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

Werner Herr
CERN


For many more details:
http://cern.ch/Werner.Herr/METHODS
Primary purpose of this lectures

- Assumption: familiar with linear, transverse dynamics
- Need to introduce new tools for non-linear dynamics
- Avoid mathematical derivations and proofs rather give "raison d'etre" and "mode d'emploi"
- Give an overview of the modern tools used in accelerator physics
- Necessarily brief and incomplete
  -- An invitation to further studies ...

*) modern: "contemporary", not "fashionable"!
Recommended Bibliography:


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


**Why Beam Dynamics in Rings?**

- Most lectures deal with rings
- Rings are periodic systems
- Implies stability (at least for some time) and confinement

---

- This restricts the methods and tools applicable to study of beam dynamics
- Applicable to other machine and beam lines!
Outline of this lectures

- Motivation, introduction and classical concepts
- New concepts and modern techniques
  - Maps
  - Computation: maps, symplectic integration
  - Hamiltonian theory (for our purpose)
  - Analysis: Lie transforms, normal forms
  - Analysis: Differential algebra
- Identify possible traps and pitfalls ...
Treatment of **LINEAR** dynamics in rings

- Standard introduction using Hill’s equation (for simplicity: show for one dimension first):

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

- \( K(s) \) periodic, smooth function

- Is that true?

- No, normally not
Arrangement of beam line elements

- Cannot be described by Hill’s equation
- Not smooth, not periodic
Treatement of **LINEAR** dynamics in rings

Used to ”derive” Courant-Snyder ansatz:

\[ x(s) = \sqrt{\beta(s)} \cdot \epsilon \cdot \cos(\mu(s) + \mu_0) \]

\[ x'(s) = \sqrt{\frac{\epsilon}{\beta(s)}} \cdot (\sin(\mu(s) + \mu_0) + \alpha \cdot \cos(\mu(s) + \mu_0)) \]

Is the solution to any system that is: confined and periodic!

Do particles really move like this?
Not a solution of the above ..... 

What if we put additional elements (distortions ?)
Treatment of DISTORTED dynamics

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

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

or in general as (any order) multipoles:

\[ \frac{d^2 x(s)}{ds^2} + K(s)x(s) = \sum_{i,j,k,l \geq 0} p_{ijkl}(s)x^i x'^j y^k y'^l \]

Very non-linear differential equation to solve ...

→ Enter the field of non-linear dynamics
Can we deal with that?

Under certain circumstances (see lecture by Oliver Brüning):

- All $p_{ijkl}(s)$ are perturbations, i.e. (very) small
- Only a few $p_{ijkl}(s)$ are non-zero
- You can avoid resonances
- Perturbations are smooth or possibly periodic
- Perturbation treatment to leading order is sufficient

Would you build a 3 billion Euro machine on these assumptions and approximations?
What is normally not said ....

- Hill’s equation, $\beta$-function, ...etc.:
  - All concepts developed for synchrotrons! (Courant and Snyder, 1957)
- Strictly speaking, not applicable to:
  - Beam lines, LINACs, cyclotrons, ....
- Computer programs do not use Hill’s equation
- Can we find a better framework?
A disclaimer ...

- Traditional treatment requires many approximations
- Useful to understand and demonstrate concepts
- See Oliver Brüning’s, Bernhard Holzer’s lectures
- For practical work on realistic machine:
  - New tools required
  - Should exploit modern computing techniques to the maximum
  - It is much easier that you think ( .. and other people tell you !)
Start with the differential equation:

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

Bad news:
- Description not very realistic (see above)
- We have no global analytical solution

Good news: An analytical solution is not needed!
A better framework

Why not?

We do not want to know:

- The particle’s position and momentum at 2h 45min 22.3s?
  (Remember Thermodynamics!)

We do want to know:

- Is the beam stable for a long time?
- Is the motion confined?
- Does the beam hit the target?
An every day example ...

- Not important to know trajectory as function of time
- Very important to know trajectory at end of flight
- Can we get a framework to get that (easily)?
An every day example ...

- Not important to know trajectory as function of time
- Very important to know trajectory at end of flight
- Can we get a framework to get that (easily) ?
- Yes we can ! Should not go back 50 years !
A better framework - go back 100 years ...

"Old" to "New" classical dynamics:
- Topology and properties of phase space (see Oliver’s lecture)
- Chaotic motion, non-integrable systems
- Sensitivity to initial conditions
How is an beam line described?

- Beam line (or ring) made of machine elements and drifts
- Described by maps for magnets ($\mathcal{M}$) and drifts ($\mathcal{D}$)
How can an element really be described?

- You need to describe what happens to the particle in $M$ and in the drifts $D$

- In general: $\vec{z}_2 \neq \vec{z}_1$
How is an element described?

Let $\vec{z}_1, \vec{z}_2$ describe a quantity (coordinates, beam sizes ...) before and after the element.

Take an machine element (e.g. magnet) and build a mathematical model $\mathcal{M}$.

\[ \vec{z}_2 = \mathcal{M}(\vec{z}_1) \]

$\mathcal{M}$ is a so-called map.

Very important: no need to know what happens in the rest of the machine!!

The complete sequence of MAPS connects the pieces together to make a ring (or beam line).
MAPS transform coordinates through an element

- Use coordinate vector: \( \vec{z}' = (x, \ x' = \frac{\partial x}{\partial s}, \ y, \ y' = \frac{\partial y}{\partial s}) \) *

- \( M_3 \) transforms the coordinates \( \vec{z}_1(s_1) \) through the magnet \( M_3 \) at position \( s_1 \) to new coordinates \( \vec{z}_2(s_2) \) at position \( s_2 \):

\[
\vec{z}_2(s_2) = \begin{pmatrix} x \\ x' \\ y \\ y' \end{pmatrix}_{s_2} = M_3 \circ \begin{pmatrix} x \\ x' \\ y \\ y' \end{pmatrix}_{s_1} = M_3 \circ \vec{z}_1(s_1)
\]

* not unique, see later
MAPS transform coordinates through an element

The MAP fully describes what happens inside the magnet
What can $\mathcal{M}$ be?

- Any "description" to go from $\vec{z}_1$ to $\vec{z}_2$

- This "description" can be:
  - A simple linear matrix or transformation
  - A non-linear transformation (Taylor series, Lie Transform ...)
  - High order integration algorithm
  - A computer program, subroutine etc.

- Let us look at linear theory first!

  Then generalize to non-linear theory
Simple examples (one dimensional)

First a drift space of length \( L \)

Two possible descriptions are (there are more):

1. Go straight from \( s_1 \) to \( s_2 \)!!

2. More formal:

\[
\begin{pmatrix}
x \\
x'
\end{pmatrix}_{s_2} = \begin{pmatrix}
1 & L \\
0 & 1
\end{pmatrix} \circ \begin{pmatrix}
x \\
x'
\end{pmatrix}_{s_1}
\]
Simple examples (one dimensional)

Focusing quadrupole of length $L$ and strength $k$:

\[
\begin{pmatrix}
    x \\
    x'
\end{pmatrix}_{s_2} =
\begin{pmatrix}
    \cos(L \cdot k) & \frac{1}{k} \cdot \sin(L \cdot k) \\
    -k \cdot \sin(L \cdot k) & \cos(L \cdot k)
\end{pmatrix}
\circ
\begin{pmatrix}
    x \\
    x'
\end{pmatrix}_{s_1}
\]

Quadrupole with short length $L$ (i.e.: $1 \gg L \cdot k^2$)

\[
\begin{pmatrix}
    x \\
    x'
\end{pmatrix}_{s_2} =
\begin{pmatrix}
    1 & 0 \\
    -k^2 \cdot L (= \frac{1}{f}) & 1
\end{pmatrix}
\circ
\begin{pmatrix}
    x \\
    x'
\end{pmatrix}_{s_1}
\]

They are $\mathcal{M}$aps, describe the movement in an element (quadrupole)
Interlude: there was already a trap ... !

According to B. Holzer (lectures) or K. Wille (textbook):

\[ k = \frac{1}{B \rho} \frac{dB_y}{dx} \]

According to ”Handbook for Accelerator Physics” ([AC2]):

\[ k^2 = \frac{1}{B \rho} \frac{dB_y}{dx} \]

→ The lesson: check what people use !!

(remember Air Canada 143)
You also find (and it may even be useful ...):

\[ K^2 = k = \frac{1}{B \rho} \frac{d B_y}{d x} \]

Often different conventions in simulation programs!

Some programs want fields, not gradients!

Found this construction:

\[ B_y = \frac{1}{0.1} \cdot k \cdot x \cdot B \cdot \rho \]
Interlude: what about 3D ... ?

Formally extended by adding more variables:

- $(x, x', y, y', \Delta s, \frac{\Delta p}{p})$
- $\Delta s = c\Delta t$: longitudinal displacement with respect to reference particle
- $\frac{\Delta p}{p}$: relative momentum difference with respect to reference particle

Not all programs use this, but rather canonical variables

- $(x, p_x/p_s, y, p_y/p_s, -c\Delta t, p_t = \frac{\Delta E}{p_sc})$

$p_s$ may be: $p_s = p_0$ or $p_s = p_0(1 + \delta_s) = m\beta_s\gamma_s$

$\delta_s$: difference between reference momentum and design momentum
Putting the ”pieces” together

- We have to deal with many elements in our machines

- To make a ring or beam line:
  - Combine all elements maps together
  - Concatenated maps are a map again
  - Represents a bigger part of the machine (or the whole machine ...)

How is an beam line described?

