motion through an individual element
- Inside a single element the motion is ”smooth” (in the full machine it is not !)
- Treats ”thick” elements ($L > 0$) and is still symplectic !

  For an element of length L: $f = L \cdot H$

Back to maps:

we relate a non-linear map $\mathcal{M}$ to:

$$\mathcal{M} \quad = \quad e^{:f:}$$

note, we also have with this formulation:

$$\mathcal{M}^{-1} \quad = \quad e^{-:f:}$$

the proof is rather trivial, see in a few minutes

A (most) important feature !

Assume we have the map:

$$\mathcal{M} \;=\; e^{:f:}$$

we can write it in a different form (factorization):

$$e^{:f:} \;=\; e^{:f_2:} e^{:f_3:} e^{:f_4:} \ldots$$

Here f_k are power series of k-th order.

The miracle:

since all exponential maps are symplectic, one can truncate the factorized map at any order k ... and it remains symplectic !!

Remember: truncated Taylor maps are not symplectic !

Lie transformations - example

Lie operators act on functions like x, p (canonical momentum, instead of x'), for example:

$$: p^2 :x = -2p \qquad : p^2 :p = 0$$

or as a Lie transformation with $f = -Lp^2/2$:

$$e^{:-Lp^2/2:}x = x - \frac{1}{2}L\underbrace{: p^2 : x}_{=-2p} + \frac{1}{8}L^2\underbrace{(: p^2 :)^2 x}_{=0} + ..$$

$$= \boxed{x + Lp}$$

$$e^{:-Lp^2/2:}p = p - \frac{1}{2}L\underbrace{: p^2 : p}_{=0} + ...$$

$$= \boxed{p}$$

Lie transformations - example

Lie operators act on functions like x, p (canonical momentum, instead of x'), for example:

$$: p^2 :x = -2p \qquad : p^2 :p = 0$$

or as a Lie transformation with $f = -Lp^2/2$:

$$e^{:-Lp^2/2:}x = x - \frac{1}{2}L\underbrace{: p^2 : x}_{=-2p} + \frac{1}{8}L^2\underbrace{(: p^2 :)^2 x}_{=0} + ..$$

$$= \boxed{x + Lp}$$

$$e^{:-Lp^2/2:}p = p - \frac{1}{2}L\underbrace{: p^2 : p}_{=0} + ...$$

$$= \boxed{p}$$

This is the transformation of a drift space of length L !!

Drift space - for the ambitious

The full and exact Hamiltonian in two transverse dimensions and with a relative momentum deviation δ is:

$$H = -\sqrt{(1+\delta)^2 - p_x^2 - p_y^2}$$

The map for a drift space is now:

$$\begin{aligned}
x^{new} &= x + L \cdot \frac{p_x}{\sqrt{(1+\delta)^2 - p_x^2 - p_y^2}} \\
p_x^{new} &= p_x \\
y^{new} &= y + L \cdot \frac{p_y}{\sqrt{(1+\delta)^2 - p_x^2 - p_y^2}} \\
p_y^{new} &= p_y
\end{aligned}$$

In 2D and with $\delta \neq 0$ it is a complicated beast ! And it creates chromaticity !!

In practice the map can (usually) be simplified to the well known form.

Some formulae for Lie transforms

With a constant, f, g, h arbitrary functions:

$$: a : \;=\; 0 \qquad \longrightarrow \qquad e^{:a:} \;=\; 1$$

$$: f : a \;=\; 0 \qquad \longrightarrow \qquad e^{:f:} a \;=\; a$$

$$e^{:f:} g(x) \;=\; g(e^{:f:} x)$$

$$e^{:f:} G(: g :) e^{-:f:} \;=\; G(: e^{:f:} g :)$$

$$e^{:f:} [g, h] = [e^{:f:} g, e^{:f:} h]$$

$$(e^{:f:})^{-1} \;=\; e^{-:f:}$$

and very important:

$$e^{:f:} e^{:g:} e^{-:f:} \;=\; e^{:e^{:f:} g:}$$

More examples (1D):

For:

$$f = -\frac{L}{2}(kx^2 + p^2)$$

we write for the transformation (map):

$$\begin{aligned} e^{:f:}x &= e^{:-\frac{L}{2}(kx^2+p^2):}x \\ e^{:f:}p &= e^{:-\frac{L}{2}(kx^2+p^2):}p \end{aligned}$$

Remember:

$$e^{:f:}g = \sum_{n=0}^{\infty} \frac{:f:^n}{n!} g$$

we would get (try it, using the ”useful formulae” and some intelligent sorting !):

$$
\begin{aligned}
e^{:-\frac{L}{2}(kx^2+p^2):} x &= \sum_{n=0}^{\infty} \left(\frac{(-kL^2)^{2n}}{(2n)!} \cdot x + L \frac{(-kL^2)^{2n+1}}{(2n+1)!} \cdot p \right) \\
e^{:-\frac{L}{2}(kx^2+p^2):} p &= \sum_{n=0}^{\infty} \left(\frac{(-kL^2)^{2n}}{(2n)!} \cdot p - k \frac{(-kL^2)^{2n+1}}{(2n+1)!} \cdot x \right)
\end{aligned}
$$

Looks familiar:

Starting from:

$$f = -\frac{L}{2}(kx^2 + p^2)$$

we finally have obtained:

$$\begin{aligned} e^{:f:}x &= x\cos(\sqrt{k}L) + \frac{p}{\sqrt{k}}\sin(\sqrt{k}L) \\ e^{:f:}p &= -\sqrt{k}x\sin(\sqrt{k}L) + p\cos(\sqrt{k}L) \end{aligned}$$

→ Thick, focusing quadrupole !

A little riddle, what if we have $f = -\frac{1}{2}kx^2$

Hamiltonians of some thick machine elements (3D)

dipole:

$$H = -\frac{-x\delta}{\rho} + \frac{x^2}{2\rho^2} + \frac{p_x^2 + p_y^2}{2(1+\delta)}$$

quadrupole:

$$H = \frac{1}{2}k_1(x^2 - y^2) + \frac{p_x^2 + p_y^2}{2(1+\delta)}$$

sextupole:

$$H = \frac{1}{6}k_2(x^3 - 3xy^2) + \frac{p_x^2 + p_y^2}{2(1+\delta)}$$

octupole:

$$H = \frac{1}{4}k_3(x^4 - 6x^2y^2 + y^4) + \frac{p_x^2 + p_y^2}{2(1+\delta)}$$

Remark:

In many cases the non-linear effects by the kinematic term are negligible and

$$H = \frac{1}{2}k_1(x^2 - y^2) + \frac{p_x^2 + p_y^2}{2(1+\delta)}$$

is written as:

$$H = \frac{1}{2}k_1(x^2 - y^2) + \frac{p_x^2 + p_y^2}{2}$$

In 1D it reduces to previous example

$$f = -\frac{L}{2}(kx^2 + p^2) = -L \cdot H$$

Why all that ???

If we know the Hamiltonian H of a machine element then:

$$e^{:H:} x_0 = x_1$$

$$e^{:H:} p_0 = p_1$$

This is also true for any function of x and p:

$$e^{:H:} f_0(x,p) = f_1(x,p) \qquad \text{e.g. : } x^2, \quad x \cdot p, \ldots$$

- The miracles:
  - Poisson brackets create symplectic maps
  - Exponential form $e^{:H:}$ is always symplectic
  - H can be very complicated, any non-linear contraption

A very good reason to use canonical variabls !!

Many machine elements

- We can combine N machine elements m_n by applying one transformation after the other:

$$e^{:h:} = e^{:m_1:} e^{:m_2:} .... e^{:m_N:}$$

- Not restricted to matrices, i.e. linear elements ...
- And arrive at a transformation for the full ring
  - → a one turn map, a ring with circumference C
- The one turn map is the exponential of the effective Hamiltonian:

$$\mathcal{M}_{ring} = e^{:-CH_{eff}:}$$

concatenation very easy:

$$e^{:h:} = e^{:f:} e^{:g:} = e^{:f+g:}$$

when f and g commute (i.e. $[f, g] = [g, f] = 0$)

Otherwise formalism exist

Concatenation

To combine:

$$e^{:h:} = e^{:f:} e^{:g:}$$

We can use the formula (Baker-Campbell-Hausdorff (BCH)):

$$\begin{aligned} h = f \quad & + g + \tfrac{1}{2}[f,g] + \tfrac{1}{12}[f,[f,g]] + \tfrac{1}{12}[g,[g,f]] \\ & + \tfrac{1}{24}[f,[g,[g,f]]] - \tfrac{1}{720}[g,[g,[g,[g,f]]]] \\ & - \tfrac{1}{720}[f,[f,[f,[f,g]]]] + \tfrac{1}{360}[g,[f,[f,[f,g]]]] + \ldots \end{aligned}$$

or :

$$\begin{aligned} h = f \quad & + g + \tfrac{1}{2} : f : g + \tfrac{1}{12} : f :^2 g + \tfrac{1}{12} : g :^2 f \\ & + \tfrac{1}{24} : f :: g :^2 f - \tfrac{1}{720} : g :^4 f \\ & - \tfrac{1}{720} : f :^4 g + \tfrac{1}{360} : g :: f :^3 g + \ldots \end{aligned}$$

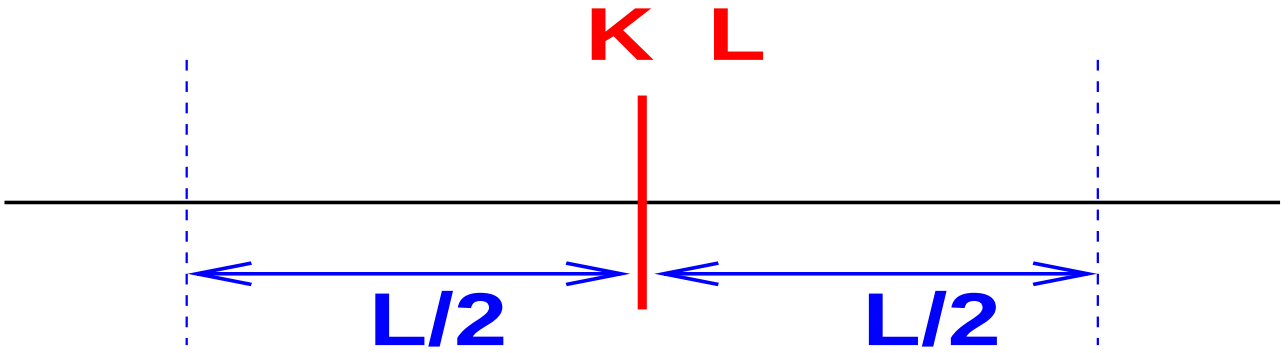
To combine in a (often applicable) case:

$$e^{:h:} = e^{:f:} e^{:g:}$$

if one of them (f or g) is small, can truncate the series and get a very useful formula. Assume g is small:

$$e^{:f:}\ e^{:g:} \ = \ e^{:h:} \ = \ \exp\left[: f + \left(\frac{: f :}{1 - e^{-:f:}} \right) g + \mathcal{O}(g^2) : \right]$$

Side note:



Would be generated by a Hamiltonian:

$$H = \tfrac{1}{2}H_{drift} + H_{kick} + \tfrac{1}{2}H_{drift}$$

$$e^{-L:\frac{1}{2}H_{drift}+H_{kick}+\frac{1}{2}H_{drift}:}$$

We can use this to construct our integrators with kicks $f(x)$

Non-linear kicks

Assume a general thin lens kick $f(x)$ i.e. for the map:

$$\begin{aligned} x &= x_0 \\ p &= p_0 + f(x) \end{aligned}$$

then the Lie transform is obtained by:

$$e^{:\int_0^x f(x')dx':}$$

Example: thin lens multipole of order n ($f(x) = a \cdot x^n$):

$$e^{:\frac{a}{n+1} \cdot x^{n+1}:}$$

Extension: general monomials

Useful for many calculations

Monomials in x and p of orders n and m ($x^n p^m$)

$$e^{:ax^n p^m:}$$

gives for the map (for $n \neq m$):

$$\begin{aligned} e^{:ax^n p^m:} x &= x \cdot [1 + a(n-m)x^{n-1} p^{m-1}]^{m/(m-n)} \\ e^{:ax^n p^m:} p &= p \cdot [1 + a(n-m)x^{n-1} p^{m-1}]^{n/(n-m)} \end{aligned}$$

gives for the map (for $n = m$):

$$\begin{aligned} e^{:ax^n p^n:} x &= x \cdot e^{-anx^{n-1} p^{n-1}} \\ e^{:ax^n p^n:} p &= p \cdot e^{anx^{n-1} p^{n-1}} \end{aligned}$$

A very important example ...

$$\mathcal{M} = \begin{pmatrix} \cos\mu + \alpha sin(\mu) & \beta \sin\mu \\ -\gamma \sin\mu & \cos\mu - \alpha sin(\mu) \end{pmatrix} \Rightarrow \begin{pmatrix} \cos\mu & \sin\mu \\ -\sin\mu & \cos\mu \end{pmatrix} = \mathcal{R}$$

corresponds to:

$$e^{:h:} = e^{:S_1:} = e^{:-\mu\frac{1}{2}(\gamma x^2 + 2\alpha xp + \beta p^2):}$$

Written in our normal (simple) form, i.e. with the invariant J_x :

$$e^{:h:} = e^{:S_1:} = e^{:-\mu J_x:}$$

S_1 produces the linear transport matrix $\mathcal{M}$...