Beam line (or ring) is combination of all elements

\[ M_{\text{all}} = M_4 \circ D_3 \circ M_3 \circ D_2 \circ M_2 \circ D_1 \circ M_1 \]
Putting the "pieces" together

Starting from a position $s_0$ and applying all maps (for $N$ elements) in sequence around a ring with circumference $C$ to get the One-Turn-Map (OTM) for the position $s_0$ (for one dimension only):

$$
\begin{pmatrix}
  x \\
  x'
\end{pmatrix}_{s_0 + C} = M_1 \circ M_2 \circ \ldots \circ M_N \circ 
\begin{pmatrix}
  x \\
  x'
\end{pmatrix}_{s_0}
$$

$$
\Rightarrow 
\begin{pmatrix}
  x \\
  x'
\end{pmatrix}_{s_0 + C} = M_{ring}(s_0) \circ 
\begin{pmatrix}
  x \\
  x'
\end{pmatrix}_{s_0}
$$
What does $M_{ring}$ do?

Transforms coordinates in phase space once per turn
Analysis of the One-Turn-Map

- We have obtained a map for the whole ring
- In simplest (linear) case: multiply matrices to get a One-Turn-Matrix
- Have to get now the information we want:
  - Optics parameters (Tune, Twiss functions, ..)
  - Closed orbit
  - Stability
  - etc. ...
- How to analyse a MAP (first: a matrix) ???

(see also B. Holzer lecture, but practice comes here)
Normal forms

- Maps can be transformed into (Jordan) Normal Forms
- Original maps and normal form are equivalent, but ...
- Easily used to analyse the maps:
  - Get parameters (Q, Q’, Twiss function, ..)
  - Study invariants, etc.
  - Stability
  - For resonance analysis
  - etc. ...
- Idea is to make a transformation to get a simpler form for the map
Normal forms

Assume the map $\mathcal{M}_{12}$ propagates the variables from location 1 to location 2, we try to find transformations $A_1, A_2$ such that:

$$A_1 \mathcal{M}_{12} A_2^{-1} = \mathcal{R}_{12}$$

- The map $\mathcal{R}_{12}$ is:
  - A ”Jordan Normal Form”, (or at least a very simplified form of the map)
  - Example: $\mathcal{R}_{12}$ becomes a pure rotation

- The map $\mathcal{R}_{12}$ describes the same dynamics as $\mathcal{M}_{12}$, but:
  - All coordinates are transformed
  - The transformations $A_1, A_2$ ”analyse” the motion
Normal forms - linear case

Pictorial form of the transformation

Motion on a complicated ellipse becomes motion on a circle (i.e. a pure rotation)
Normal forms - linear case

\[ M = \mathcal{A} \circ \mathcal{R}(\Delta \mu) \circ \mathcal{A}^{-1} \quad \text{or:} \quad \mathcal{R}(\Delta \mu) = \mathcal{A}^{-1} \circ M \circ \mathcal{A} \]
Normal forms - linear case (1D)

Assume the one-turn-map (here a matrix) $M(s)$ at the position $s$ is (e.g. lecture on transverse dynamics):

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

Describes the motion on a phase space ellipse

Re-write $M$ such that one part $R$ becomes a pure rotation (a circle), i.e.:

$$ARAR^{-1} = M$$

How? Remember lectures on Linear Algebra (Eigenvectors, Eigenvalues ...)
Normal forms - linear case (1D)

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

with

\[ A = \begin{pmatrix} \sqrt{\beta(s)} & 0 \\ -\alpha \sqrt{\beta} & 1 \end{pmatrix} \quad \text{and} \quad R = \begin{pmatrix} \cos(\Delta \mu) & \sin(\Delta \mu) \\ -\sin(\Delta \mu) & \cos(\Delta \mu) \end{pmatrix} \]
Normal forms - linear case (1D)

We had:

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

with

\[ A = \begin{pmatrix} \sqrt{\beta(s)} & 0 \\ -\frac{\alpha}{\sqrt{\beta}} & \frac{1}{\sqrt{\beta(s)}} \end{pmatrix} \quad \text{and} \quad \mathcal{R} = \begin{pmatrix} \cos(\Delta \mu) & \sin(\Delta \mu) \\ -\sin(\Delta \mu) & \cos(\Delta \mu) \end{pmatrix} \]

This is just the Courant-Snyder transformation to get \( \beta, \alpha, \ldots \) etc., \( \Delta \mu \) is the tune!

That is: the Courant-Snyder analysis is just a normal form transform of the linear one turn matrix

Works in more than one dimension
Normalized variables:

Please note that:

\[
\begin{pmatrix}
  x_n \\
  x' \end{pmatrix} = A^{-1} \circ \begin{pmatrix}
  x \\
  x' \end{pmatrix}
\]

is just a variable transformation to new, normalized variables.

- Tune \((\Delta \mu)\) in the **normalized map**, stability for real values of phase advance \((\Delta \mu)\)
- Optical functions \((\beta, \alpha, ...\)) in the **normalizing map**
- No need to make any assumptions, ansatz, approximation, ...

Optical functions \((\beta, \alpha, ...\)) in the **normalizing map**
Once the particles "travel" on a circle:

- Radius (say: $\sqrt{2J}$, with $J = \frac{x_n^2 + x_n'^2}{2}$) is constant (invariant of motion): action $J$
- Phase advances by constant amount: angle $\Psi$
Another example: coupling (2D)

Assume a one-turn-matrix in 2D:

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

$M,m,N,n$ 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:

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

The matrix $R$ is our simple rotation:

- $A$ and $B$ are the one-turn-matrices for the ”normal modes”

- 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:

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

The last 2 slides: normally 1 hour lecture
Normal forms - linear case

This is extremely useful when map is applied $k$ times (e.g. $k$ turns):

$$M^k(x, x') = AR^k A^{-1}(x, x') = AR^k(X, X')$$

- For multi-turns: study effect of map in normalized coordinates
- Multiplying a matrix $k$ (e.g. 4x4) can be quite a job!
- Easier to apply $k$ times using the simple map (e.g. a rotation of $\mu$ becomes just a rotation $k \cdot \mu$)
- The $A$ just transforms back to physical coordinates at the end (once!)
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 !
The 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!
The general philosophy

- Linear elements
  - Linear map
    - Linear One-Turn-Map
      - Linear Normal Form
  - Non-linear map
    - Non-linear One-Turn-Map
      - Non-linear Normal Form

General formalism for all cases!
A small complication ...

Non-linear maps are not matrices!
Various types of non-linear MAPS

Choice depends on the application

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

(A key concept: Symplecticity)

Not all possible maps are allowed!

Requires for a matrix \( \mathcal{M} \rightarrow \mathcal{M}^T \cdot S \cdot \mathcal{M} = S \)

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

\[
\lim_{n \to \infty} \mathcal{M}^n = \text{finite} \quad \implies \quad \det \mathcal{M} = 1
\]
Introducing non-linear elements

Effect of a (short) quadrupole depends linearly on amplitude (re-written from the matrix form):

\[ \vec{z}(s_2) = \begin{pmatrix} x \\ x' \\ y \\ y' \end{pmatrix}_{s_2} = \begin{pmatrix} x \\ x' \\ y \\ y' \end{pmatrix}_{s_1} + \begin{pmatrix} 0 \\ k_1 \cdot x_{s_1} \\ 0 \\ k_1 \cdot y_{s_1} \end{pmatrix} \]

\[ \vec{z}(s_2) = \mathbf{M} \cdot \vec{z}(s_1) \]

\[ \mathbf{M} \text{ is a matrix} \]
Non-linear elements (e.g. sextupole)

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

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

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

$\Rightarrow \mathcal{M}$ is not a matrix, i.e. cannot be expressed by matrix multiplication
Non-linear elements

Cannot be written in linear matrix form!
We need something like:

\[ z_1(s_2) = x(s_2) = R_{11} \cdot x + R_{12} \cdot x' + R_{13} \cdot y + .... \]
\[ + T_{111} \cdot x^2 + T_{112} \cdot xx' + T_{122} \cdot x'^2 + \]
\[ + T_{113} \cdot xy + T_{114} \cdot xy' + .... \]
\[ + U_{1111} \cdot x^3 + U_{1112} \cdot x^2x' + .... \]

and the equivalent for all other variables ...
Higher order (Taylor -) MAPS:

We have (for: \( j = 1 \ldots 4 \)):

\[
zh(s_2) = \sum_{k=1}^{4} R_{jk} z_k(s_1) + \sum_{k=1}^{4} \sum_{l=1}^{4} T_{jkl} z_k(s_1) z_l(s_1)
\]

Let’s call it: \( A_2 = [R, T] \) (second order map \( A_2 \))

Higher orders can be defined as needed ...

\[
A_3 = [R, T, U] \implies + \sum_{k=1}^{4} \sum_{l=1}^{4} \sum_{m=1}^{4} U_{jklm} z_k(s_1) z_l(s_1) z_m(s_1)
\]
Higher order (Taylor -) MAPS:

Example: complete second order map for a (thick) sextupole with length $L$ and strength $K$ (in 4D):

\[
\begin{align*}
x_2 &= x_1 + Lx'_1 - K\left(\frac{L^2}{4}(x_1^2 - y_1^2) + \frac{L^3}{6}(x_1x'_1 - y_1y'_1) + \frac{L^4}{24}(x'_1 - y'_1)^2\right) \\
x'_2 &= x'_1 - K\left(\frac{L^2}{2}(x_1^2 - y_1^2) + \frac{L^2}{2}(x_1x'_1 - y_1y'_1) + \frac{L^3}{6}(x'_1 - y'_1)^2\right) \\
y_2 &= y_1 + Ly'_1 + K\left(\frac{L^2}{4}x_1y_1 + \frac{L^3}{6}(x_1y'_1 + y_1x'_1) + \frac{L^4}{24}(x'_1y'_1)\right) \\
y'_2 &= y'_1 + K\left(\frac{L^2}{2}x_1y_1 + \frac{L^2}{2}(x_1y'_1 + y_1x'_1) + \frac{L^3}{6}(x'_1y'_1)\right)
\end{align*}
\]

⚠️ Definition of $K$ not unique, can differ by some factor !!

\[
e.g. \quad \left(\frac{\partial^2 x}{\partial t^2} = S \cdot x^2 \quad \text{versus} \quad \frac{\partial^2 x}{\partial t^2} = \frac{k}{2} \cdot x^2\right)
\]
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:

  \[
  \mathcal{J}_{ik} = \frac{\partial z^i_2}{\partial z^k_1} \quad \left( \text{e.g. } \mathcal{J}_{xy} = \frac{\partial z^x_2}{\partial z^y_1} \right)
  \]

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

- In general: $\mathcal{J}_{ik} \neq \text{const} \quad \rightarrow \quad \text{for truncated Taylor map can be difficult to fulfil for all } \mathcal{z}$
Symplecticity for higher order MAPS

Take the sextupole map (for simplicity in one dimension):

\[
x_2 = x_1 + Lx'_1 - K \left( \frac{L^2}{4} x_1^2 + \frac{L^3}{6} x_1 x'_1 + \frac{L^4}{24} x'_1^2 + \mathcal{O}(3) \right)
\]

\[
x'_2 = x'_1 - K \left( \frac{L^2}{2} x_1^2 + \frac{L^2}{2} x_1 x'_1 + \frac{L^3}{6} x'_1^2 + \mathcal{O}(3) \right)
\]

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:

\[
\Delta S = \frac{K^2}{72} L^4 (L^2 x'_2 + 6 L x x'_1 + 6 x^2)
\]
Symplecticity for higher order MAPS

Take the sextupole map (for simplicity in one dimension):

\[
x_2 = x_1 + Lx'_1 - K \left( \frac{L^2}{4} x_1^2 + \frac{L^3}{6} x_1 x'_1 + \frac{L^4}{24} x'_1^2 + \mathcal{O}(3) \right)
\]
\[
x'_2 = x'_1 - K \left( \frac{L^2}{2} x_1^2 + \frac{L^2}{2} x_1 x'_1 + \frac{L^3}{6} x'_1^2 + \mathcal{O}(3) \right)
\]

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:

\[
\Delta S = \frac{K^2}{72} L^4 (L^2 x'_2 + 6 L x x' + 6 x^2)
\]
The way out: thin magnets

- Real magnets have a finite length, i.e. thick magnets
- Thick magnet: field and length used to compute effect, i.e. the map
- Consequence: they are not always linear elements (even not dipoles, quadrupoles)
- For thick, non-linear magnets closed solution for maps often does not exist
Thick versus thin magnets

- Thin "magnet": let the length go to zero, but keep field integral finite (constant)
- Thin dipoles and quadrupoles are linear elements
- Thin elements are much easier to use ...
No change of amplitudes $x$ and $y$

The momenta $x'$ and $y'$ receive an amplitude dependent deflection (kick)

$x' \rightarrow x' + \Delta x'$ and $y' \rightarrow y' + \Delta y'$
Can we approximate a thick element by thin element(s)?

- Yes, when the length is small or does not matter
- Yes, when we can model the thick magnet correctly
- What about accuracy, symplecticity etc.??
- Demonstrate with some simple examples
Thick $\rightarrow$ thin quadrupole

\[ M_{s \rightarrow s+L} = \begin{pmatrix} \cos(L \cdot K) & \frac{1}{K} \cdot \sin(L \cdot K) \\ -K \cdot \sin(L \cdot K) & \cos(L \cdot K) \end{pmatrix} \]

- Exact map (matrix) for quadrupole
- What happens when we make it thin?
  - Accuracy?
  - Symplecticity?
- (What follows is valid for all elements)
Accuracy of thin lenses

\[ M_{s \rightarrow s+L} = \begin{pmatrix}
\cos(L \cdot K) & \frac{1}{K} \cdot \sin(L \cdot K) \\
-K \cdot \sin(L \cdot K) & \cos(L \cdot K)
\end{pmatrix} \]

- Start with exact map
- Taylor expansion in "small" length \( L \):

\[
L^0 \cdot \begin{pmatrix}
1 & 0 \\
0 & 1
\end{pmatrix} + L^1 \cdot \begin{pmatrix}
0 & 1 \\
-K^2 & 0
\end{pmatrix} + L^2 \cdot \begin{pmatrix}
-K^2/2 & 0 \\
0 & -K^2/2
\end{pmatrix} + \ldots
\]
Keep up to first order term in $L$

$$M_{s \rightarrow s + L} = L^0 \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + L^1 \cdot \begin{pmatrix} 0 & 1 \\ -K^2 & 0 \end{pmatrix}$$

$$M_{s \rightarrow s + L} = \begin{pmatrix} 1 & L \\ -K^2 \cdot L & 1 \end{pmatrix} + O(L^2)$$

Precise to first order $O(L^1)$

$\det M \neq 1$, non-symplectic
Accuracy of thin lenses (C)

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

\[ \mathcal{M}_{s \rightarrow s+L} = \begin{pmatrix} 1 & L \\ -K^2 \cdot L & 1 - K^2 L^2 \end{pmatrix} \]

\begin{itemize}
  \item Precise to first order \( \mathcal{O}(L^1) \)
  \item "symplectified" by adding term \(-K^2 L^2\)
  \item (it is wrong to \( \mathcal{O}(L^2) \) anyway ...)
\end{itemize}
Accuracy of thin lenses

Keep up to second order term in $L$

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

Precise to second order $\mathcal{O}(L^2)$

More accurate than (C), but not symplectic
Accuracy of thin lenses (D)

Symplectification like:

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

- Precise to second order $\mathcal{O}(L^2)$
- Fully symplectic
Accuracy of thin lenses

- 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

\[ K^2 \]

quadrupole of finite length

options:

Which is a good strategy? → accuracy and speed
Thick quadrupole..
First order..

One thin quadrupole "kick" and one drift combined

Resembles "symplectification" of type (C)

\[
\begin{pmatrix}
1 & 0 \\
-K^2 \cdot L & 1
\end{pmatrix}
\begin{pmatrix}
1 & L \\
0 & 1
\end{pmatrix}
= \begin{pmatrix}
1 & L \\
-K^2 \cdot L & 1 - K^2 L^2
\end{pmatrix}
\]
One thin quadrupole "kick" between two drifts

Resembles more accurate, symplectic model of type (D)

\[
\begin{pmatrix}
1 & \frac{1}{2}L \\
0 & 1
\end{pmatrix}
\begin{pmatrix}
1 & 0 \\
-K^2 \cdot L & 1
\end{pmatrix}
\begin{pmatrix}
1 & \frac{1}{2}L \\
0 & 1
\end{pmatrix}
= 
\begin{pmatrix}
1 - \frac{1}{2}K^2 L^2 & L - \frac{1}{4}K^2 L^3 \\
-K^2 \cdot L & 1 - \frac{1}{2}K^2 L^2
\end{pmatrix}
\]
Accuracy of thin lenses

One kick at the end (or beginning):

Error (inaccuracy) of first order $O(L^1)$

One kick in the centre:

Error (inaccuracy) of second order $O(L^2)$

It is very relevant how to apply thin lenses

Aim should be to be precise and fast (and simple to implement)
What is the point???

Phase ellipse - quadrupole exact solution
**Quadrupole non-symplectic solution**

- Non-symplecticity: particles spiral towards outside
Quadrupole symplectic $\mathcal{O}(L^1)$ solution

(symplecticity: but phase space ellipse not accurate)
Quadrupole symplectic $O(L^2)$ solution

\[ O(1) \]
\[ O(2) \]

> symplecticity: phase space ellipse accurate enough
Can we do better?

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

Question: is one of the options obviously wrong? If yes, why?
Can we do better?

Try a model with 3 kicks:

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

- Optimize kicks $c_1, c_2, c_3$
- Optimize drifts $d_1, d_2, d_3, d_4$
Can we do better?

Try a model with 3 kicks:

\[
\begin{align*}
\alpha K^2 L & \quad \alpha K^2 L
\end{align*}
\]

with:

\[
\begin{align*}
a & \approx 0.6756, \quad b \approx -0.1756, \\
\alpha & \approx 1.3512, \quad \beta \approx -1.7024
\end{align*}
\]

we have a $\mathcal{O}(4)$ integrator ...

(a $\mathcal{O}(6)$ integrator would require 9 kicks (!) ...)
Can we do better?