For a one-turn-matrix: μ is $2\pi \cdot Q$

For a 3D linear system we have for S_2:

$$S_2 = -\frac{\mu_x}{2}(x^2 + p_x^2) - \frac{\mu_y}{2}(y^2 + p_y^2) - \frac{1}{2}\alpha_c\delta^2$$

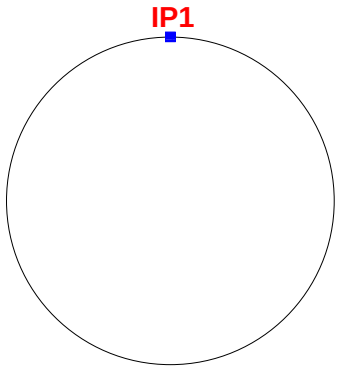
or in action variables J:

$$S_2 = -\mu_x J_x - \mu_y J_y - \frac{1}{2}\alpha_c\delta^2$$

whenever you have 1 local, non-linear object NL, you can express the whole machine as:

$$e^{:S_2:}\, e^{:NL:} = e^{:h:} \qquad \text{(or :} \quad e^{:\frac{1}{2}S_2:}\, e^{:NL:} e^{:\frac{1}{2}S_2:} \quad \text{etc.)}$$

A real life example: beam-beam interaction



- Linear beam transport around the machine
- Beam-beam interaction localized and very non-linear
- Important to understand stability

Our questions ?

- How does the particles behave in phase space ?
- Do we have an invariant and how to compute it ??

We look for invariants - start with single IP

Here in 1D, same treatment for higher dimensions

Linear transfer around the machine $e^{:S_1:}$ and beam-beam interaction $e^{:B:}$

It is factorized into the two parts (see before):

$$e^{:S_1:}\, e^{:B:} \;=\; e^{:h:}$$

with (see before):

$$S_1 \;=\; -\frac{\mu}{2}\left(\frac{x^2}{\beta} + \beta p_x^2\right) \;=\; \mu \cdot\; J_x$$

with the usual transformation to action - angle variables

$$x = \sqrt{2J\beta}\sin\Psi, \qquad p = \sqrt{\frac{2J}{\beta}}\cos\Psi$$

Beam-Beam part B(x):

For a Gaussian beam we have for for the kick $b(x)$ of the beam-beam "lens" (see lecture on beam-beam effects: T. Pieloni, next week):

$$b(x) = \frac{2}{x}(1 - e^{\frac{-x^2}{2\sigma^2}})$$

which can be directly derived from the fields

For the generator we get as defined before:

$$B(x) = \int_0^x dx' b(x')$$

and written as Fourier series (will soon be clear why):

$$B(x) = \sum_{n=-\infty}^{\infty} c_n(J) e^{in\Psi}$$

We evaluate the expression (because the beam-beam part is much smaller than the rest of the machine, see also earlier slide):

$$e^{:S_1:}\, e^{:B:} \;=\; e^{:h:} \;=\; \exp\left[: S_1 + \left(\frac{: S_1 :}{1 - e^{-:S_1:}}\right) B + O(B^2) :\right]$$

To do that we can now use some useful properties of Lie operators (as mentioned before or any textbook e.g. [AW, AC1, WH, AD]):

$$: S_1 : g(J) = 0, \qquad : S_1 : e^{in\Psi} = in\mu e^{in\Psi}, \qquad g(: S_1 :) e^{in\Psi} = g(in\mu) e^{in\Psi}$$

gives immediately for h:

$$h = -\mu J + \left(\sum_n c_n(J) \frac{in\mu}{1 - e^{-in\mu}} e^{in\Psi} \right)$$

or written differently:

$$h = -\mu J + \left(\sum_n c_n(J) \frac{n\mu}{2\sin(\frac{n\mu}{2})} e^{(in\Psi + i\frac{n\mu}{2})} \right)$$

[Nota bene : $\frac{dc_0(J)}{dJ}$ is the amplitude dependent detuning]

Some inspection - analysis of h

$$h = \underbrace{-\mu J}_{linear} + \sum_n c_n(J) \frac{n\mu}{2\sin(\frac{n\mu}{2})} e^{(in\Psi + i\frac{n\mu}{2})}$$

On resonance:

$$Q = \frac{p}{n} = \frac{\mu}{2\pi}$$

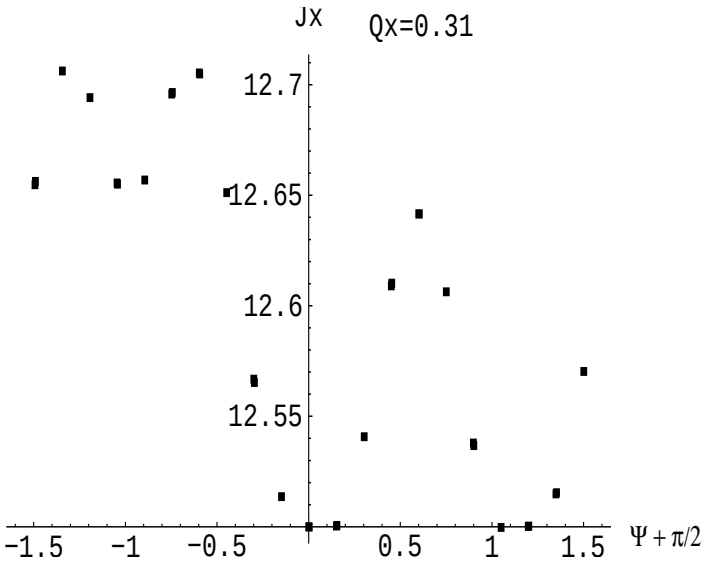
with $c_n \neq 0$:

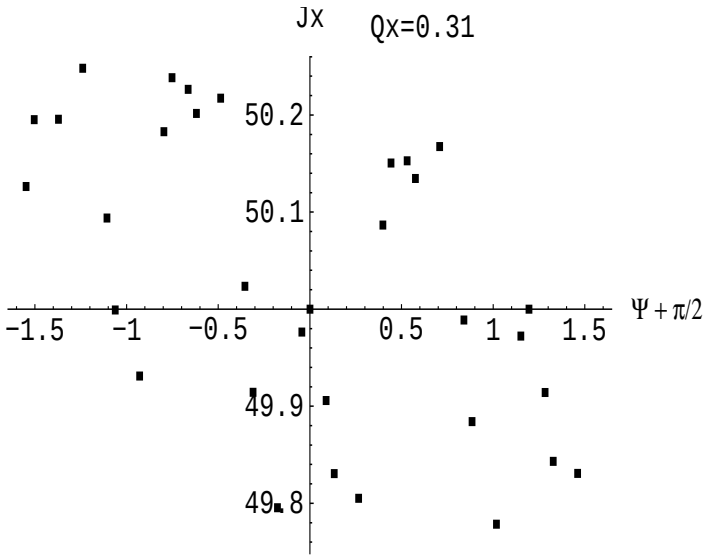
$$\sin(\frac{n\pi p}{n}) = \sin(p\pi) \equiv 0 \quad \forall \ integer \quad p$$

and h diverges, find automatically all resonance conditions

Can we also reproduce the distorted phase space (in action angle variables) ?

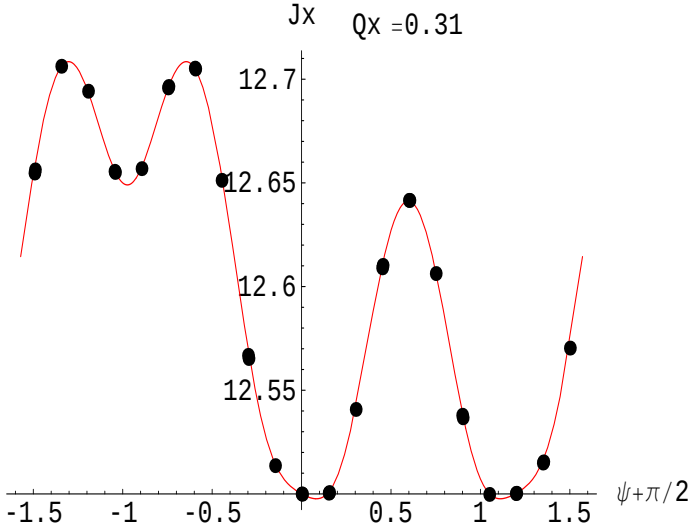
Invariant from tracking: one IP

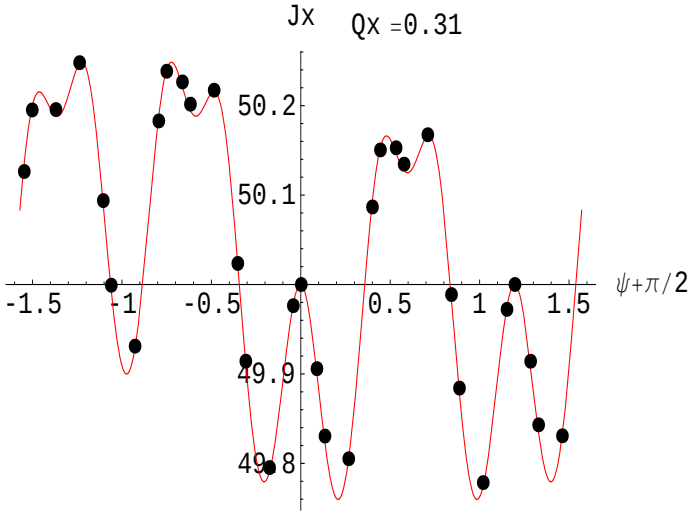




→ Shown for particle amplitudes of $5\sigma_x$ and $10\sigma_x$

Invariant versus tracking: one IP





→ Shown for particle amplitudes of $5\sigma_x$ and $10\sigma_x$

(try that with other methods ...)

First summary: Lie transforms and integrators

- We have powerful tools to describe non-linear elements
- They are always symplectic !
- Can be combined to form a ring (and therefore a non-linear One-Turn-Map)
- Tools and programs are available for their manipulation and computation
- How do we analyse the maps ? Guess: Normal Forms

Normal forms non-linear case

Normal form transformations can be generalized for non-linear maps. If $\mathcal{M}$ is our usual one-turn-map, we try to find a transformation:

$$\mathcal{N} = \mathcal{A}\mathcal{M}\mathcal{A}^{-1}$$

- where $\mathcal{N}$ is a simple form (like the rotation we had before)

Of course we now do not have matrices, we use a Lie transform F to describe the transform $\mathcal{A}$:

$$\mathcal{N} = e^{-:h:} = \mathcal{A}\mathcal{M}\mathcal{A}^{-1} = e^{:F:}\mathcal{M}e^{-:F:}$$

The matrices $\mathcal{A}$ and $\mathcal{A}^{-1}$ describe the transformation between the ”ideal” and ”real” motion.

Side note: beam lines

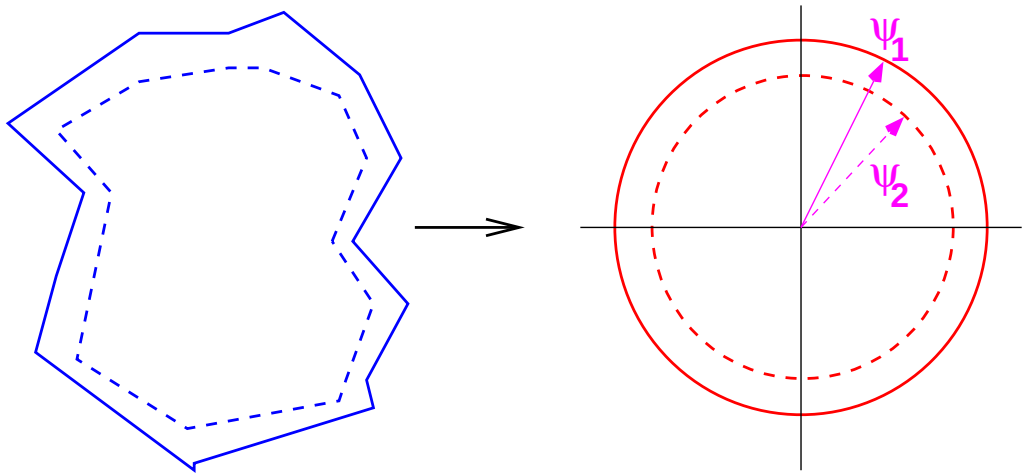
The same strategy can be used for beam lines or any single pass machine: If $\mathcal{M}$ is total transfer map, the we find a transformation:

$$\mathcal{N} = \mathcal{M}\mathcal{A}$$

- Again $\mathcal{N}$ describes the wanted behaviour
- The map $\mathcal{A}$ the deviation (aberrations)
- One can add correctors (i.e. their maps) into $\mathcal{M}$ to minimise these aberrations

  (ditto for circular machines, example in ”Non-linear Dynamics” lecture)

Back to one-turn-maps



- More complicated transformation F required
- Transform to coordinates where map is just a rotation
- In general better done in action angle variables: J, Ψ
- Rotation angle may be amplitude dependent: $\Psi \rightarrow \Psi(J)$

The canonical transformation $\mathcal{A}$:

$$\mathcal{N} = \mathcal{A}\mathcal{M}\mathcal{A}^{-1} \quad \Rightarrow \quad \mathcal{A} = e^{:F:}$$

should be the transformation to produce our simple form

- $h(J_x, \Psi_x, J_y, \Psi_y, z, \delta) \Rightarrow h(J_x, J_y, \delta) = h_{eff}(J_x, J_y, \delta)$
- It depends only on (J_x, J_y) → exact solution !
- Once we know $h_{eff}(J_x, J_y, \delta)$ we can derive everything !
- Formalism and software tools exist to find F (see e.g. Chao[1)] or E.Forest, M. Berz, J. Irwin, SSC-166)

[1)] A. Chao, Lecture Notes on Topics in Accelerator Physics, 2001

Normal forms - non-linear case

Once we can write the map as (now example in 3D):

$$\mathcal{N} = e^{-:h_{eff}(J_x,J_y,\delta):}$$

where h_{eff} depends only on J_x, J_x, and δ, then we have the tunes:

$$Q_x(J_x, J_y, \delta) = \frac{1}{2\pi}\frac{\partial h_{eff}}{\partial J_x}$$

$$Q_y(J_x, J_y, \delta) = \frac{1}{2\pi}\frac{\partial h_{eff}}{\partial J_y}$$

and the change of path length:

$$\Delta z = -\frac{\partial h_{eff}}{\partial \delta}$$

Particles with different J_x, J_y and δ have different tunes:

→ Dependence on J is amplitude detuning, dependence on δ are the chromaticities !