Try a model with 3 kicks:

- Must track backwards! Change interpretation!
- Thin lenses not a new sequence of magnets (a la MAD)
- What about space charge calculations?
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 $= 3 \times 1$ kick) like:

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

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

Can be considered as an iterative scheme (see e.g. H. Yoshida, 1990, E. Forest, 1998\textsuperscript{2})�:

\textbf{If }\mathcal{S}_{2k}(t)\textbf{ is a symmetric integrator of order }2k, \textbf{then:}

\[\mathcal{S}_{2k+2}(t) = \mathcal{S}_{2k}(x_1 t) \circ \mathcal{S}_{2k}(x_0 t) \circ \mathcal{S}_{2k}(x_1 t)\]

\textbf{with:}

\[x_0 = \frac{-2^{k+1} \sqrt{2}}{2 - 2^{k+1} \sqrt{2}}, \quad x_1 = \frac{1}{2 - 2^{k+1} \sqrt{2}}\]

\textbf{Higher order integrators can be obtained in a similar way}

\textsuperscript{2) E. Forest, ”Beam Dynamics, A New Attitude and Framework”, 1998}
**Symplectic integration**

Example: From a 4th order to 6th order

\[ S_6(t) = S_4(x_1 t) \circ S_4(x_0 t) \circ S_4(x_1 t) \]

We get 3 times 4th order with 3 kicks each, we have the 9 kick, 6th order integrator mentioned earlier
Integrator of order 6

- Requires 9 kicks
- We have 3 interleaved 4th order integrators
- Can be used in iterative scheme
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 !!
  - 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 thin lenses

What about accuracy of non-linear elements?

assume a general case:

\[ x'' = f(x) \]

\< Disadvantage: usually a closed solution through the element does not exist, integration necessary \>

\< Advantage: They 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 first ...
Accuracy of thin lenses - our $\mathcal{O}(2)$ model

1. Step

\[
\begin{pmatrix}
x \\
x'
\end{pmatrix}_{s_1+L/2} = \begin{pmatrix} 1 & \frac{L}{2} \\ 0 & 1 \end{pmatrix} \circ \begin{pmatrix}
x \\
x'
\end{pmatrix}_{s_1}
\]

2. Step

\[
\begin{pmatrix}
x \\
x'
\end{pmatrix}_{s_1+L/2} = \begin{pmatrix} x \\ x' + \Delta x' \end{pmatrix}_{s_1+L/2}
\]

3. Step

\[
\begin{pmatrix}
x \\
x'
\end{pmatrix}_{s_1+L} = \begin{pmatrix} 1 & \frac{L}{2} \\ 0 & 1 \end{pmatrix} \circ \begin{pmatrix}
x \\
x'
\end{pmatrix}_{s_1+L/2}
\]
Accuracy of thin lenses

Assume the general case:

\[ x'' = f(x) \equiv \Delta x' \]

Using this thin lens approximation (type D, O(2)) gives:

\[ x'(L) \approx x_0' + L f(x_0 + \frac{L}{2} x_0') \]
\[ x(L) \approx x_0 + \frac{L}{2} (x_0' + x'(L)) \]

This is also called ”leap frog” algorithm/integration
It is symplectic (... and time reversible) !!!
For any: \( x'' = f(x, x', t) \) we can solve it by:

\[
\begin{align*}
    x'_{i+3/2} & \approx x'_{i+1/2} + f(x_{i+1})\Delta t \\
    x_{i+1} & \approx x_i + x'_{i+1/2} \Delta t
\end{align*}
\]
Accuracy of thin lenses

Accuracy of "leap frog" algorithm/integration"

the (exact) Taylor expansion gives:

\[ x(L) = x_0 + x'_0 L + \frac{1}{2} f(x_0) L^2 + \frac{1}{6} x'_0 f'(x_0) L^3 + \ldots \]

the (approximate) "leap frog" algorithm gives:

\[ x(L) = x_0 + x'_0 L + \frac{1}{2} f(x_0) L^2 + \frac{1}{4} x'_0 f'(x_0) L^3 + \ldots \]

Errors are \( O(L^3) \) (of course)

For small \( L \) acceptable, and symplectic, extend to our symplectic integration
Accuracy of thin lenses

For bar/coffee discussions:

why did I write:

\[ x'' = f(x) \]

and not:

\[ x'' = f(x, x') \]
Accuracy of thin lenses

An application, assume a (1D) sextupole with (definition of $k$ not unique!):

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

using the thin lens approximation (type D) gives:

$$x(L) = x_0 + x'_0 L + \frac{1}{2} k x_0^2 L^2 + \frac{1}{2} k x_0 x'_0 L^3 + \frac{1}{8} k x'_0^2 L^4$$

$$x'(L) = x'_0 + k x_0^2 L + k x_0 x'_0 L^2 + \frac{1}{4} k x'_0^2 L^3$$

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

An application, assume a (1D) sextupole with (definition of $k$ not unique!):

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

using the thin lens approximation (type D) gives:

$$x(L) = x_0 + x'_0 L + \frac{1}{2} k x_0^2 L^2 + \frac{1}{8} k x_0 x'_0 L^3 + \frac{1}{4} k x'_0^2 L^4$$

$$x'(L) = x'_0 + k x_0^2 L + k x_0 x'_0 L^2 + \frac{1}{4} k x'_0^2 L^3$$

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

- We have interleaved kicks with drifts
- Is that always necessary?
  - No!
  - Can be any map with an exact expression
  - In most cases the exact map is a linear map (matrix)
- We have derived element maps for tracking from the equation of motion using this technique ➔ can track now
Simulation and tracking

We have now sufficient tools for a simulation code

- **Main purpose of such a code:** Propagate particles around a ring or along a beam line

- **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)!

- **Can we extract more ”analytical” maps ?**
So far ...

- Concept and representation by MAPS
- Computation and analysis of One-Turn-Maps
- Normal form analysis of LINEAR MAPS
- Introduction of Taylor maps
- Introduction of symplectic integration
Mathematical and Numerical Methods for Non-linear Beam Dynamics in Rings (an introduction)

Part 2

Werner Herr, CERN


For many more details:
http://cern.ch/Werner.Herr/METHODS

Werner Herr, non-linear methods, CAS 2013, Trondheim
The plan now ...

- Extend all concepts to non-linear dynamics
  - Lagrangian and Hamiltonian dynamics
  - How to use that → Lie transforms
  - How to analyse that → Non-linear normal forms
  - How to analyse that better → Differential Algebra (DA)

- Avoid abstract definitions and formulation, but:
  - Intuitive (but correct !) treatment
  - Useful formulae and examples
  - Real life examples and demonstration (DA)
Hamilton principle

Problem: describe the motion of a system (e.g. 1 or more particles) between times $t_1$ and $t_2$

Describe by coordinates $q_i$ \hspace{1cm} (i = 1, n)

$n$ are degrees of freedom of the system

(Goldstein convention)
Hamilton principle

- Describe motion by a function $L$

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

$(q_1, ... q_n)$ ... generalized coordinates

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

- The function $L$ defines the Lagrange function

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

Without proof or derivation:

$L = T - V = \text{kinetic energy} - \text{potential energy}$
Hamilton principle

\[ I = \int_1^2 L(q_i, \dot{q}_i, t) dt = \text{extremum} \]

\[ \Rightarrow \text{Hamiltonian principle: system moves such that the action } I \text{ becomes an extremum} \]
Extremum principle?

Not new:

- Used in optics: Fermat principle
- Quantum mechanics (path integrals)
- General relativity
- ...

...
Lagrange formalism

Without proof:

\[ I = \int_{1}^{2} L(q_i, \dot{q}_i, t) dt = \text{extremum} \]

is fulfilled when:

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

(\text{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

- Try to get:
  - Generalized momenta instead of velocities
  - First order differential equations (always solvable)

Corresponding (so-called conjugate) momenta \( p_j \) are:

\[
p_j = \frac{\partial L}{\partial \dot{q}_j}
\]
From Lagrangian to Hamiltonian ..

Lagrangian:

- $n$ second order equations
- $n$-dimensional coordinate space

Hamiltonian:

- $2n$ first order equations
- $2n$-dimensional phase space
From Lagrangian to Hamiltonian ..

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:

$H = T + V = \text{kinetic energy + potential energy}$

we obtain 2 first order equation of motion:

$$\frac{\partial H}{\partial q_j} = -\dot{p}_j = -\frac{dp_j}{dt}, \quad \frac{\partial H}{\partial p_j} = \dot{q}_j = \frac{dq_j}{dt}$$
Hamiltonian of particle in EM fields

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

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

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

Using canonical variables and the design path length \(s\) as independent variable (bending in x-plane):

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

where \(\delta = (p - p_0)/p\) is relative momentum deviation and \(A_s(x, y)\) longitudinal component of the vectorpotential [MB].
Hamiltonian of particle in EM fields

The magnetic fields can be described with the multipole expansion:

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

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

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

\( n = 1 \) refers to dipole (not always true !)