How does h_{eff} look like ?

The effective Hamiltonian can be written (here to 3rd order) as:

$$\begin{aligned} h_{eff} = & + \mu_x J_x + \mu_y J_y + \frac{1}{2}\alpha_c \delta^2 \\ & + c_{x1} J_x \delta + c_{y1} J_y \delta + c_3 \delta^3 \\ & + c_{xx} J_x^2 + c_{xy} J_x J_y + c_{yy} J_y^2 + c_{x2} J_x \delta^2 + c_{y2} J_y \delta^2 + c_4 \delta^4 \end{aligned}$$

and then:

$$Q_x(J_x, J_y, \delta) = \frac{1}{2\pi}\frac{\partial h_{eff}}{\partial J_x} = \frac{1}{2\pi}\left(\mu_x + 2c_{xx}J_x + c_{xy}J_y + c_{x1}\delta + c_{x2}\delta^2\right)$$

$$Q_y(J_x, J_y, \delta) = \frac{1}{2\pi}\frac{\partial h_{eff}}{\partial J_y} = \frac{1}{2\pi}\left(\mu_y + 2c_{yy}J_y + c_{xy}J_x + c_{y1}\delta + c_{y2}\delta^2\right)$$

$$\Delta z = -\frac{\partial h_{eff}}{\partial \delta} = \alpha_c \delta + 3c_3\delta^2 + 4c_4\delta^3 + c_{x1}J_x + c_{y1}J_y + 2c_{x2}J_x\delta + 2c_{y2}J_y\delta$$

What's the meaning of it ?

- μ_x, μ_y: tunes
- $\frac{1}{2}\alpha_c, c_3, c_4$: linear and non-linear "momentum compaction"
- c_{x1}, c_{y1}: first order chromaticities
- c_{x2}, c_{y2}: second order chromaticities
- c_{xx}, c_{xy}, c_{yy}: detuning with amplitude

The coefficients are the various aberrations of the optics

How to deal with it ?

Try to find the effective Hamiltonian (analytically)

- Detailed analysis for magnetic multipoles, see lectures by Y. Papaphilippou

  (sextupole, octupole, ..)

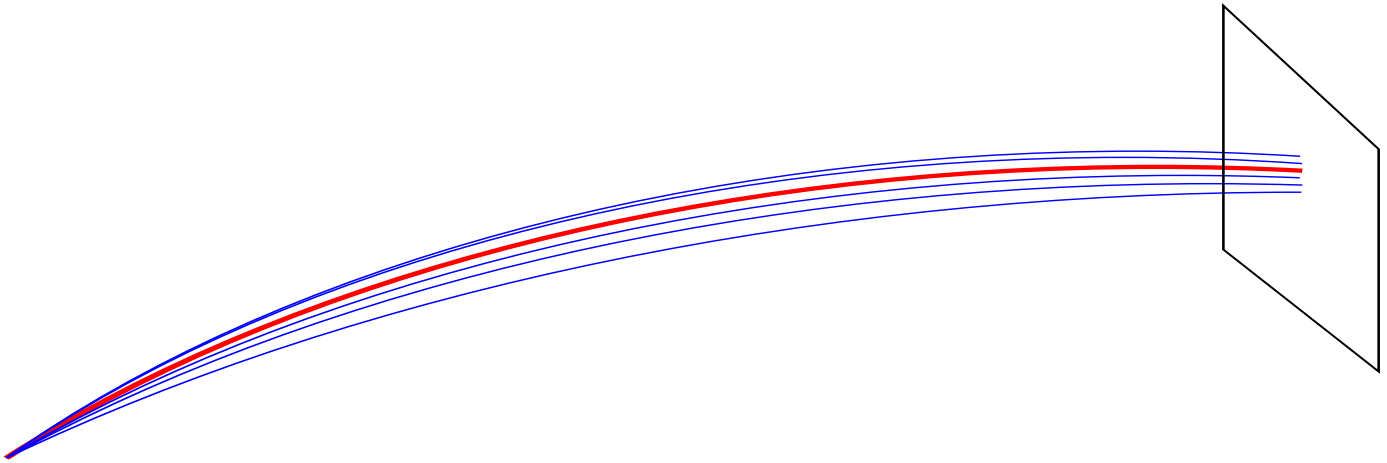
- What about complicated arrangement, e.g. as simulated by a computer program ?

- Tracking particles with a computer code is the most reliable (and flexible) method

- But can we get an ”effective Hamiltonian” for a huge and messy computer code ?

Yes ! It is possible ! We have the tools !

Let us study the paraxial behaviour:



- Red line is the particle trajectory (we have tracked that one)
- Blue lines are small deviations (i.e. close particles)
- If we understand how small deviations behave, we understand the system much better

Now remember the definition of the Taylor series:

$$f(a+\Delta x) = f(a) + \sum_{n=1}^{\infty} \frac{f^{(n)}(a)}{n!}\Delta x^n$$

$$f(a+\Delta x) = f(a) + \frac{f'(a)}{1!}\Delta x^1 + \frac{f''(a)}{2!}\Delta x^2 + \frac{f'''(a)}{3!}\Delta x^3 + \ldots$$

- The coefficients determine the behaviour of small deviations Δx from the ideal orbit a
- $f(a)$ is the ”tracked particle”, $f(a+\Delta x)$ is a ”close particle”

Why are Taylor series useful ?

If we truncate the expansion to the m-th order:

$$f(a+\Delta x) = f(a) + \sum_{n=1}^{m} \frac{f^{(n)}(a)}{n!} \Delta x^n$$

- We can represent the function $f(x)$ by the vector
  $(f(a),\ f'(a),\ f''(a),\ ..., f^{(m)}(a))$
- This vector is what we need to understand the behaviour around a
- We have now an analytical solution for our problem
- How to get these coefficients (derivatives) without much work ?

Numerical differentiation

The problem getting the derivatives $f^{(n)}(a)$ of $f(x)$ at a:

$$f'(a) = \frac{f(a + \Delta x) - f(a)}{\Delta x}$$

- Need to subtract almost equal numbers and divide by small number.
- For higher orders f'', f'''.., accuracy hopeless !
- The way out: we use Truncated Power Series Algebra (TPSA) (M. Berz, 1988 and [MB])

Truncated Power Series Algebra

- The tracking of a particle in a complicated system relates the output numerically to the input:

$$z_1 = (x, p_x, y, p_y, s, \delta)_1 \overset{tracking}{\Longrightarrow} z_2 = (x, p_x, y, p_y, s, \delta)_2$$

- Can a tracking code produce a more ”analytic” one-turn-map ??
- For example coefficients of a Taylor series ?

$$z_1 = (x, p_x, y, p_y, s, \delta)_1 \overset{???}{\Longrightarrow} z_2 = \sum C_j(a) z_1^j = \sum f^{(n)}(a) z_1^j$$

→ The answer is automatic differentiation

Automatic Differentiation

1. Define a pair (q_0, q_1), with q_0, q_1 real numbers

Automatic Differentiation

1. Define a pair (q_0, q_1), with q_0, q_1 real numbers
2. Define operations on a pair like:

$$(q_0, q_1) + (r_0, r_1) = (q_0 + r_0, q_1 + r_1)\} \quad \text{obvious !}$$

$$c \cdot (q_0, q_1) = (c \cdot q_0, c \cdot q_1) \quad \text{obvious !}$$

$$(q_0, q_1) \cdot (r_0, r_1) = (q_0 \cdot r_0,\ q_0 \cdot r_1 + q_1 \cdot r_0) \quad \text{not obvious !}$$

Automatic Differentiation

1. Define a pair (q_0, q_1), with q_0, q_1 real numbers
2. Define operations on a pair like:

$$(q_0, q_1) + (r_0, r_1) = (q_0 + r_0, q_1 + r_1)$$

$$c \cdot (q_0, q_1) = (c \cdot q_0, c \cdot q_1)$$

$$(q_0, q_1) \cdot (r_0, r_1) = (q_0 \cdot r_0,\ q_0 \cdot r_1 + q_1 \cdot r_0)$$

3. And some ordering:

$$(q_0, q_1) < (r_0, r_1) \quad \text{if} \quad q_0 < r_0 \quad \text{or} \quad (q_0 = r_0 \quad \text{and} \quad q_1 < r_1)$$

$$(q_0, q_1) > (r_0, r_1) \quad \text{if} \quad q_0 > r_0 \quad \text{or} \quad (q_0 = r_0 \quad \text{and} \quad q_1 > r_1)$$

Automatic Differentiation

1. Define a pair (q_0, q_1), with q_0, q_1 real numbers
2. Define operations on a pair like:

$$(q_0, q_1) + (r_0, r_1) = (q_0 + r_0, q_1 + r_1)$$

$$c \cdot (q_0, q_1) = (c \cdot q_0, c \cdot q_1)$$

$$(q_0, q_1) \cdot (r_0, r_1) = (q_0 \cdot r_0,\ q_0 \cdot r_1 + q_1 \cdot r_0)$$

3. And some ordering:

$$(q_0, q_1) < (r_0, r_1) \quad \text{if} \quad q_0 < r_0 \quad \text{or} \quad (q_0 = r_0 \quad \text{and} \quad q_1 < r_1)$$

$$(q_0, q_1) > (r_0, r_1) \quad \text{if} \quad q_0 > r_0 \quad \text{or} \quad (q_0 = r_0 \quad \text{and} \quad q_1 > r_1)$$

4. This implies something strange:

$$(0, 0) < (0, 1) < (r, 0) \text{ (for any positive } r)$$

$$(0, 1) \cdot (0, 1) = (0, 0) \rightarrow (0, 1) = \sqrt{(0, 0)}!!$$

Automatic Differentiation

This means that (0,1) is between 0 and ANY real number → infinitely small !!!

We call this therefore "differential unit" $d = (0, 1) = \delta$.

Of course $(q, 0)$ is just the real number q and we define "real part" and "differential part" (a bit like complex numbers..):

$$q_0 = \mathcal{R}(q_0,\ q_1) \quad \text{and} \quad q_1 = \mathcal{D}(q_0,\ q_1)$$

With our rules we can further see that:

$$(1, 0) \cdot (q_0, q_1) = (q_0, q_1)$$

$$(q_0, q_1)^{-1} = \left(\frac{1}{q_0}, -\frac{q_1}{q_0^2}\right)$$

Automatic Differentiation

We let a function f act on the pair (or vector) using our rules.

For example:

$$f(x) \to f(x, 0)$$

acts like the function f on the real variable x:

$$f(x) = \mathcal{R}\,[f(x,\ 0)]$$

What about the differential part $\mathcal{D}$?

Automatic Differentiation

Key formula (without proof): for a function $f(x)$:

$$\mathcal{D}[f(x+d)] = \mathcal{D}[f((x,0)+(0,1))] = \mathcal{D}[f(x,1)] = f'(x)$$

An example instead:

$$f(x) = x^2 + \frac{1}{x}$$

then using school calculus:

$$f'(x) = 2x - \frac{1}{x^2}$$

For $x = 2$ we get then:

$$f(2) = \frac{9}{2}, \quad f'(2) = \frac{15}{4}$$

The miracle:

For x in:

$$f(x) = x^2 + \frac{1}{x}$$

we substitute: x $\rightarrow$ (x, 1) = (2, 1) and use our rules:

$$\begin{aligned} f[(2,1)] &= (2,1)^2 + (2,1)^{-1} \\ &= (4,4) + (\frac{1}{2}, -\frac{1}{4}) \\ &= (\frac{9}{2}, \frac{15}{4}) = (f(2), f'(2)) \quad !!! \end{aligned}$$

The computation of derivatives becomes an algebraic problem, no need for small numbers, exact !