For a large machine (\( x \ll \rho \)) we expand the root and sort the variables →
Hamiltonian (for large machine) ..

\[ H = \frac{p_x^2 + p_y^2}{2(1 + \delta)} - \frac{x\delta}{\rho} + \frac{x^2}{2\rho^2} + \frac{k_1}{2}(x^2 - y^2) + \frac{k_2}{6}(x^3 - 3xy^2) \] (using (MAD convention): \( k_n = \frac{1}{B\rho} \frac{\partial^n B_y}{\partial x^n} \))

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 $g$):

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

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

Poisson brackets encode Hamilton’s equations
Poisson brackets

Poisson bracket \([f, H]\) describes the time evolution of the system (the function \(f\))

It is a special case of:

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

If \(H\) does not explicitly depend on time and:

\([f, H] = 0\)

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

Poisson brackets determine invariants
Lie transformations

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**

It acts on functions \( g(x, p) \), special cases:

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

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

**Lie operators are Poisson brackets in waiting**
Useful formulae for calculations

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

\[ x : = \frac{\partial}{\partial p} \quad \quad p : = - \frac{\partial}{\partial x} \]

\[ x^{2} : = \frac{\partial^{2}}{\partial p^{2}} \quad \quad p^{2} : = \frac{\partial^{2}}{\partial x^{2}} \]

\[ x^{2} : = 2x \frac{\partial}{\partial p} \quad \quad p^{2} : = -2p \frac{\partial}{\partial x} \]

\[ xp : = p \frac{\partial}{\partial p} - x \frac{\partial}{\partial x} \quad \quad x :: p : = p :: x : = - \frac{\partial^{2}}{\partial x \partial p} \]
More useful formulae for calculations

With $x$ coordinate, $p$ momentum, as usual:

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

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

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

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

We can define powers as:

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

In particular:

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

\[ e^{f} = 1 + :f: + \frac{1}{2!} (: f :)^2 + \frac{1}{3!} (: f :)^3 + ... \]

The operator \( e^{f} \) is called an \textbf{Lie Transformation}
Lie transformations - example

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

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

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

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

$$= x + Lp$$

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

$$= p$$
Lie transformations - example

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

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

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

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

$$= x + Lp$$

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

$$= p$$

This is the transformation of a drift space of length $L$ !!
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 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!) 
- Can track "thick" elements (and still symplectic!)
Some formulae for Lie transforms

With a constant, \( f, g, k \) arbitrary functions:

\[
\begin{align*}
: a: &= 0 & e^{a}: &= 1 \\
: f : a &= 0 & e^{f:a} &= a \\
\end{align*}
\]

\[
e^{f:g(x)} = g(e^{f:x})
\]

\[
e^{f:G(g: :)} e^{-f:} = G(e^{f:g:})
\]

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

\[
(e^{f:})^{-1} = e^{-f:}
\]

and very important:

\[
e^{f:e^g:e^{-f:}} = e^{e^{f:}g:}
\]
More examples (1D):

For:

\[ f = -\frac{L}{2}p^2 \]

we obtained:

\[ e^{if} x = x + Lp \]
\[ e^{if} p = p \]

Drift space, seen that already
More examples (1D):

For:

\[ f = -\frac{L}{2}(k^2 x^2 + p^2) \]

we would get (try it !):

\[ e^f \cdot x = e^{-\frac{L}{2}(k^2 x^2 + p^2)} \cdot x \]
\[ e^f \cdot p = e^{-\frac{L}{2}(k^2 x^2 + p^2)} \cdot p \]

Remember:

\[ e^f \cdot g = \sum_{n=0}^{\infty} \frac{f^n}{n!} g \]
More examples (1D):

For:

\[ f = -\frac{L}{2}(k^2x^2 + p^2) \]

we would get (try it!):

\[ e^{-\frac{L}{2}(k^2x^2+p^2)}: x = \sum_{n=0}^{\infty} \left( \frac{(-k^2L^2)^{2n}}{(2n)!} \cdot x + L \frac{(-k^2L^2)^{2n+1}}{(2n+1)!} \cdot p \right) \]

\[ e^{-\frac{L}{2}(k^2x^2+p^2)}: p = \sum_{n=0}^{\infty} \left( \frac{(-k^2L^2)^{2n}}{(2n)!} \cdot p - k \frac{(-k^2L^2)^{2n+1}}{(2n+1)!} \cdot x \right) \]
More examples (1D):

For:

\[ f = -\frac{L}{2}(k^2x^2 + p^2) \]

we would get (try it !):

\[ e^f : x = x \cos(kL) + \frac{p}{k} \sin(kL) \]
\[ e^f : p = -kx \sin(kL) + p \cos(kL) \]

Thick, focusing quadrupole!
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 is 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
Why all that ???

If we know the Hamiltonian of a machine elements (magnet) then:

\[ e^H x_0 = x_1 \]
\[ e^H p_0 = p_1 \]

This is also true for functions of \( x \) and \( p \):

\[ e^H f_0(x, p) = f_1(x, p) \]

The miracles:

- Poisson brackets create symplectic maps
- Exponential form \( e^h \) is always symplectic
- Better: the exponent is directly connected to the invariant of the transfer map !!
Many machine elements

We can combine many machine elements $f_n$ by applying one transformation after the other:

$$e^{hf_1} e^{hf_2} \ldots e^{hf_N}$$

Not restricted to matrices, i.e. linear elements ...

And arrive at a transformation for the full ring

→ a one turn map

The one turn map is the exponential of the effective Hamiltonian:

$$\mathcal{M}_{ring} = e^{-CH_{eff}}$$
Why all that ???

concatenation very easy:

\[ e^{\cdot h} = e^{\cdot f} e^{\cdot g} = e^{\cdot f + g} \]

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)):

\[
\begin{align*}
  h &= f + g + \frac{1}{2} [f, g] + \frac{1}{12} [f, [f, g]] + \frac{1}{12} [g, [g, f]] \\
  &\quad + \frac{1}{24} [f, [g, [g, f]]] - \frac{1}{720} [g, [g, [g, [g, f]]]] \\
  &\quad - \frac{1}{720} [f, [f, [f, [f, g]]]] + \frac{1}{360} [g, [f, [f, [f, g]]]] + \ldots
\end{align*}
\]

or:

\[
\begin{align*}
  h &= f + g + \frac{1}{2} : f : g + \frac{1}{12} : f :^2 g + \frac{1}{12} : g :^2 f \\
  &\quad + \frac{1}{24} : f : : g :^2 f - \frac{1}{720} : g :^4 f \\
  &\quad - \frac{1}{720} : f :^4 g + \frac{1}{360} : g : : f :^3 g + \ldots
\end{align*}
\]
Concatenation

To combine:

\[ e^{h} : e^{f} : e^{g} : = e^{h} : e^{f} : e^{g} : \]

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

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

General thin lens kick $f(x)$:

$$e^{\int_{0}^{x} f(x')dx'}:$$

gives for the map:

\[
\begin{align*}
x &= x_0 \\
p &= p_0 + f(x)
\end{align*}
\]

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

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

gives for the map:

\[
\begin{align*}
x &= x_0 \\
p &= p_0 + ax^n
\end{align*}
\]
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 $n \neq m$):

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

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

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

$e^{ax^n p^n} \cdot x = x \cdot e^{-anx^{n-1}p^{n-1}}$

$e^{ax^n p^n} \cdot p = p \cdot e^{anx^{n-1}p^{n-1}}$
From the Hamiltonian to the map

We have seen that given the Hamiltonian $f$ of a machine element is known, the Lie operator becomes:

$$f \rightarrow : f :$$

the corresponding map is than:

$$e^{LF} (e^{-LF})$$

This map is always symplectic and we have (in 1D):

$$e^{LF} x_0 = x_1$$

$$e^{LF} p_0 = p_1$$

or using $Z = (x, p_x, y, p_y, ...)$ (in 2D):

$$e^{LF} Z_0 = Z_1$$
From the map to the Hamiltonian

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

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

i.e.:

$$MZ_0 = Z_1$$

how do we find the corresponding form for $f$?

$$M \leftrightarrow e^{f} \quad (e^{-\mu f})$$
From the map to the Hamiltonian

For the linear matrix we know that $f$ must be a quadratic form in $(x, p, ...)$. Any quadratic form can be written as:

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

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

$$f : Z = SFZ$$

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

$$e^{f : Z} \leftrightarrow e^{SF} Z$$
From the map to the Hamiltonian

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:

$$\lambda_{\pm} = \pm i \sqrt{ac - b^2}$$
From the map to the Hamiltonian

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

\[ e^{SF} = \cos(\sqrt{ac - b^2}) + \frac{\sin(\sqrt{ac - b^2})}{\sqrt{ac - b^2}} \begin{pmatrix} b & c \\ -a & -b \end{pmatrix} \]
From the map to the Hamiltonian

For a general $2 \times 2$ matrix:

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

we get by comparison:

$$\cos(\sqrt{ac - b^2}) = \frac{1}{2} \text{tr}(M)$$

and

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

for the quadratic form of $f$:

$$f = -\frac{1}{2}(a \cdot x^2 + b \cdot xp + c \cdot p^2)$$
From the map to the Hamiltonian

For the example of a 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:

\[ f = -\frac{1}{2f}(x^2) \]
A very important example ...

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

corresponds to:

\[ e^{\mathcal{h}} = e^{\mathcal{f}_2} = e^{-\mu \frac{1}{2}(\gamma x^2 + 2\alpha xp + \beta p^2)}: \]

In this form \( f \) is: \(-\mu \cdot \text{(Courant-Snyder invariant)}\)

\[ e^{\mathcal{h}} = e^{\mathcal{f}_2} = e^{-\mu e}.: \]

We have standard \((e^{\mathcal{f}_2}:)\) for the linear one-turn-matrix (a rotation)...
A very important example ...