Higher orders

1. The pair $(q_0, 1)$, becomes a vector of order N:

   $(q_0, 1) \rightarrow (q_0, 1, 0, 0, ...,0)$ $\quad \delta = (0, 1, 0, 0, 0, ...)$

2. $(q_0, q_1, q_2, ...q_N) + (r_0, r_1, r_2, ...r_N) = (s_0, s_1, s_2, ...s_N)$

   with: $s_i = q_i + r_i$

3. $c \cdot (q_0, q_1, q_2, ...q_N) = (c \cdot q_0, c \cdot q_1, c \cdot q_2, ...c \cdot q_N)$

4. $(q_0, q_1, q_2, ...q_N) \cdot (r_0, r_1, r_2, ...r_N) = (s_0, s_1, s_2, ...s_N)$

   with:

$$s_i = \sum_{k=0}^{i} \frac{i!}{k!(i-k)!} q_k r_{i-k}$$

More variables

If we had started with:

$$x = (a, 1, 0, 0, 0...)$$

we would get:

$$f(a+d) = (\ f(a),\ f'(a),\ f''(a),\ f'''(a), \ldots\ f^{(n)}(a)\ )$$

we can extend it to more variables x, y (here only two):

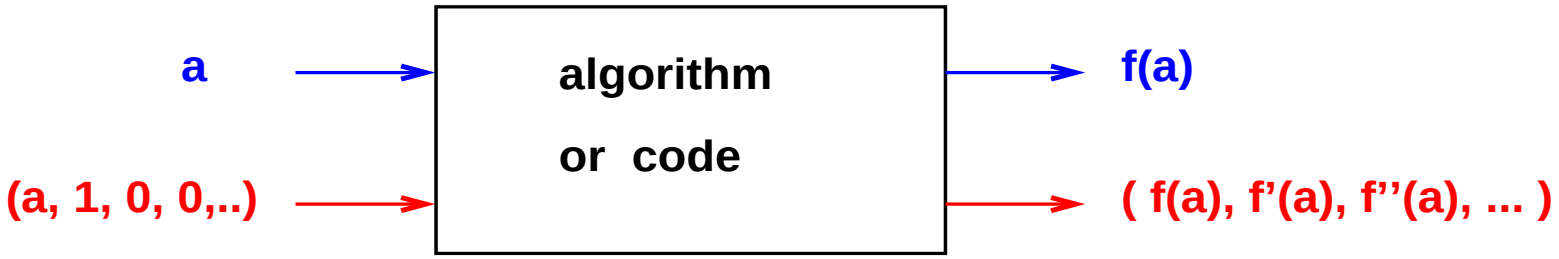
$$x = (a, 1, 0, 0, 0...) \qquad dx = (0, 1, 0, 0, 0, ..)$$

$$y = (b, 0, 1, 0, 0...) \qquad dy = (0, 0, 1, 0, 0, ..)$$

and get (with more complicated multiplication rules):

$$f(a+dx) \quad \rightarrow \quad f((a+dx), b+dy)) = \left(f, \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial^2 f}{\partial x^2}, \frac{\partial^2 f}{\partial x \partial y}, \ldots\right)(a, b)$$

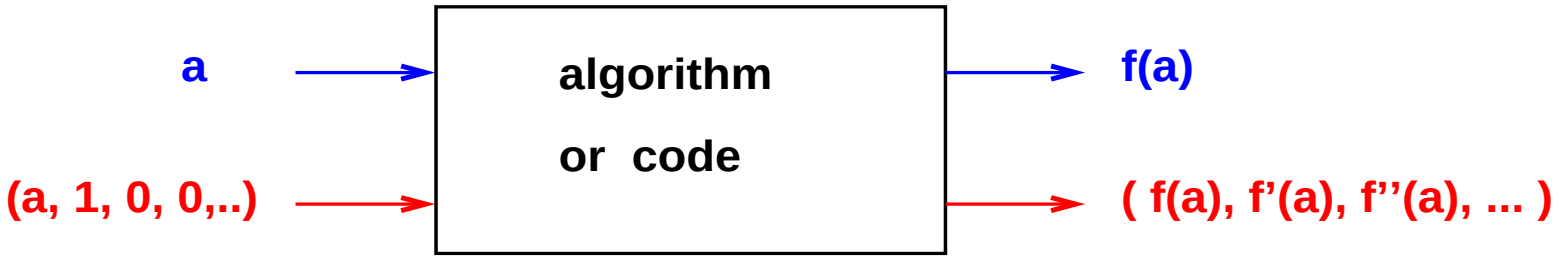
What is the use of that:



"Algorithm" can be a mathematical function or a complex computer code

1. We replace all standard operations with our new definitions
2. We push the vector $f(x) = (a, 1, 0, 0, 0...)$ through the algorithm as if it is a vector in phase space
3. We extract a truncated Taylor map with the desired accuracy and to any order

What is the use of that:



- Replacing the operations with a new definition is very easy using polymorphism of modern languages (e.g. C++, FORTRAN 95,..) and operator overloading, see following example
- We have a Taylor map, i.e. analytical representation of a computer code !!!

→ Normal form analysis on Taylor series is much easier: the coefficients are directly related to the various aberrations !!

What is the use of that:

- Demonstrate with simple examples (FORTRAN 95):
  - First show the concept
  - Simple FODO cell
  - Normal form analysis of the FODO cell with octupoles
- All examples and all source code in:

  Website: http://cern.ch/Werner.Herr/CAS2015/DA

  Small DA package provided by E. Forest

Look at this small example: $f(a) = \sin(\frac{\pi}{6})$

```
PROGRAM DATEST1
use my_own_da
real x,z, dx
my_order=3
dx=0.0
x=3.141592653/6.0 + dx
call track(x, z)
call print(z,6)
END PROGRAM DATEST1

SUBROUTINE TRACK(a, b)
use my_own_da
real a,b
b = sin(a)
END SUBROUTINE TRACK
```

```
PROGRAM DATEST2
use my_own_da
type(my_taylor) x,z, dx
my_order=3
dx=1.0.mono.1    ! this is our (0,1)
x=3.141592653/6.0 + dx
call track(x, z)
call print(z,6)
END PROGRAM DATEST2

SUBROUTINE TRACK(a, b)
use my_own_da
type(my_taylor) a,b
b = sin(a)
END SUBROUTINE TRACK
```

Look at the results:

(0,0) 0.5000000000000E+00

(0,0) 0.5000000000000E+00
(1,0) 0.8660254037844E+00
(0,1) 0.0000000000000E+00
(2,0) -0.2500000000000E+00
(0,2) 0.0000000000000E+00
(1,1) 0.0000000000000E+00
(3,0) -0.1443375672974E+00
(0,3) 0.0000000000000E+00
(2,1) 0.0000000000000E+00
(1,2) 0.0000000000000E+00

We have $sin(\frac{\pi}{6}) = 0.5$ all right, but what is the rest ??

Look at the results:

(0,0) 0.5000000000000E+00
(1,0) 0.8660254037844E+00
(0,1) 0.0000000000000E+00
(2,0) -0.2500000000000E+00
(0,2) 0.0000000000000E+00
(1,1) 0.0000000000000E+00
(3,0) -0.1443375672974E+00
(0,3) 0.0000000000000E+00
(2,1) 0.0000000000000E+00
(1,2) 0.0000000000000E+00

(0,0) 0.5000000000000E+00

$$sin(\frac{\pi}{6} + \Delta x) = sin(\frac{\pi}{6}) + cos(\frac{\pi}{6})\Delta x^1 - \frac{1}{2}sin(\frac{\pi}{6})\Delta x^2 - \frac{1}{6}cos(\frac{\pi}{6})\Delta x^3$$

What is the use of that:

- We have used a simple algorithm here (sin) but it can be anything very complex (try: $\frac{sin(x)}{x}$ using the DA package provided on the webpage)
- We can compute nonlinear maps as a Taylor expansion of anything the program computes
- Simply by:
  - Replacing regular (e.g. REAL) types by TPSA types (my_taylor) i.e. variables x, p are automatically replaced by $(x, 1, 0, ..)$ and $(p, 0, 1, 0, ..)$ etc.
  - Operators and functions $(+, -, *, =, ..., exp, sin, ...)$ automatically overloaded, i.e. behave according to new type

What is the use of that:

Assume the *Algorithm* describes one turn, then:

- Normal tracking:
  - $X_n = (x, p_x, y, p_y, s, \delta)_n \rightarrow X_{n+1} = (x, p_x, y, p_y, s, \delta)_{n+1}$
  - Coordinates after one completed turn
- TPSA tracking:
  - $X_n = (x, p_x, y, p_y, s, \delta)_n \rightarrow X_{n+1} = \sum C_j X_n^j$
  - Taylor coefficients after one completed turn
  - The C_j contain useful information about behaviour
  - Taylor map directly used for normal form analysis →

Demo 1:

- Track through a FODO lattice:

  QF - DRIFT - QD - DRIFT

  Integrate 100 steps in the quadrupoles

  Now we use three variables:

  x, p, Δp = (z(1), z(2), z(3))

Demo 1:

```
program fodo1
use my_own_da
use my_analysis
type(my_taylor) z(3)
type(normalform) NORMAL
type(my_map) M,id

real(dp) L,DL,k1,k3,fix(3)
integer i,nstep



my_order=4  ! maximum order 4
fix=0.0 ! fixed point
id=1
z=fix+id

LC=62.5     ! half cell length
L=3.0     ! quadrupole length
nstep=100
DL=L/nstep
k1=0.003     ! quadrupole strength
```

```
do i=1,nstep     ! track through quadrupole
z(1)=z(1)+DL/2*z(2)
z(2)=z(2)-k1*DL*z(1)/(1 + z(3))
z(1)=z(1)+DL/2*z(2)
enddo

z(1)=z(1)+LC*z(2)     !  drift of half cell length

do i=1,nstep     ! track through quadrupole
z(1)=z(1)+DL/2*z(2)
z(2)=z(2)-k1*DL*z(1)/(1 + z(3))
z(1)=z(1)+DL/2*z(2)
enddo

z(1)=z(1)+LC*z(2)     !  drift of half cell length

call print(z(1),6)
call print(z(2),6)
M=z     !  overloads coefficient with the map
NORMAL=M     !  overloads map with normal form analysis
write(6,*) normal%tune, normal%dtune_da
end program fodo1
```

The result is:

(0,0,0) 0.9442511679729E-01
(0,0,1) -0.9729519276183E-01

(1,0,0) 0.6972061935061E-01
(0,1,0) 0.1677727932585E+03
(1,0,1) 0.1266775134236E+01
(0,1,1)-0.3643444875882E+02
(1,0,2)-0.1603248617779E+01
(0,1,2) 0.3609522079691E+02
(1,0,3) 0.1939697138318E+01
(0,1,3)-0.3575511053483E+02

(1,0,0)-0.5300319873866E-02
(0,1,0) 0.1588490329398E+01
(1,0,1) 0.1060055415702E-01
(0,1,1)-0.5832024543075E+00
(1,0,2)-0.1590066005419E-01
(0,1,2) 0.5779004431627E+00
(1,0,3) 0.2120059477024E-01
(0,1,3)-0.5725843143370E+00

From the elements in the Taylor expansion, the result for the matrix per cell:

$$\Delta x_f = 0.06972\Delta x_i + 167.77\Delta p_i$$

$$\Delta p_f = -0.00530\Delta x_i + 1.5885\Delta p_i$$

The output from the normal form analysis are (per cell !):

Tune = (0,0,0) = 0.094425

Chromaticity = (0,0,1) = -0.097295

The miracle ...

- In FORTRAN95 derived ”type” plays the role of ”structures” in C

  After the assignment: NORMAL = M, the parameters are derived from the Taylor map (M) and give:

  NORMAL%tune is the tune

  NORMAL%dtune_da is the detuning with amplitude $\frac{dQ}{da}$

  NORMAL%R, NORMAL%A_T, NORMAL%A_T**-1 are the matrices

  $\mathcal{A}\mathcal{M}\mathcal{A}^{-1} = \mathcal{R}$ from the normal form transformation

  From $\mathcal{A}$ we get α, β, γ ...

  .. and some more

Modified previous example (with one octupole):

```
program fodo3
use my_own_da
use my_analysis
type(my_taylor) z(3)
type(normalform) NORMAL
type(my_map) M,id

real(dp) L,DL,k1,k3,fix(3)
integer i,nstep



my_order=4  ! maximum order 4
fix=0.0 ! fixed point
id=1
z=fix+id

LC=62.5     ! half cell length
L=3.0     ! quadrupole length
nstep=100
DL=L/nstep
k1=0.003     ! quadrupole strength
k3=0.01     ! octupole strength

do i=1,nstep     ! track through quadrupole
z(1)=z(1)+DL/2*z(2)
z(2)=z(2)-k1*DL*z(1)/(1 + z(3))
z(1)=z(1)+DL/2*z(2)
enddo
z(2)=z(2)-k3*z(1)**3/(1 + z(3))  ! now we add an octupole kick !!!
z(1)=z(1)+LC*z(2)     ! drift of half cell length

do i=1,nstep     ! track through quadrupole
z(1)=z(1)+DL/2*z(2)
z(2)=z(2)-k1*DL*z(1)/(1 + z(3))
z(1)=z(1)+DL/2*z(2)
enddo

z(1)=z(1)+LC*z(2)     ! drift of half cell length

call print(z(1),6)
call print(z(2),6)
M=z
NORMAL=M
write(6,*) normal%tune, normal%dtune_da
end program fodo3
```

The result is:

(0,0,0) 0.9442511679729E-01
(0,0,1) -0.9729519276183E-01

(2,0,0) 0.5374370086899E+02
(0,2,0) 0.5374370086899E+02
(0,0,2) 0.1018391758451E+00
(2,0,1) 0.2035776281196E+02
.....

(1,0,0) 0.6972061935061E-01
(0,1,0) 0.1677727932585E+03
(1,0,1) 0.1266775134236E+01
(0,1,1)-0.3643444875882E+02
(3,0,0)-0.1586519461687E+01
(2,1,0)-0.1440953324752E+02
(1,2,0)-0.4362477179879E+02

.....
(1,0,0)-0.5300319873866E-02
(0,1,0) 0.1588490329398E+01
(1,0,1) 0.1060055415702E-01
(0,1,1)-0.5832024543075E+00
(3,0,0)-0.1519218878892E-01

Linear matrix as before, but effects of octupole.

The output from the normal form analysis are (per cell !):

Tune = (0,0,0) = 0.094425

Chromaticity = (0,0,1) = -0.097295

Detuning with amplitude = (2,0,0) = 53.74 !

Modified previous example (with octupole):

Remember the normal form transformation:

$$\mathcal{A}\mathcal{M}\mathcal{A}^{-1} = \mathcal{R}$$

The type normalform in the demonstration package also contains the maps $\mathcal{A}$ and $\mathcal{R}$!

j2=(x**2+p**2)*NORMAL%A**(-1)

(remember: x**2+p**2 is the tilted ellipse ....

Can get the optical functions out because

- β: coefficient of p**2 of invariant j2
- α: coefficient of x*p of invariant j2
- γ: coefficient of x**2 of invariant j2

Modified previous example (with octupole):

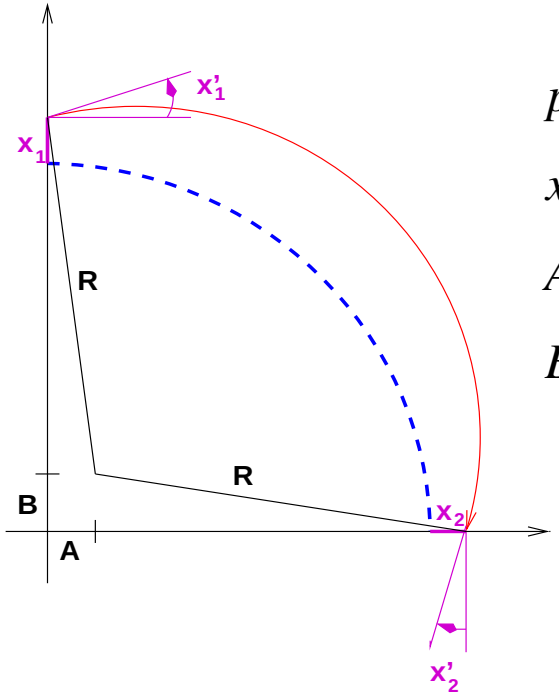
We obtain (here at the end of the cell):

beta, alpha, gamma

300.080714 -1.358246 9.480224E-003

This was trivial - now a (normally) hard one

The exact map:



$$p_2 = \quad \sin(x_2') = -\frac{B}{R}$$

$$x_2 = \quad A - R(1 - \cos(x_2')) = A - R(1 - \sqrt{1 - p_2^2})$$

$$A = \quad R \cdot p_1 = R \cdot \sin(x_1')$$

$$B = \quad R(1 - \cos(x_1')) + x_1 = R(1 - \sqrt{1 - p_1^2}) + x_1$$

A 90^0 bending magnet ..

How to apply Differential Algebra here ...

- Start with initial coordinates in DA style:

$$x_1 = (0, 1, 0, ...)$$

$$p_1 = (0, 0, 1, ...) \quad \text{and have:}$$

$$A = (0, 0, R, 0, ...)$$

$$B = (0, 1, 0, 0, 0, R, 0, ...)$$

- After pushing them through the algorithm:

$$x_2 = (0, 0, R, -\tfrac{1}{R}, 0, 0, 0...) = (0, \tfrac{\partial x_2}{\partial x_1}, \tfrac{\partial x_2}{\partial p_1}, \tfrac{\partial^2 x_2}{\partial x_1^2}, \tfrac{\partial^2 x_2}{\partial x_1 \partial p_1}, ...)$$

$$p_2 = (0, -\tfrac{1}{R}, 0, 0, 0, -1, 0...) = (0, \tfrac{\partial p_2}{\partial x_1}, \tfrac{\partial p_2}{\partial p_1}, \tfrac{\partial^2 p_2}{\partial x_1^2}, \tfrac{\partial^2 p_2}{\partial x_1 \partial p_1}, ...)$$

- Automatically evaluates all non-linearities to any desired order ..

How to apply Differential Algebra here ...

- Start with initial coordinates in DA style:

$$x_1 = (0, 1, 0, ...)$$

$$p_1 = (0, 0, 1, ...) \quad \text{and have:}$$

$$A = (0, 0, R, 0, ...)$$

$$B = (0, 1, 0, 0, 0, R, 0, ...)$$

- After pushing them through the algorithm:

$$x_2 = (0, 0, R, -\tfrac{1}{R}, 0, 0, 0...) = (0, \tfrac{\partial x_2}{\partial x_1}, \tfrac{\partial x_2}{\partial p_1}, \tfrac{\partial^2 x_2}{\partial x_1^2}, \tfrac{\partial^2 x_2}{\partial x_1 \partial p_1}, ...)$$

$$p_2 = (0, -\tfrac{1}{R}, 0, 0, 0, -1, 0...) = (0, \tfrac{\partial p_2}{\partial x_1}, \tfrac{\partial p_2}{\partial p_1}, \tfrac{\partial^2 p_2}{\partial x_1^2}, \tfrac{\partial^2 p_2}{\partial x_1 \partial p_1}, ...)$$

- Automatically evaluates all non-linearities to any desired order ..

Some we know ...

Transfer matrix of a dipole:

$$M_{dipole} = \begin{pmatrix} cos(\frac{L}{R}) & Rsin(\frac{L}{R}) \\ -\frac{1}{R}sin(\frac{L}{R}) & cos(\frac{L}{R}) \end{pmatrix} = \begin{pmatrix} \frac{\partial x_2}{\partial x_1} & \frac{\partial x_2}{\partial p_1} \\ \frac{\partial p_2}{\partial x_1} & \frac{\partial p_2}{\partial p_1} \end{pmatrix}$$

For a 90^0 bending angle we get:

$$M_{dipole} = \begin{pmatrix} 0 & R \\ -\frac{1}{R} & 0 \end{pmatrix}$$

as computed, but we also have all derivatives and non-linear effects !

What is the use of that:

- Although not strictly a symbolic method in the traditional sense:
  - TPSA provide analytic expression (Taylor series) for the one turn map
  - Parameters (i.e. strength) can be used as input and make it parameter dependent
  - Can be used for tracking
  - Can be analysed for dynamic behaviour of the system
  - Typical use: Normal Form Analysis discussed earlier, rather straightforward from a Taylor expansion

A Summary

- Contemporary methods are very powerful and can:
  - Largely avoid: assumptions, approximations and simplifications
  - They form a natural extension from linear to non-linear beam dynamics
- For realistic machines: symplectic, iterative mapping is appropriate, using
  - Symplectic integration
  - Lie transformations and normal form analysis
  - Truncated Power Series Algebra (TPSA)

A Summary

- Contemporary methods are very powerful and can:
  - Largely avoid: assumptions, approximations and simplifications
  - They form a natural extension from linear to non-linear beam dynamics
- For realistic machines: symplectic, iterative mapping is appropriate, using
  - Symplectic integration
  - Lie transformations and normal form analysis
  - Truncated Power Series Algebra (TPSA)

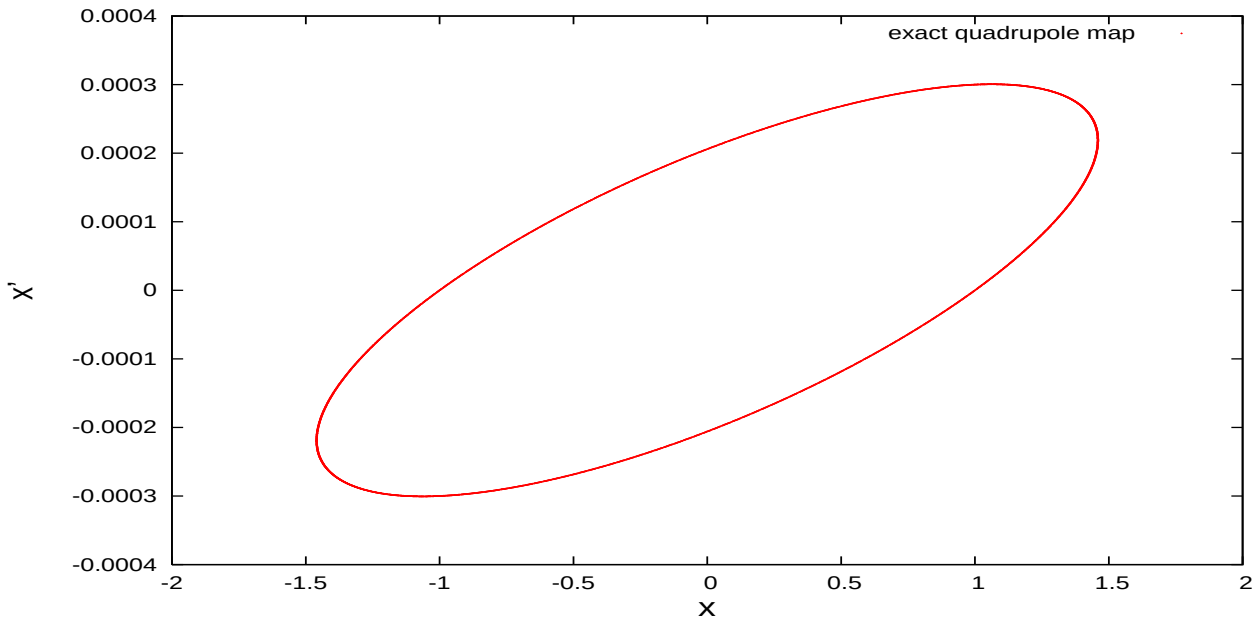
If you don't like all that, get used to it ...

and thanks for your stamina

Back up

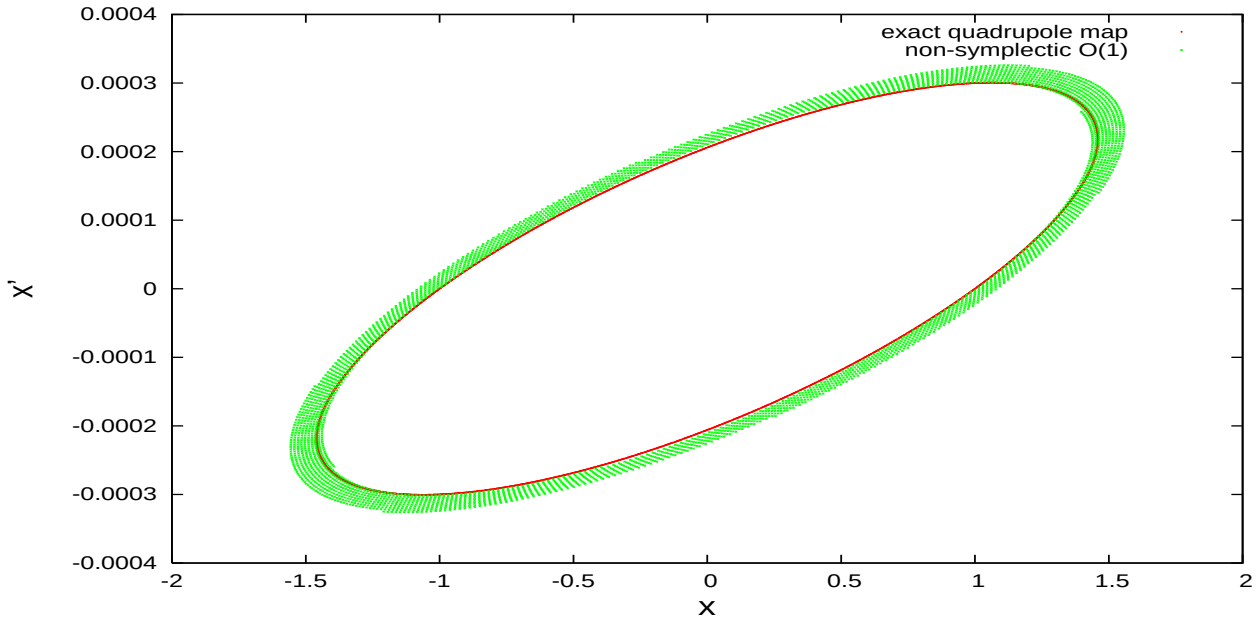
- backup slides -

What is the point ???