With our linear transformation to normalized variables:

\[
\begin{pmatrix}
\cos \mu + \alpha \sin(\mu) & \beta \sin \mu \\
-\gamma \sin \mu & \cos \mu - \alpha \sin(\mu)
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
\cos \mu & \sin \mu \\
-\sin \mu & \cos \mu
\end{pmatrix}
\]

therefore:

\[
e^{-\mu \frac{1}{2}(\gamma x^2 + 2\alpha xp + \beta p^2)}: \Rightarrow e^{-\mu \frac{1}{2}(x^2 + p^2)}:
\]

and for a 3D linear system we have for \( f_2 \):

\[
f_2 = -\frac{\mu_x}{2}(x^2 + p_x^2) - \frac{\mu_y}{2}(y^2 + p_y^2) - \frac{1}{2}\alpha_c \delta^2
\]

or in action variables \( J \):

\[
f_2 = -\mu_x J_x - \mu_y J_y - \frac{1}{2}\alpha_c \delta^2
\]

⇒ A standard \( e^{f_2} \) transformation in 3D
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 (i.e. not matrices). If $\mathcal{M}$ is our usual one-turn-map, we try to find a transformation:

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

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 $A$:

$$\mathcal{N} = e^{-h F} = A\mathcal{M}A^{-1} = e^{F : \mathcal{M} e^{-F}}.$$
Normal forms - non-linear case

More complicated transformation \( F \) required

Transform to coordinates where map is just a rotation

In general better done in action angle variables: \( J, \Psi \)

Rotation angle may be amplitude dependent: \( \mu \rightarrow \mu(J) \)
Normal forms - non-linear case

The canonical transformation $A$:

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

should be the transformation to produce our simple form

$$h(J_x, \Psi_x, J_y, \Psi_y, z, \delta) \Rightarrow h(J_x, J_y, \delta) = h_{\text{eff}}(J_x, J_y, \delta)$$

Should work for any kind of local perturbation

Formalism and software tools exist to find $F$ (see e.g. Chao\textsuperscript{1}) or E. Forest, M. Berz, J. Irwin, SSC-166)

Once we know $h_{\text{eff}}(J_x, J_y, \delta)$ we can derive everything!

\textsuperscript{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 \( J_x, J_y \), and \( \delta \), 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 \( \delta \) have different tunes:

→ Dependence on \( J \) is amplitude detuning, dependence on \( \delta \) are the chromaticities!


How does $h_{eff}$ look like?

The effective Hamiltonian can be written (here to 3rd order) (see e.g. E. Forest, M. Berz, J. Irwin, SSC-166) as:

$$h_{eff} = + \mu_x J_x + \mu_y J_y + \frac{1}{2} \alpha_c \delta^2$$

$$+ c_{x1} J_x \delta + c_{y1} J_y \delta + c_3 \delta^3$$

$$+ c_{xx} J_x^2 + c_{xy} J_x J_y + c_{yy} J_y^2 + c_{x2} J_x \delta^2 + c_{y2} J_y \delta^2 + c_4 \delta^4$$

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 ?

- $\mu_x, \mu_y$: 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
Example: sextupole

A linear map followed by a single (weak) sextupole:

\[ \mathcal{M} = e^{-\frac{\mu}{2}x^2 + p^2 + \frac{1}{2} \alpha_c \delta^2}: \quad e^{f_3}: = e^{-\mu J_x + \frac{1}{2} \alpha_c \delta^2}: \quad e^{kx^3 + \frac{p^2}{2(1+\delta)}}: \]

we get for \( h_{eff} \) (see e.g. [AC1, EF]):

\[ h_{eff} = \mu_x J_x + \frac{1}{2} \alpha_c \delta^2 - k D^3 \delta^3 - 3 k \beta_x J_x D \delta \]

or in 3D:

\[ h_{eff} = \mu_x J_x + \mu_y J_y + \frac{1}{2} \alpha_c \delta^2 - k D^3 \delta^3 - 3 k \beta_x J_x D \delta + 3 k \beta_y J_y D \delta \]
Example: sextupole

When we have $h_{eff}$ in 3D we obtain:

$$Q_x(J_x, J_y, \delta) = \frac{1}{2\pi} \frac{\partial h_{eff}}{\partial J_x} = \frac{1}{2\pi} (\mu_x - 3k\beta_x D\delta)$$

$$Q_y(J_x, J_y, \delta) = \frac{1}{2\pi} \frac{\partial h_{eff}}{\partial J_y} = \frac{1}{2\pi} (\mu_y + 3k\beta_y D\delta)$$

and the change of path length:

$$\Delta s = -\frac{\partial h_{eff}}{\partial \delta} = \alpha_c \delta - 3kD^3\delta^2 - 3kD(\beta_x J_x - \beta_y J_y)$$
Normal forms - non-linear case

Assume a linear rotation (as always) followed by an octupole, the Hamiltonian is (1D to keep it simple):

\[ \mathcal{H} = \frac{\mu}{2}(x^2 + p^2) + k_3 \cdot \frac{x^4}{4} \quad (p = p_x) \]

With the Hamilton’s equation leading to:

\[ \dot{x} = \frac{\partial \mathcal{H}}{\partial p} = \mu p \]

\[ \dot{p} = -\frac{\partial \mathcal{H}}{\partial x} = -\mu x - k_3 \cdot x^3 \]
Normal forms - non-linear case

The map, written in Lie representation is:

\[ \mathcal{M} = e^{-\frac{\mu^2}{2}}x^2 + p^2 \quad e^{k_3 \cdot \frac{x^4}{4}} = Re^{k_3 \cdot \frac{x^4}{4}} \]

we transform by applying:

\[ \mathcal{N} = \mathcal{A} \mathcal{M} \mathcal{A}^{-1} = e^{\mathcal{F}} \quad Re^{k_3 \cdot \frac{x^4}{4}} \quad e^{-\mathcal{F}} = RR^{-1} \quad e^{\mathcal{F}} \quad Re^{k_3 \cdot \frac{x^4}{4}} \quad e^{-\mathcal{F}} \]

\[ = Re^{R^{-1}F + k_3 \cdot \frac{x^4}{4} - F} + O(\epsilon^2) = Re^{(R^{-1} - 1)F + k_3 \cdot \frac{x^4}{4} + O(\epsilon^2)} \]

we have now to choose \( F \) to simplify the expression:

\[ = (R^{-1} - 1)F + k_3 \cdot \frac{x^4}{4} \]

and get [EF1, AW]:

\[ F = -\frac{1}{64} \left\{ -5x^4 + 3p^4 + 6x^2 p^2 + x^3 p \cdot (8\cot(\mu) + 4\cot(2\mu)) + xp^3(8\cot(\mu) - 4\cot(2\mu)) \right\} \]
Normal forms - non-linear case

We go back to $x$ and $p$ coordinates and with:

$$J = \frac{x^2 + p^2}{2}$$

we can write the map:

$$M = e^{-F}: e^{-\mu J + \frac{3}{8} k_3 \cdot J^2}: e^{F}:$$

the term $\frac{3}{8} k_3 \cdot J^2$ produces the tune shift with amplitude we know for an octupole ($\cdot < \beta^2 >$ in real space)

Note: the normalized map (our most simple map):

$$R = e^{-\mu J + \frac{3}{8} k_3 \cdot J^2}:$$

is again a rotation in phase space, but the rotation angle now depends on the amplitude $J$
For the tune shift: octupole (1D)

When we have $h_{eff}$ in 1D for a single octupole (see before):

$$h_{eff} = -\mu J + \frac{3}{8} k_3 \cdot J^2$$

$$Q_x(J_x, J_y) = \frac{1}{2\pi} \frac{\partial h_{eff}}{\partial J_x} = -\frac{1}{2\pi} \mu_x + \frac{3}{8 \cdot 2\pi} k_3 J_x$$

and with normalization in real space (i.e. $\beta \neq 1$):

$$\Delta Q_x(J_x, J_y) = \frac{3}{8 \cdot 2\pi} k_3 < \beta^2 > J_x$$

Example: $\beta = 300 \text{m}$, $k_3 = 0.01$

$$\Delta Q_x(J_x, J_y) = 53.7 \cdot J_x$$
A real life example: beam-beam interaction

- Localized distortion, very strong non-linearity
- Standard perturbation theory not appropriate
Effect on invariants - start with single IP

- Look for invariant $h$

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

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

with (see before):

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

and (see before):

$$F = \int_{0}^{x} dx' f(x')$$
Effect on invariants

For a Gaussian beam we have for $f(x)$ (see lecture on beam-beam effects):

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

as usual go to action angle variables $\Psi, J$:

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

and write $F(x)$ as Fourier series:

$$F(x) = \sum_{n=-\infty}^{\infty} c_n(J) e^{in\Psi}$$
We need:

REMEMBER: with this transform $f_2$ becomes very simple:

$$f_2 = -\mu J$$

and useful properties of Lie operators (any textbook\(^2\)):

$$: f_2 : g(J) = 0, \quad : f_2 : e^{i\mu \Psi} = i\mu e^{i\mu \Psi}, \quad g(: f_2 :) e^{i\mu \Psi} = g(i\mu) e^{i\mu \Psi}$$

and the formula (any textbook\(^2\)):

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

Single IP

gives immediately for $h$:

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

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

away from resonance normal form transformation gives:

$$h_n = -\mu J + c_0(J) = \text{const.}$$

$$\left[\text{homework : } \frac{dc_0(J)}{dJ}\right]$$
Single IP - analysis of $h$

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

On resonance:

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

with $c_n \neq 0$:

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

and $h$ diverges, find automatically all resonance conditions
Invariant from tracking: one IP

\[
\begin{align*}
I_x & = \nu_x = 0.31 + \pi/2 \\
\end{align*}
\]

Shown for \(5\sigma_x\) and \(10\sigma_x\)
Invariant versus tracking: one IP

Shown for $5\sigma_x$ and $10\sigma_x$
Truncated Power Series Algebra (TPSA)

- Tracking particles is very reliable method
- Simulation can produce maps for complicated configurations
- How can we analyse the map produced by a tracking code?
- Now we put the final nail into the coffin of any other approach ...
The tracking of a complicated system relates the output numerically to the input.

Could we imagine something that relates the output algebraically to the input?

For example a Taylor series?

\[ z_2 = \sum_j C_j z_1^j = \sum f^{(n)} z_1^j \]
Why are Taylor series useful?

Let us study the paraxial behaviour:

- Red line is the ideal orbit
- Blue lines are small deviations
- If we understand how small deviations behave, we understand the system much better
Why are Taylor series useful?

Now remember the definition of the Taylor series:

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

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

→ The coefficients determine the behaviour of small deviations \( \Delta x \) from the ideal orbit \( x \)

→ The Taylor expansion does a paraxial analysis of the system
Why are Taylor series useful?

If the function \( f(x) \) is represented by a Taylor series:

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

if it is truncated to the \( m \)-th order:

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

There is an equivalence between the function \( f(x) \) and the vector \( (f(a), f'(a), f''(a), ..., f^{(m)}(a)) \).

This vector is a **Truncated Power Series Algebra (TPSA)** representation of \( f(x) \) around \( a \).

How to get these coefficients without extra work?
Numerical differentiation

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

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

- Need to subtract almost equal numbers and divide by small number.

- For higher orders $f''$, $f'''$.., accuracy hopeless !

- We can use Differential Algebra (DA) (M. Berz, 1988 and [MB])
Differential Algebra

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

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

\[(q_0, q_1) + (r_0, r_1) = (q_0 + r_0, q_1 + r_1)\]
\[c \cdot (q_0, q_1) = (c \cdot q_0, c \cdot q_1)\]
\[(q_0, q_1) \cdot (r_0, r_1) = (q_0 \cdot r_0, q_0 \cdot r_1 + q_1 \cdot r_0)\]
Differential Algebra

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:

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

\[
(q_0, q_1) > (r_0, r_1) \text{ if } q_0 > r_0 \text{ or } (q_0 = r_0 \text{ and } q_1 > r_1)
\]
Differential Algebra

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:

\[(q_0, q_1) < (r_0, r_1) \text{ if } q_0 < r_0 \text{ or } (q_0 = r_0 \text{ and } q_1 < r_1)\]
\[(q_0, q_1) > (r_0, r_1) \text{ if } q_0 > r_0 \text{ or } (q_0 = r_0 \text{ and } q_1 > r_1)\]

4. This implies something strange:

\[(0, 0) < (0, 1) < (r, 0) \text{ (for any pos. } r)\]
\[(0, 1) \cdot (0, 1) = (0, 0) \quad \rightarrow \quad (0, 1) = \sqrt{(0, 0)}!!\]
Differential Algebra

This means that \((0,1)\) is between 0 and ANY real number \(\text{infinitely small}!!!\)

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..):

\[
q_0 = R(q_0, q_1) \quad \text{and} \quad q_1 = D(q_0, q_1)
\]

With our rules we can further see that:

\[
(1, 0) \cdot (q_0, q_1) = (q_0, q_1)
\]

\[
(q_0, q_1)^{-1} = \left(\frac{1}{q_0}, -\frac{q_1}{q_0^2}\right)
\]
Differential Algebra

Of course can 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 $\mathcal{D}$?
Differential Algebra

For a function \( f(x) \) without proof:

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

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:

\[
f(2) = \frac{9}{2}, \quad f'(2) = \frac{15}{4}
\]
Differential Algebra

For x in:

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

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

$$f[(2, 1)] = (2, 1)^2 + (2, 1)^{-1}$$

$$= (4, 4) + \left(\frac{1}{2}, -\frac{1}{4}\right)$$

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

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

   \[s_i = \sum_{k=0}^{i} \frac{i!}{k!(i-k)!} q_k r_{i-k}\]
Differential Algebra

If we had started with:

\[ x = (a, 1, 0, 0, 0...) \]

we would get:

\[ f(x) = (f(a), f'(a), f''(a), f'''(a), \ldots f^{(n)}(a)) \]

can be extended to more variables \( x, y \):

\[ x = (a, 1, 0, 0, 0...) \quad dx = (0, 1, 0, 0, 0, \ldots) \]

\[ y = (b, 0, 1, 0, 0...) \quad dy = (0, 0, 1, 0, 0, \ldots) \]

and get (with more complicated multiplication rules):

\[ f((x + dx), (y + dy)) = \left( f, \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial^2 f}{\partial x^2}, \frac{\partial^2 f}{\partial x \partial y}, \ldots \right)(x, y) \]
What is the use of that:

Can extract a truncated Taylor map of a beam line or ring by pushing the identity map \( f(x) = (a, 1, 0, 0, 0...) \) through the algorithm as if it is a vector in phase space! The maps are provided with the desired accuracy and to any order.
What is the use of that:

\[ \text{Input } z_1 \xrightarrow{\text{Algorithm}} \text{Output } z_2 \]

- "Algorithm" can be a mathematical function
- "Algorithm" can be a complex computer code
- Easy using polymorphism of modern languages (see example)
- Normal form analysis on Taylor series is much easier!!
- We get a Taylor map for a computer code!!!
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/CAS2013/DA
  
  Small DA package provided by E. Forest
Look at this small example:

PROGRAM DATEST1
use my_own_da
real (8) 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 (8) 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

2) Courtesy E. Forest for the small DA package used here ...
Look at the results:

(0,0) 0.50000000000000E+00
(1,0) 0.86602540378440E+00
(0,1) 0.00000000000000E+00
(2,0) -0.25000000000000E+00
(0,2) 0.00000000000000E+00
(1,1) 0.00000000000000E+00
(3,0) -0.14433756729741E+00
(0,3) 0.00000000000000E+00
(2,1) 0.00000000000000E+00
(1,2) 0.00000000000000E+00

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

\[
\begin{align*}
(0,0) & \quad 0.5000000000000E+00 \\
(1,0) & \quad 0.8660254037844E+00 \\
(0,1) & \quad 0.0000000000000E+00 \\
(2,0) & \quad -0.2500000000000E+00 \\
(0,2) & \quad 0.0000000000000E+00 \\
(1,1) & \quad 0.0000000000000E+00 \\
(3,0) & \quad -0.1443375672974E+00 \\
(0,3) & \quad 0.0000000000000E+00 \\
(2,1) & \quad 0.0000000000000E+00 \\
(1,2) & \quad 0.0000000000000E+00
\end{align*}
\]

\[
\sin\left(\frac{\pi}{6} + \Delta x\right) = \sin\left(\frac{\pi}{6}\right) + \cos\left(\frac{\pi}{6}\right)\Delta x - \frac{1}{2}\sin\left(\frac{\pi}{6}\right)\Delta x^2 - \frac{1}{6}\cos\left(\frac{\pi}{6}\right)\Delta x^3
\]
What is the use of that:

- We have used a simple algorithm here \((\text{sin})\) but it can be anything very complex.

- 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 \((+, -, *, =, ..., \text{exp, sin}, ...)\) 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:**

\[ X_n = (x, p_x, y, p_y, s, \delta)_n \rightarrow X_{n+1} = \sum C_j X^j_n \]

- Taylor expansion after one completed turn
- Automatically all \( X_{n+1} \) where it converges
- The \( C_j \) contain useful information about behaviour
- Taylor map directly used for normal form analysis
Another example:

- Track through a FODO lattice:
  - QF - DRIFT - QD
  - Integrate 100 steps in the quadrupoles
  - Now we use three variables:
    - $x, p, \Delta p = (z(1), z(2), z(3))$
Another example:

```fortran
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

doi=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)
  z(1)=z(1)+LC*z(2) ! drift of half cell length
  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 fodo1
```

2) Courtesy E. Forest for the small DA package used here ...
The result is:

Only linear 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 = 0.094425
Chromaticity = -0.097295