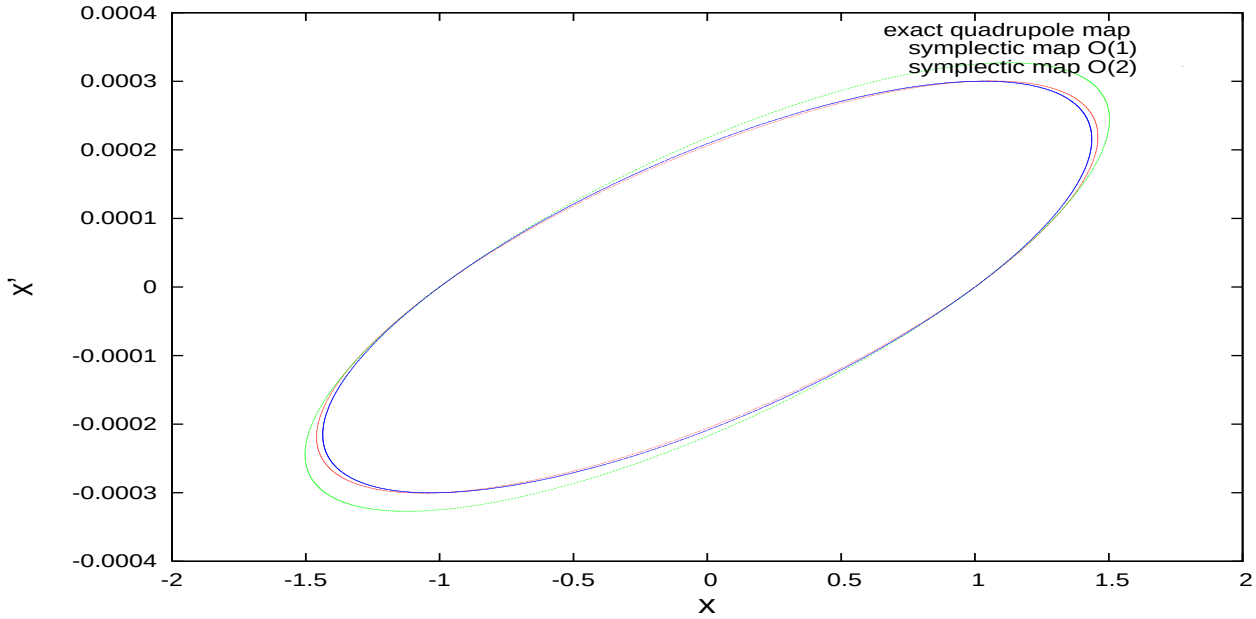
- Phase ellipse - quadrupole exact solution

Quadrupole non-symplectic solution



- Non-symplecticity: particles spiral towards outside

Quadrupole symplectic $O(L^1)$ and $O(L^2)$ solutions



- symplecticity: phase space ellipse accurate enough

Example: sextupole (1D)

Given the Hamiltonian h:

$$h = -\mu J - \frac{3}{8} k (2\beta J)^{3/2} \cdot \left(\frac{sin(3\Psi + \frac{3\mu}{2})}{sin\frac{3\mu}{2}} - \frac{sin(\Psi + \frac{\mu}{2})}{sin\frac{\mu}{2}} \right)$$

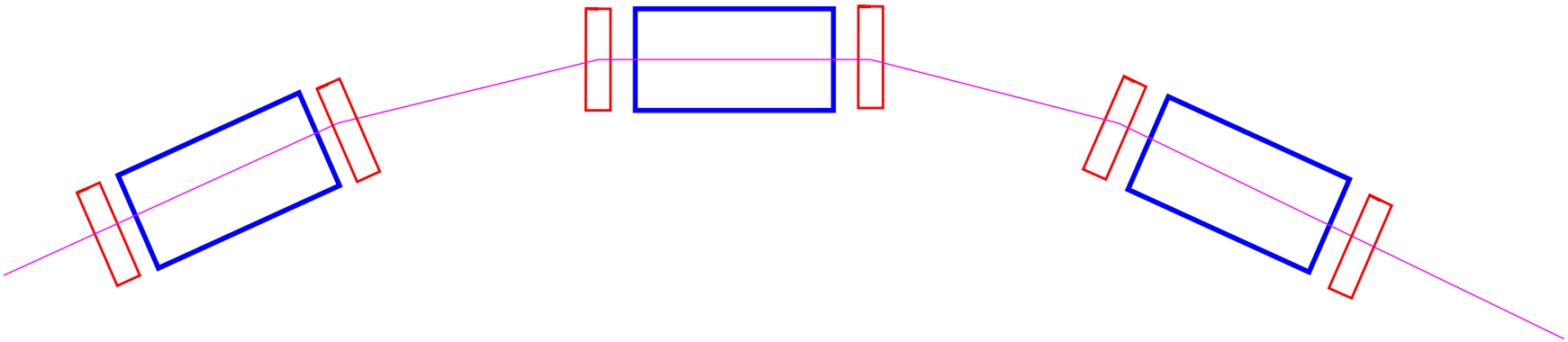
particles move in phase space along constant h.

Back to Cartesian coordinates we get for h:

$$h = -\frac{\mu}{2}(x^2 + x'^2)\frac{3}{8}\mu\beta^{3/2} x[(3x'^2 - x^2)cot\frac{3\mu}{2} - (x^2 + x'^2)cot\frac{\mu}{2} - 4xx']$$

Constant h defines the trajectory in phase space !

Where to put the elements in an accelerator ?



$$\frac{d^2x}{ds^2} + K(s)\,x = 0$$

- Usually use s (pathlength) along ”reference path”
- ”Reference path” defined geometrically by straight sections and bending magnets

Second order MAPS concatenation

Assume now 2 maps of second order:

$\mathcal{A}_2 = [R^A, T^A]$ and $\mathcal{B}_2 = [R^B, T^B]$

the combined second order map

$C_2 = \mathcal{A}_2 \circ \mathcal{B}_2$ is $C_2 = [R^C, T^C]$ with:

$$R^C = R^A \cdot R^B$$

and (after truncation of higher order terms !!):

$$T^C_{ijk} = \sum_{l=1}^{4} R^B_{il} T^A_{ljk} + \sum_{l=1}^{4} \sum_{m=1}^{4} T^B_{ilm} R^A_{lj} R^A_{mk}$$

Symplecticity for higher order MAPS

try truncated Taylor map in 2D, second order:

$$\begin{pmatrix} x \\ x' \end{pmatrix} = \begin{pmatrix} R_{11}x_0 + R_{12}x'_0 + T_{111}x_0^2 + T_{112}x_0x'_0 + T_{122}x'^2_0 \\ R_{21}x_0 + R_{22}x'_0 + T_{211}x_0^2 + T_{212}x_0x'_0 + T_{222}x'^2_0 \end{pmatrix}$$

The Jacobian becomes:

$$\mathcal{J} = \begin{bmatrix} R_{11} + 2T_{111}x_0 + T_{112}x'_0 & R_{12} + T_{112}x_0 + 2T_{122}x'_0 \\ R_{21} + 2T_{211}x_0 + T_{212}x'_0 & R_{22} + T_{212}x_0 + 2T_{222}x'_0 \end{bmatrix}$$

symplecticity condition requires that:

det $\mathcal{J}$ = 1 for all x_0 and all x'_0

Symplecticity for higher order MAPS

This is only possible for the conditions:

$$\left(\begin{array}{c} R_{11}R_{22} - R_{12}R_{21} = 1 \\ R_{11}T_{212} + 2R_{22}T_{111} - 2R_{12}T_{211} - R_{21}T_{112} = 0 \\ 2R_{11}T_{222} + R_{22}T_{112} - R_{12}T_{212} - 2R_{21}T_{122} = 0 \end{array}\right)$$

- 10 coefficients, but 3 conditions
- number of independent coefficients only 7 !
- Taylor map requires more coefficients than necessary
- e.g. 4D, order 4: coefficients 276 instead of 121

Canonical transformations

- With Hamiltonian's equations, still have to solve (2n) differential equations
- Not necessarily easy, but:
  - More freedom to choose the variables q and p (because they have now "equal" status)
  - Try to find variables where they are easy to solve
- Change of variables through "canonical transformations"

Why canonical transformations ?

- Hamiltonian have one advantage over Lagrangians:
  - If the system has a symmetry, i.e. a coordinate q_i does not occur in H (i.e. $\frac{\partial H}{\partial q_i} = 0 \quad \rightarrow \quad \frac{dp_i}{dt} = 0$) → the corresponding momentum p_i is conserved (and the coordinate q_i can be ignored in the other equations of the set).
  - Comes also from Lagrangian, but the velocities still occur in $\mathcal{L}$!

Canonical transformations

Starting with $H(q, p, t)$ get new coordinates:

$$Q_i = Q_i(q, p, t)$$

$$P_i = P_i(q, p, t)$$

and new Hamiltonian $K(Q, P, t)$ with:

$$\frac{\partial K}{\partial Q_j} = -\dot{P}_j = -\frac{dP_j}{dt}, \qquad \frac{\partial K}{\partial P_j} = \dot{Q}_j = \frac{dQ_j}{dt}$$

- We can two types of canonical transformations

Canonical transformations - type 1

- Ideally one would like a Hamiltonian H and coordinates with:

$$\frac{\partial H}{\partial q_j} = -\dot{p}_j = -\frac{dp_j}{dt} = 0$$

- Coordinate q_j not explicit in H

- p_j is a constant of the motion (!) and:

$$\frac{dq_j}{dt} = \frac{\partial H(p_1, p_2, ..p_n)}{\partial p_j} = F_j(p_1, p_2, ..p_n)$$

which can be directly integrated to get $q_j(t)$

Canonical transformations - type 1, example

Harmonic oscillator:

$$H = T + V = \frac{1}{2}mv^2 + \frac{m\omega^2}{2}x^2 = \frac{p^2}{2m} + \frac{m\omega^2}{2}x^2$$

try: $x = \sqrt{\frac{2P}{m\omega}} \cdot sin(X)$ and $p = \sqrt{2m\omega P} \cdot cos(X)$ and we get:

$$K = \omega P cos^2(X) + \omega P sin^2(X) = \omega P$$

then:

$$\frac{dX}{dt} = \frac{\partial K}{\partial P} = \omega \rightarrow X = \omega t + \alpha$$

back transformation to x,p:

$$x = \sqrt{\frac{2E}{m\omega^2}} sin(\omega t + \alpha)$$

Canonical transformations - type 2

- Find a transformation of q, p at time t to values q_0, p_0 at time $t = 0$.

$$q = q(q_0, p_0, t)$$

$$p = p(q_0, p_0, t)$$

- The transformations ARE the solution of the problem !

For both types: how to find the transformation ?

- Without details: Hamilton-Jacobi equation ...

Extension: general monomials

Monomials in x and p of orders n and m ($x^n p^m$)

$$e^{:ax^n p^m:}$$

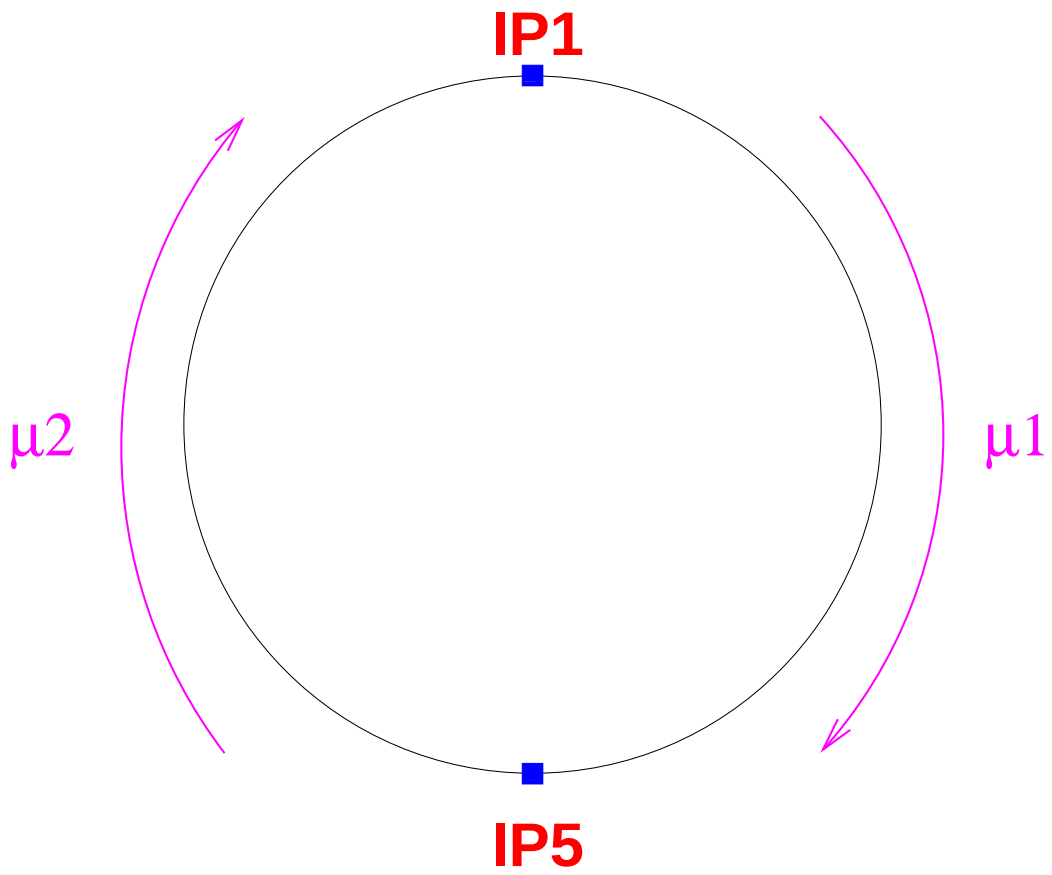
gives for the map (for $n \neq m$):

$$\begin{aligned} e^{:ax^n p^m:} x &= x \cdot [1 + a(n-m)x^{n-1} p^{m-1}]^{m/(m-n)} \\ e^{:ax^n p^m:} p &= p \cdot [1 + a(n-m)x^{n-1} p^{m-1}]^{n/(n-m)} \end{aligned}$$

gives for the map (for $n = m$):

$$\begin{aligned} e^{:ax^n p^n:} x &= x \cdot e^{-anx^{n-1} p^{n-1}} \\ e^{:ax^n p^n:} p &= p \cdot e^{anx^{n-1} p^{n-1}} \end{aligned}$$

Collision scheme - two IPs



$$\mu = \mu 1 + \mu 2$$

Two IPs

→ two transfers f_2^1, f_2^2 and two beam-beam kicks F^1, F^2, first IP at μ_1, second IP at μ:

$$= \quad e^{:f_2^1:}\, e^{:F^1:}\, e^{:f_2^2:}\, e^{:F^2:} \;=\; e^{:h_2:}$$

$$= \quad e^{:f_2^1:}\, e^{:F^1:}\, e^{-:f_2^1:}\, e^{:f_2^1:}\, e^{:f_2^2:}\, e^{:F^2:} \;=\; e^{:h_2:}$$

$$= \quad e^{:f_2^1:}\, e^{:F^1:}\, e^{-:f_2^1:}\, e^{:f_2:}\, e^{:F^2:}\, e^{-:f_2:}\, e^{:f_2:} \;=\; e^{:h_2:}$$

$$= \quad e^{:e^{-:f_2^1:}F^1:}\, e^{:e^{-:f_2:}F^2:}\, e^{:f_2:} \;=\; e^{:h_2:}$$

$$f_2 \;=\; -\mu A, \quad f_2^1 \;=\; -\mu_1 A, \quad \textit{and} \quad f_2^2 \;=\; -\mu_2 A$$

Two IPs

here a miracle occurs (remember $g(:f_2:)e^{in\Psi} = g(in\mu)e^{in\Psi}$):

$$e^{:f_2^1:}e^{in\Psi} = e^{in\mu_1}e^{in\Psi} = e^{in(\mu_1+\Psi)}$$

i.e. the Lie transforms of the perturbations are phase shifted[2]).
Therefore:

$$e^{:e^{-:f_2^1:}F^1:}\ e^{:e^{-:f_2:}F^2:}\ e^{:f_2:} = e^{:h_2:}$$

becomes simpler with substitutions of $\Psi_1 = \Psi + \mu_1$ and $\Psi = \Psi + \mu$ in F^1 and F:

$$e^{:F^1(\Psi_1):}e^{:F(\Psi):}e^{:f_2:} \Rightarrow e^{:F^1(\Psi_1)+F(\Psi):}e^{:f_2:}$$

2) E. Forest, "Beam Dynamics, A New Attitude and Framework", 1998

Two IPs

gives for h_2:

$$h_2 = -\mu A + \sum_{n=-\infty}^{\infty} \frac{n\mu c_n(A)}{2\sin(n\frac{\mu}{2})} e^{-in(\Psi+\mu/2+\mu_1)} + e^{-in(\Psi+\mu/2)}$$

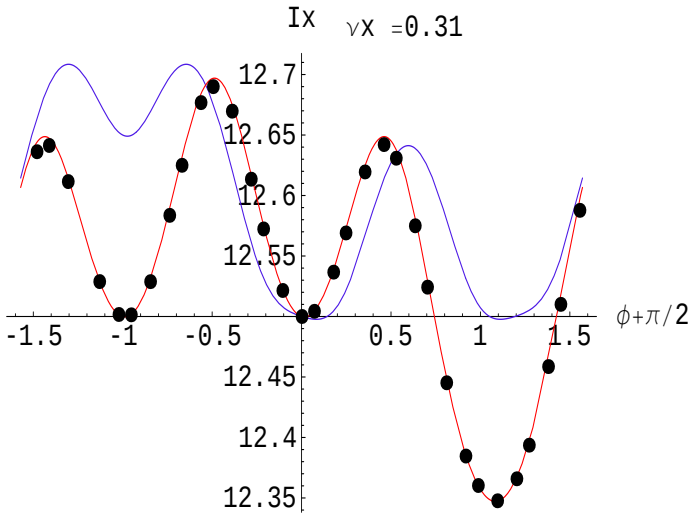
$$h_2 = -\mu A + 2c_0(A) + \underbrace{\sum_{n=1}^{\infty} \frac{2n\mu c_n(A)}{2\sin(n\frac{\mu}{2})} \cos(n(\Psi + \frac{\mu}{2} + \frac{\mu_1}{2}))\cos(n\frac{\mu_1}{2})}_{\text{interesting part}}$$

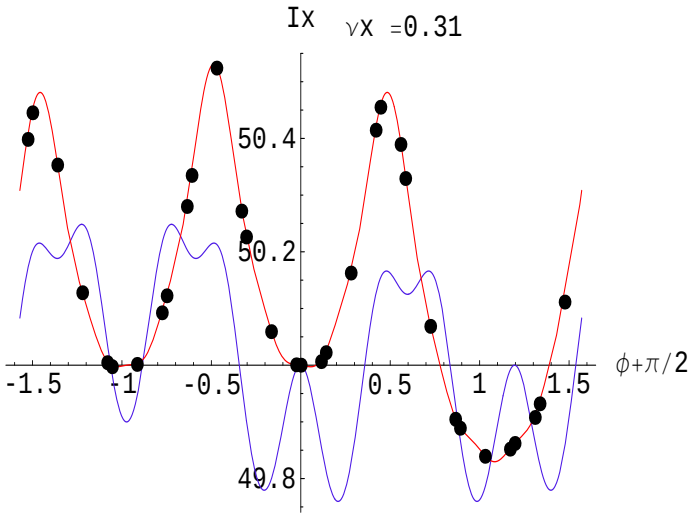
Nota bene, because of:

$$e^{:F(\Psi):} e^{:f_2:} \quad \rightarrow \quad e^{:F^1(\Psi_1)+F(\Psi):} e^{:f_2:}$$

can be generalized to more interaction points ...

Invariant versus tracking: two IPs





→ Shown for $5\sigma_x$ and $10\sigma_x$

Recap: Hamiltonian for a finite length element

We have from the Hamiltonian equations for the motion through an element with the Hamiltonian H for the element of length L:

$$\frac{dq}{dt} = [q, H] =: -H : q \quad \text{(from lecture 5)}$$

$$\rightarrow \frac{d^k q}{dt^k} = (: -H :)^k q$$

$$\rightarrow q(t) = \sum_{k=0}^{\infty} \frac{t^k}{k!} \left(\frac{d^k q}{dt^k} \right) = \sum_{k=0}^{\infty} \frac{t^k}{k!} (- : H :)^k = e^{:-tH:}$$

with independent variable s instead of t (nota bene: $s_0 = 0, t_0 = 0$):

$$\rightarrow q(s) = e^{:-LH:}$$

Lie transformations on moments:

We have used Lie transformations mainly to propagate coordinates and momenta, i.e. like:

$$e^{:f:} x_0 = x_1$$

$$e^{:f:} p_0 = p_1$$

of course for more dimensions $(x, p_x, y, p_y, ...)$:

- Remember: can be applied to any function of x and p !!
- In particular to moments like $x^2, xp, p^2, ...$

Lie transformations on moments

Assume a matrix M of the type:

$$M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}$$

described by a generator f, we have for the Lie transformation on the moment:

$$e^{:f:} x^2 = (e^{:f:} x)^2 \qquad \text{(see lecture 5)}$$

therefore:

$$(e^{:f:} x)^2 = (m_{11} x + m_{12} p)^2$$

$$(e^{:f:} x)^2 = m_{11}^2 x^2 \; + \; 2\, m_{11} m_{12} xp \; + \; m_{12}^2 p^2$$

From the map to the Hamiltonian

The other question → assuming we do not have the Hamiltonian, but a matrix M (from somewhere):

$$\mathcal{M} \equiv \begin{pmatrix} \cos(\mu) + \alpha sin(\mu) & \beta \sin(\mu) \\ -\gamma \sin(\mu) & \cos(\mu) - \alpha sin(\mu) \end{pmatrix}$$

i.e.:

$$\mathcal{M} Z_0 = Z_1$$

how do we find the corresponding form for f ?

$$\mathcal{M} \quad \leftrightarrow \quad e^{:f:}$$

rather clumsy, skim over the next 3 slides and show the result, (for derivation see e.g. [AC1])

For the interested: the details

For the linear matrix f must be a quadratic form in (x, p, ...).

Any quadratic form can be written as:

$$f = -\frac{1}{2} Z^* F Z \quad [\ = \ -\frac{1}{2}(a \cdot x^2 + b \cdot xp + c \cdot p^2)\ ]$$

where F is a symmetric, positive definite (why ?) matrix.
Then we can write (without proof, see e.g. Dragt):

$$: f : Z = S F Z$$

where S is the "symplecticity" matrix.
Therefore we get for the Lie transformation:

$$e^{:f:} Z \quad \leftrightarrow \quad e^{SF} Z$$

For the interested: the details

Since we have $n = 2$, we get (using $Hamilton - Cayley$ theorem):

$$e^{SF} = \exp\begin{pmatrix} b & c \\ -a & -b \end{pmatrix} = a_0 + a_1 \begin{pmatrix} b & c \\ -a & -b \end{pmatrix}$$

We now have to find a_0 and a_1 !

The eigenvalues of SF are:

$$\lambda_{\pm} = \pm i\sqrt{ac - b^2}$$

For the interested: the details

This tells us for the coefficients the conditions:

$$e^{\lambda_+} = a_0 + a_1 \cdot \lambda_+$$

$$e^{\lambda_-} = a_0 + a_1 \cdot \lambda_-$$

and therefore:

$$a_0 = \cos(\sqrt{ac - b^2})$$

$$a_1 = \frac{\sin(\sqrt{ac - b^2})}{\sqrt{ac - b^2}}$$

and

$$e^{SF} = \cos(\sqrt{ac - b^2}) + \frac{\sin(\sqrt{ac - b^2})}{\sqrt{ac - b^2}} \begin{pmatrix} b & c \\ -a & -b \end{pmatrix}$$

Example: a (simple) map to the Hamiltonian

For a general 2×2 matrix:

$$M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}$$

For the linear matrix f must be a quadratic form in (x, p, ...).

The result of the calculation is:

$$\frac{a}{-m_{21}} = \frac{2b}{m_{11} - m_{22}} = \frac{c}{m_{12}} = \frac{\sqrt{ac - b^2}}{\sin(\sqrt{ac - b^2})}$$

for the quadratic form: $f = -\frac{1}{2}(a \cdot x^2 + b \cdot xp + c \cdot p^2)$

From the map to the Hamiltonian

For the example of a (simple) drift:

$$\mathcal{M} \equiv \begin{pmatrix} 1 & L \\ 0 & 1 \end{pmatrix}$$

we find:

$$a = 0, \quad b = 0, \quad c = L$$

and for the generator:

$$f = -\frac{1}{2}(Lp^2)$$

From the map to the Hamiltonian

For the example of a thin quadrupole:

$$\mathcal{M} \equiv \begin{pmatrix} 1 & 0 \\ -\frac{1}{F} & 1 \end{pmatrix}$$

we find:

$$a = \frac{1}{F}, \quad b = 0, \quad c = 0$$

and for the generator:

$$f = -\frac{1}{2F}(x^2) = \frac{1}{2}kx^2$$

More on moments

To summarize the moments:

$$\begin{pmatrix} x^2 \\ xp \\ p^2 \end{pmatrix}_{s_2} = \begin{pmatrix} m_{11}^2 & 2m_{11}m_{12} & m_{12}^2 \\ m_{11}m_{21} & m_{11}m_{22} + m_{12}m_{21} & m_{12}m_{22} \\ m_{21}^2 & 2m_{21}m_{22} & m_{22}^2 \end{pmatrix} \circ \begin{pmatrix} x^2 \\ xp \\ p^2 \end{pmatrix}_{s_1}$$

This is the well known transfer matrix for optical parameters

A real life example: beam-beam interaction*)

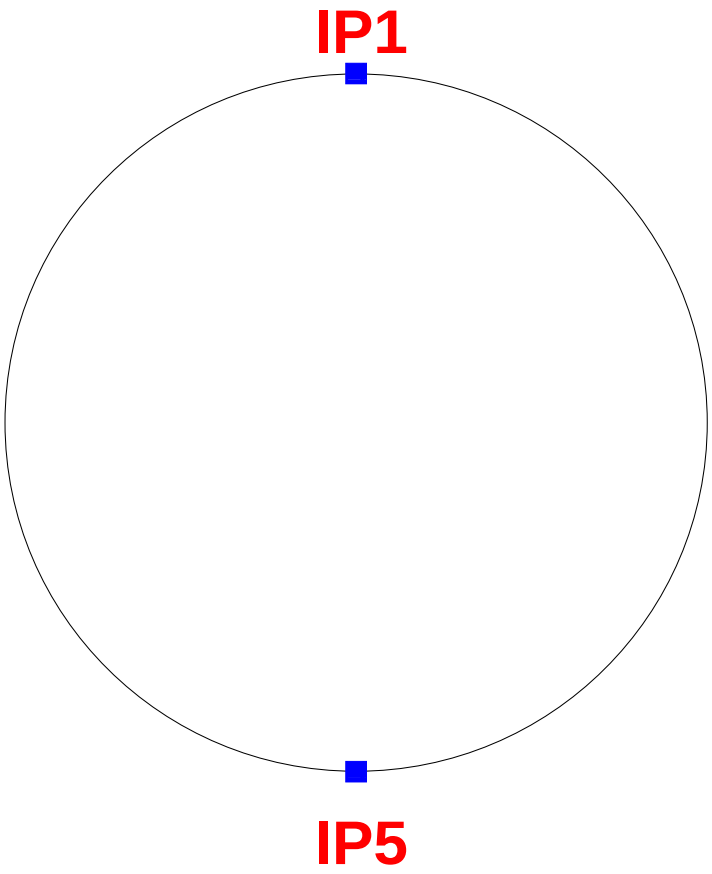
- Beam-beam interaction very non-linear
- Important to understand stability
- Non-linear effects such as amplitude detuning very important

Our questions ?

- How does the particles behave in phase space ?
- Do we have an invariant ?
- Can we calculate the invariant ?

*) From: W. Herr, D. Kaltchev, LHC Project Report 1082, (2008).

Collision scheme - two IPs



Start with single IP

”Classic” (B.C.) approach:

- Interaction point at beginning (end) of the ring (very local interactions, δ-functions)
- s-dependent Hamiltonian and perturbation theory:

$$\mathcal{H} = ... + \delta(s)\epsilon V$$

- Disadvantages:
  - for several IPs endless mathematics
  - conceptually and computationally easier method

Effect on invariants - start with single IP

Look for invariants h, (see e.g. Dragt[1)]), and evaluate for different number of interactions and phase advance.