\(^2\text{Courtesy E. Forest for the small DA package used here ...} \)
Modified previous example (with one octupole):

```fortran
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))  ! 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%dttune_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) -0.364344875882E+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

Now non-linear elements in the Taylor expansion,

The output from the normal form analysis are (per cell !):
Tune = 0.094425
Chromaticity= -0.097295
The detuning with amplitude is 53.74 !

2) Courtesy E. Forest for the small DA package used here ...
Modified previous example (with octupole):

Remember the normal form transformation:

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

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

\[ j_2 = (x**2+p**2) \times \text{NORMAL}\%A\%^{-1} \]

(remember: $x**2+p**2$ is the tilted ellipse ....
Can get the optical functions out because

- $\beta$: coefficient of $p**2$ of invariant $j_2$
- $\alpha$: coefficient of $x*p$ of invariant $j_2$
- $\gamma$: coefficient of $x**2$ of invariant $j_2$

\[ ^2) \text{Courtesy E. Forest for the small DA package used here ...} \]
Modified previous example (with octupole):

In our code use like:

\[ \beta = j2.sub.beta \]
\[ \alpha = 0.5*j2.sub.twoalpha \]
\[ \gamma = j2.sub.gamma \]

we obtain (here at the end of the cell):
beta, alpha, gamma
300.080714 -1.358246 9.480224E-003

\(^2\) Courtesy E. Forest for the small DA package used here ...
This was trivial - now a (normally) hard one

The exact map:

\[ p_2 = \sin(x'_2) = -\frac{B}{R} \]
\[ x_2 = A - R(1 - \cos(x'_2)) = A - R(1 - \sqrt{1 - p_2^2}) \]
\[ A = R \cdot p_1 = R \cdot \sin(x'_1) \]
\[ B = R(1 - \cos(x'_1)) + x_1 = R(1 - \sqrt{1 - p_1^2}) + x_1 \]

A 90° bending magnet ..
How to apply Differential Algebra here ...

Start with initial coordinates in DA style:

\[ \begin{align*}
  x_1 &= (0, 1, 0, 
  p_1 &= (0, 0, 1, 
\end{align*} \]

and have:

\[ \begin{align*}
  A &= (0, 0, R, 0, \ldots) \\
  B &= (0, 1, 0, 0, R, 0, \ldots) \\
\end{align*} \]

After pushing them through the algorithm:

\[ \begin{align*}
  x_2 &= (0, 0, R, -\frac{1}{R}, 0, 0, 0\ldots) = (0, \frac{\partial x_2}{\partial x_1}, \frac{\partial x_2}{\partial p_1}, \frac{\partial^2 x_2}{\partial x_1^2}, \frac{\partial^2 x_2}{\partial x_1 \partial p_1}, \ldots) \\
  p_2 &= (0, -\frac{1}{R}, 0, 0, 0, -1, 0\ldots) = (0, \frac{\partial p_2}{\partial x_1}, \frac{\partial p_2}{\partial p_1}, \frac{\partial^2 p_2}{\partial x_1^2}, \frac{\partial^2 p_2}{\partial x_1 \partial p_1}, \ldots) \\
\end{align*} \]

Automatically evaluates all non-linearities to any desired order ..
How to apply Differential Algebra here ...

Start with initial coordinates in DA style:

\[ x_1 = (0, 1, 0, ...) \]
\[ p_1 = (0, 0, 1, ...) \]

and have:

\[ A = (0, 0, R, 0, ...) \]
\[ B = (0, 1, 0, 0, 0, R, 0, ...) \]

After pushing them through the algorithm:

\[ x_2 = (0, 0, R, -\frac{1}{R}, 0, 0, ...) = (0, \frac{\partial x_2}{\partial x_1}, \frac{\partial x_2}{\partial p_1}, \frac{\partial^2 x_2}{\partial x_1^2}, \frac{\partial^2 x_2}{\partial x_1 \partial p_1}, ...) \]
\[ p_2 = (0, -\frac{1}{R}, 0, 0, 0, -1, 0, ...) = (0, \frac{\partial p_2}{\partial x_1}, \frac{\partial p_2}{\partial p_1}, \frac{\partial^2 p_2}{\partial x_1^2}, \frac{\partial^2 p_2}{\partial x_1 \partial p_1}, ...) \]

Automatically evaluates all non-linearities to any desired order ..
Some we know ...

Transfer matrix of a dipole:

\[
M_{\text{dipole}} = \begin{pmatrix}
\cos\left(\frac{L}{R}\right) & R\sin\left(\frac{L}{R}\right) \\
-\frac{1}{R}\sin\left(\frac{L}{R}\right) & \cos\left(\frac{L}{R}\right)
\end{pmatrix} = \begin{pmatrix}
\frac{\partial x_2}{\partial x_1} & \frac{\partial x_2}{\partial p_1} \\
\frac{\partial p_2}{\partial x_1} & \frac{\partial p_2}{\partial p_1}
\end{pmatrix}
\]

For a 90° bending angle we get:

\[
M_{\text{dipole}} = \begin{pmatrix}
0 & R \\
-\frac{1}{R} & 0
\end{pmatrix}
\]

as computed, but we also have all derivatives and non-linear effects!
What is the use of that:

Although not strictly an analytic method in the traditional sense:

- TPSA provide analytic expression (Taylor series) for the one turn map
- 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
Is there a summary?

\[ m = z \]

\[ \text{NORMAL} = m \]

- Get the map \( m \) somehow (no matter how)
- Analyse this map (Normal form)
And another summary

- Perturbation treatment limited to:
  - Small perturbations (not in real machines)
  - Pedagogical purpose

- For realistic machines symplectic, iterative mapping is appropriate, using:
  - Symplectic integration
  - Lie transformations and normal form analysis
  - Differential algebra
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 $h$:

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

Constant $h$ defines the trajectory in phase space!
Where to put the elements in an accelerator?

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

- 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
\( \mathcal{C}_2 = \mathcal{A}_2 \circ \mathcal{B}_2 \) is \( \mathcal{C}_2 = [R^C, T^C] \) 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:

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

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:

\[\det \mathcal{J} = 1 \text{ for all } x_0 \text{ and all } x'_0\]
Symplecticity for higher order MAPS

This is only possible for the conditions:

\[
\begin{pmatrix}
R_{11}R_{22} - R_{12}R_{21} = 1 \\
R_{11}T_{212} + 2R_{22}T_{111} - 2R_{12}T_{211} - R_{21}T_{112} = 0 \\
2R_{11}T_{222} + R_{22}T_{112} - R_{12}T_{212} - 2R_{21}T_{122} = 0
\end{pmatrix}
\]

- 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. $\frac{\partial H}{\partial q_i} = 0 \rightarrow \frac{dp_i}{dt} = 0$) → 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:

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

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:

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

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:

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

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 $n \neq m$):

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

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

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

\[ e^{ax^n p^n}: x = x \cdot e^{-anx^{n-1}p^{n-1}} \]

\[ e^{ax^n p^n}: p = p \cdot e^{anx^{n-1}p^{n-1}} \]
Collision scheme - two IPs

\[ \mu = \mu_1 + \mu_2 \]
Two IPs

→ two transfers $f_1^2, f_2^2$ and two beam-beam kicks $F_1^1, F_2^2$, first IP at $\mu_1$, second IP at $\mu$:

\[
e^{f_2^1}e^{F_1^1}e^{f_2^2}e^{F_2^2} = e^{h_2^2} = e^{f_2^1}e^{F_1^1}e^{-f_2^1}e^{f_2^1}e^{f_2^2}e^{F_2^2} = e^{h_2^2} = e^{f_2^1}e^{F_1^1}e^{-f_2^1}e^{f_2}e^{F_2^2}e^{-f_2}e^{f_2} = e^{h_2^2} = e^{e^{-f_2^1}F_1^1}e^{e^{-f_2^2}F_2^2}e^{f_2} = e^{h_2^2}.
\]

\[f_2 = -\mu A, \quad f_2^1 = -\mu_1 A, \quad \text{and} \quad f_2^2 = -\mu_2 A \]
here a miracle occurs (remember \( g(\cdot f_2 \cdot)e^{in\Psi} = g(in\mu)e^{in\Psi} \)):

\[
e^{f_2^1}e^{in\Psi} = e^{in\mu_1}e^{in\Psi} = e^{in(\mu_1+\Psi)}
\]

i.e. the Lie transforms of the perturbations are phase shifted\(^2\). Therefore:

\[
e^{e^{\cdot f_2^1}}F^1 : e^{e^{-\cdot f_2^1}F^1} : e^{\cdot f_2} = e^{\cdot h_2}:
\]

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

\[
e^{F^1(\Psi_1)} : e^{F(\Psi)} : e^{\cdot f_2} : \Rightarrow e^{F^1(\Psi_1)+F(\Psi)} : e^{\cdot f_2}:
\]

Two IPs

gives for $h_2$:

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

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

Nota bene, because of:

$$e^{iF(\Psi)} e^{iF_2} \rightarrow e^{iF_1(\Psi_1)+F(\Psi)} e^{iF_2}$$

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 $L$:

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

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

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

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

$$q(s) = e^{-LH}$$
Lie transformations on moments:

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

$$e^f: x_0 = x_1$$

$$e^f: p_0 = p_1$$

or using $$Z = (x, p_x, y, p_y, ...)$$:

$$e^f: Z_0 = Z_1$$

- 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:

$$e^{\cdot f} x^2 = (e^{\cdot f} x)^2$$  \quad (see lecture 5)

dependence:

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

$$(e^{\cdot f} x)^2 = m_{11}^2 x^2 + 2 m_{11} m_{12} x p + m_{12}^2 p^2$$
More on moments

To summarize the moments:

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

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?

Collision scheme - two IPs

IP1

IP5
Start with single IP

"Classic" (B.C.) approach:

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

\[ \mathcal{H} = \ldots + \delta(s) \epsilon V \]

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

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

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 \( f(x) \):

\[
f(x) = \frac{2}{x} \left( 1 - e^{-\frac{x^2}{2\sigma^2}} \right)
\]

as usual go to action angle variables \( \Psi, A \):

\[
x = \sqrt{2A\beta}\sin\Psi, \quad p = \sqrt{\frac{2A}{\beta}}\cos\Psi
\]

and write \( F(x) \) as Fourier series:

\[
F(x) = \sum_{n=-\infty}^{\infty} c_n(A)e^{in\Psi} \quad \text{with} \quad c_n(A) = \frac{1}{2\pi} \int_{0}^{2\pi} e^{-in\Psi} F(x) d\Psi
\]
We need:

REMEMBER: with this transform:

\[ f_2 = -\mu A \]

and useful properties of Lie operators (any textbook\(^2\)):

\[ :f_2: g(A) = 0, \quad :f_2: e^{i\mu} = i\mu e^{i\mu}, \quad g(:f_2:)e^{i\mu} = g(i\mu)e^{i\mu} \]

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

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

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:

\[
h = -\mu A + c_0(A) = \text{const.}
\]

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

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

On resonance:

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

with $c_n \neq 0$:

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

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 $I = \frac{\beta^*}{2\sigma^2}(x^2 + p_x^2\beta^*)$
→ and phase $\Psi = \arctan(\frac{p_x}{x})$
→ compare $I$ with $\hbar$
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:

\[
\frac{I}{h} \quad \mu/2\pi = 0.33
\]

- Behaviour near a resonance: 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:

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

The trick is to observe the motion every 3 turns:

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

We get a factor:

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

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

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

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