Very well suited for local distortions (e.g. beam-beam kick)

Linear transfer $e^{:f_2:}$ and beam-beam interaction $e^{:F:}$, i.e.:

$$e^{:f_2:} \cdot e^{:F:} = e^{:h:}$$

with

$$f_2 = -\frac{\mu}{2}\left(\frac{x^2}{\beta} + \beta p_x^2\right)$$

and

$$F = \int_0^x dx' f(x')$$

1) A. Dragt, AIP Conference proceedings, Number 57 (1979)

Effect on invariants

using for a Gaussian beam f(x):

$$f(x) = \frac{2}{x}(1 - e^{\frac{-x^2}{2\sigma^2}})$$

as usual go to action angle variables Ψ, A:

$$x = \sqrt{2A\beta}\sin\Psi, \qquad p = \sqrt{\frac{2A}{\beta}}\cos\Psi$$

and write F(x) as Fourier series:

$$B(x) = \sum_{n=-\infty}^{\infty} c_n(A)e^{in\Psi} \quad \text{with :} \quad c_n(A) = \frac{1}{2\pi}\int_0^{2\pi} e^{-in\Psi}F(x)d\Psi$$

We need:

REMEMBER: with this transform:

$$f_2 = -\mu A$$

and useful properties of Lie operators (any textbook[2)]):

$$: f_2 : g(A) = 0, \qquad : f_2 : e^{in\Psi} = in\mu e^{in\Psi}, \qquad g(: f_2 :)e^{in\Psi} = g(in\mu)e^{in\Psi}$$

and the formula (because the beam-beam perturbation is small !):

$$e^{:f_2:}\, e^{:F:} \;=\; e^{:h:} \;=\; \exp\left[: f_2 + \left(\frac{: f_2 :}{1 - e^{-:f_2:}}\right) F + O(F^2) :\right]$$

2) E. Forest, "Beam Dynamics, A New Attitude and Framework", 1998

Single IP

gives immediately for h:

$$h = -\mu A + \sum_n c_n(A) \frac{in\mu}{1 - e^{-in\mu}} e^{in\Psi}$$

$$h = -\mu A + \sum_n c_n(A) \frac{n\mu}{2\sin(\frac{n\mu}{2})} e^{(in\Psi + i\frac{n\mu}{2})}$$

away from resonance, a normal form transformation takes away the pure oscillatory part and we have only:

$$h \;=\; -\mu A + c_0(A) \;= const.$$

$$\left[homework: \qquad \frac{dc_0(A)}{dA} \right]$$

Single IP

If you are too lazy or too busy:

$$\Delta Q = \frac{-1}{2\pi}\frac{dc_0(A)}{dA}$$

is the detuning with amplitude, i.e. the amplitude dependent frequency change of the transformation we had before ...

We get:

$$\Delta Q = \frac{-1}{2\pi}\frac{Nr_0}{\gamma A}[1 - e^{-A\beta/2\sigma^2} I_0(A\beta/2\sigma^2)]$$

Single IP - analysis of h

$$h = -\mu A + \sum_n c_n(A) \frac{n\mu}{2\sin(\frac{n\mu}{2})} e^{(in\Psi + i\frac{n\mu}{2})}$$

On resonance:

$$Q = \frac{p}{n} = \frac{\mu}{2\pi}$$

with $c_n \neq 0$:

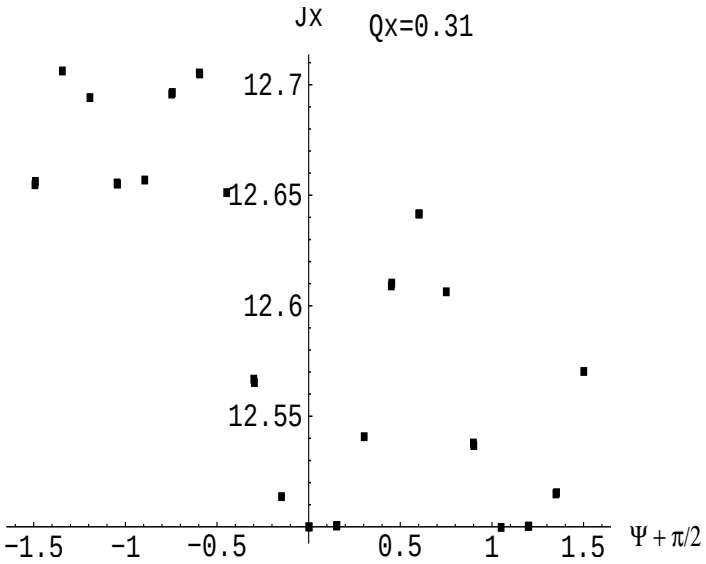
$$\sin(\frac{n\pi p}{n}) = \sin(p\pi) \equiv 0 \quad \forall \ integer \quad p$$

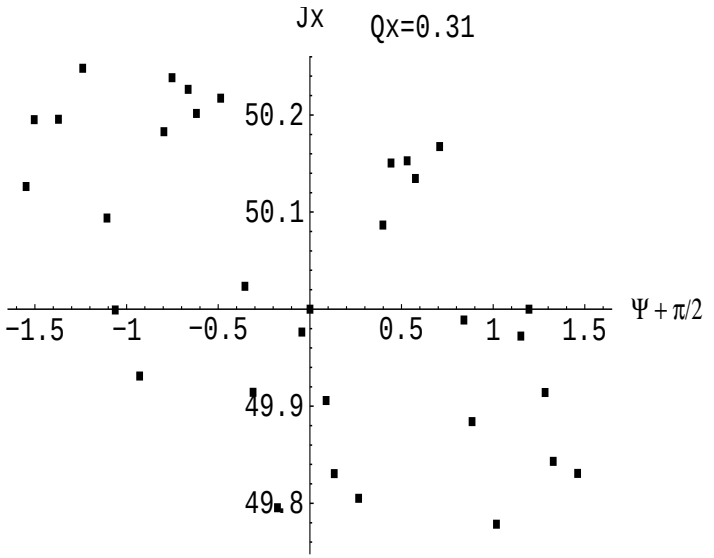
and h diverges

Invariant versus tracking

- Is it useful what we obtained ?
  - → Debug and compare (”benchmark”)
- Compare to very simple tracking program:
  - → linear transfer between interactions
  - → beam-beam kick for round beam
  - → compute action $I = \frac{\beta^*}{2\sigma^2}(\frac{x^2}{\beta^*} + p_x^2\beta^*)$
  - → and phase $\Psi = \arctan(\frac{p_x}{x})$
  - → compare I with h

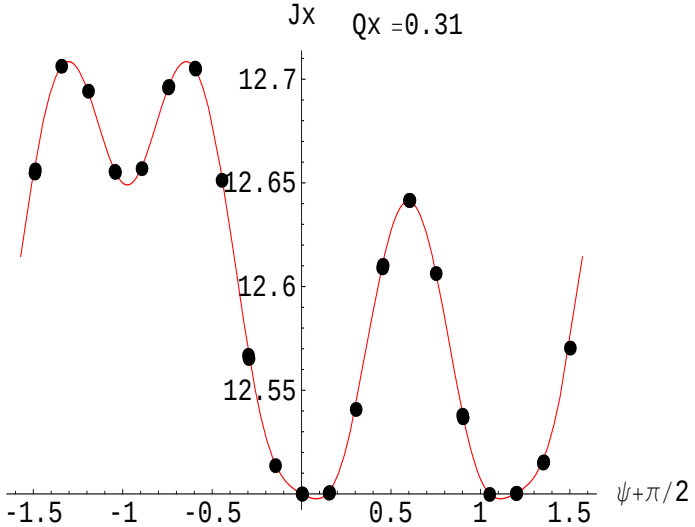
Invariant from tracking: one IP

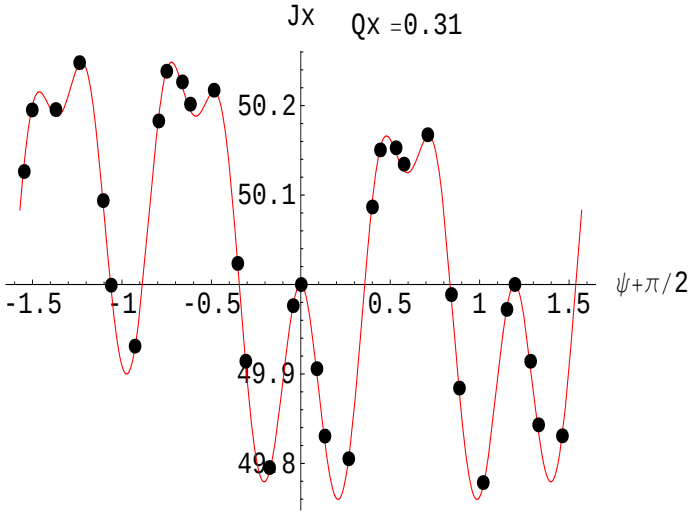




→ Shown for $5\sigma_x$ and $10\sigma_x$

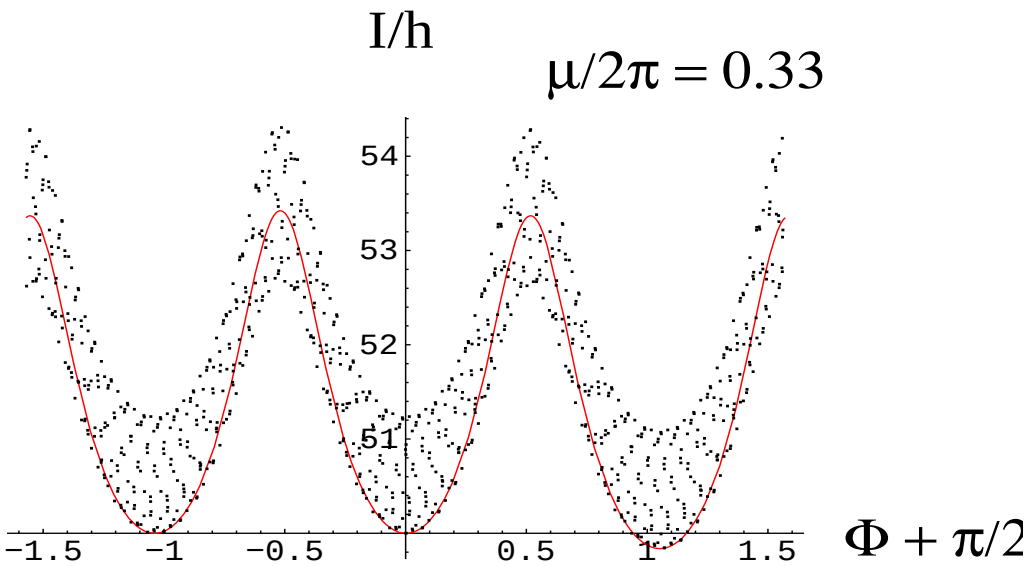
Invariant versus tracking: one IP





→ Shown for $5\sigma_x$ and $10\sigma_x$

Invariant versus tracking:



- Behaviour near a resonances: no more invariant possible
- Envelope of tracking well described

What about close to resonance ?

If we have $Q = \frac{\mu}{2\pi} \approx \frac{m}{3}$ (3rd order resonance). Using a "distance to resonance d" as:

$$Q = \frac{m+d}{3} \qquad \text{where : } \; d \ll 1$$

The trick is to observe the motion every 3 turns:

$$\mathcal{M}^3 = (e^{:-\mu J:} e^{:kx^3:})^3 = e^{:3h:}$$

We get a factor:

$$e^{:-3\mu J:} \; = \; e^{:-2\pi d J:} \quad (\text{because : } \; e^{:-2\pi m J:} \equiv 1)$$

$$d = \frac{3\mu}{2\pi}$$

Extension: general monomials

Monomials in x and p of orders n and m ($x^n p^m$)

$$e^{:ax^n p^m:}$$

gives for the map (for $n \neq m$):

$$\begin{aligned} e^{:ax^n p^m:} x &= x \cdot [1 + a(n-m)x^{n-1} p^{m-1}]^{m/(m-n)} \\ e^{:ax^n p^m:} p &= p \cdot [1 + a(n-m)x^{n-1} p^{m-1}]^{n/(n-m)} \end{aligned}$$

gives for the map (for $n = m$):

$$\begin{aligned} e^{:ax^n p^n:} x &= x \cdot e^{-anx^{n-1} p^{n-1}} \\ e^{:ax^n p^n:} p &= p \cdot e^{anx^{n-1} p^{n-1}} \end{aligned}$$

What about close to resonance ?

Without proof (but like before, see e.g. Chao), we get:

$$h = -\frac{2\pi}{3}dJ - \frac{\pi}{12}dk(2J)^{3/2} \cdot \left(\frac{sin(3\Psi + \frac{3\mu}{2})}{sin\frac{3\mu}{2}} - \frac{sin(\Psi + \frac{\mu}{2})}{sin\frac{\mu}{2}} \right)$$

For small d ($sin\frac{3\mu}{2} \approx -\pi d$) we can simplify:

$$h \approx -\frac{2\pi}{3}dJ - \frac{1}{\sqrt{2}}k(\beta J)^{3/2} sin(3\Psi)$$

Analysis give fixed points, i.e. (back in Cartesian again):

$$\frac{\partial h}{\partial x} = -\frac{2\pi}{3}dx - \frac{1}{4}\beta^{3/2}(3x'^2 - 3x^2) = 0$$

$$\frac{\partial h}{\partial x'} = -\frac{2\pi}{3}dx' - \frac{1}{4}\beta^{3/2}3xx' = 0$$