Numerical and Computational Tools in Accelerator Physics

An introduction

Werner Herr
CERN, AB Department

Why do we need computations and simulations?

- To explore new fields
- To answer scientific or technical questions
- To make design choices
- To go from a theoretical to ”real” machine
- To deal with the ”far from ideal” world

(B. Holzer)
Why do we need computations and simulations?

- Why especially important nowadays?
  - Larger equipment, more people, more money
  - Many more applications
  - Safety issues
  - Complex control and operation

- Eventually have to get the details right
Where are computational tools needed?

- Studies of Beam Dynamics
  - Design and simulation of an accelerator
  - Control and operation

- Design of accelerator equipment
  - Magnets, RF cavities ...
  - Vacuum components, cryogenics
Steps of accelerator design

1. Define basic parameters
2. Design machine layout and optics
3. Analysis of local and global properties
   - Evaluation of performance
4. Stability of beams
5. Geometry and construction
**Accelerator physics programs needed**

- Initial parameter calculation
- Optics and lattice design program
- Geometry
- Single- and multi-particle dynamics modelling
- Analyse and understand the behaviour

> Probably several programs needed

> Probably written by other people (i.e. not you)
Working with accelerator physics programs

Computer programs are only as good as the underlying concepts (and the way they are used):

- May only work on certain class of idealized problems
- May be optimized for special purpose
- If something cannot be computed, does not mean that it does not exist

Plenty of possible traps!
Working with accelerator physics programs

Do we need to understand what they are doing (algorithms, technicalities, basic concepts)?

Using any program you must ask yourself:

▷ Is the problem well defined?
▷ Is the description (model) adequate?
▷ What are the effects of approximations, etc.?
▷ Do I use the appropriate program?

To get the right conclusions → Yes!
Objectives of these lectures:

- Introduction to selected methods and programs
- Make you aware of possible problems and limitations to help understand the results correctly
- Get you acquainted with the terminology and to speak the same language
- Introduce some modern concepts and techniques
- Restrict myself to circular machines
Steps of accelerator design

1. Define basic parameters
2. Design machine layout and optics
3. Analysis of local and global properties
   Evaluation of performance
4. Stability of beams
5. Geometry and construction
Accelerator design with an optics program

- It needs: Description of machine in standard format
- It does: Optics calculations
  - Linear and non-linear optics computations
  - Parameter matching
  - Linear corrections (orbit, coupling, $\beta$, ..)
  - Non-linear and chromatic corrections
How is an accelerator described to a computer?

Not like:

\[ \frac{d^2 x}{ds^2} + K(s) \, x = 0 \]
How is an accelerator described to a computer?

The challenge:

→ Describe a machine with several thousand elements
→ Describe a complicated structure
How is an accelerator described to a computer?

The main questions:

- What kind of elements (magnets, etc.) are in the accelerator?
- Where are these elements in the accelerator?
- How do the elements act on a particle?

Other issues:

- Simplicity, speed, accuracy ...
- Description usually serves multiple purpose: optics calculation ($\beta$, ..), simulations (tracking), civil engineering
Where to put the elements in an accelerator?

- Take a simple structure:
- How to describe the position of the elements?
Where to put the elements in an accelerator?

Cartesian coordinates in 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
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"
- Specify coordinates at centre of the element ...
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"
- Specify coordinates at entry of the element ...
How does an accelerator look like to a computer?

Bending and focusing is (in general) not a continuous function of $s$.

A (finite) sequence of machine elements $M$ at longitudinal positions $s_1, s_2, s_3, \ldots$:
Are there problems with this description?

⚠ Path length $s$ for positioning may have problems:

- Elements shared by several beams (colliders, switching magnets in beam lines ...)
- Same element at different ”path length” (microtron etc.)
How does an element look like to a computer?

- You need to describe what happens to the particle in $M$.
- Assume each element $M$ (e.g. magnet) acts on the beam locally in a deterministic way, functionally independent of other elements.

In general: $\vec{z}_2 \neq \vec{z}_1$
What is M? It can represent:

- Single machine elements:
  - magnet: dipole, quadrupole
  - RF cavity

- Single machine elements (not only magnets):
  - collimators, targets, obstacles
  - vacuum chamber
  - drift
  - ......
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 $\mathcal{M}$.

$\mathcal{M}$ describes the element in terms of this quantity.

In general: $\vec{z}_2 = \mathcal{M} \circ \vec{z}_1$

$\mathcal{M}$ 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).
4 coordinates needed for 2 transverse dimensions

Describe the deviations from the reference path

Coordinate vector:
\[
\vec{z} = (x, \ px, \ y, \ py) \quad \text{or:}
\]
\[
\vec{z} = (x, \ x' = \frac{\delta x}{\delta s}, \ y, \ y' = \frac{\delta y}{\delta s}) \quad \text{(see B. Holzer)}
\]

Coordinate vector follows the reference path along \( s \).
Coordinates used
4 coordinates needed for 2 transverse dimensions

Coordinate vector:
\[
\vec{z} = (x, px, y, py) \quad \text{or:}
\]
\[
\vec{z} = (x, x' = \frac{\delta x}{\delta s}, y, y' = \frac{\delta y}{\delta s})
\]

Another trap: how are the variables defined, which variables are used?

- \(p_x\) and \(p_y\) are canonical variables, are \(x'\) and \(y'\) ?
- \(x' = \frac{p_x}{p_z}\) or \(x' = \frac{p_x}{p_0}\) ?
- What if the beam has a momentum spread
\[
\Delta(p_z)/p_z \neq 0
\]
MAPS transform coordinates through an element

- We use coordinate vector: \( \vec{z} = (x, x' = \frac{\delta x}{\delta s}, y, y' = \frac{\delta y}{\delta s}) \)

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

\[
\vec{z}_2(s_2) = \begin{pmatrix} x \\ x' \\ y \\ y' \end{pmatrix}_{s_2} = \mathcal{M} \circ \begin{pmatrix} x \\ x' \\ y \\ y' \end{pmatrix}_{s_1} = \mathcal{M} \circ \vec{z}_1(s_1)
\]
... or OPTICAL functions

6 optical functions used for 2 dimensions:

\[ \mathbf{v}_2(s_2) = \begin{pmatrix} \beta_x \\ \alpha_x \\ \gamma_x \\ \beta_y \\ \alpha_y \\ \gamma_y \end{pmatrix}_{s_2} = \mathcal{M} \circ \begin{pmatrix} \beta_x \\ \alpha_x \\ \gamma_x \\ \beta_y \\ \alpha_y \\ \gamma_y \end{pmatrix}_{s_1} = \mathcal{M} \circ \mathbf{v}_1(s_1) \]
How does $\mathcal{M}$ look like?

The map $\mathcal{M}$ describes local properties of a machine element and can be:

- Any ”description” to go from $\vec{z}_1$ to $\vec{z}_2$
- Must be derived from physics, i.e. equation of motion
- Sounds maybe complicated, but we should try to develop tools that have obvious extensions to more complicated situations
- Is this description unique for a given element?
Is $\mathcal{M}$ unique for a given element?

\[
\begin{pmatrix}
\cos(kL) & \frac{\sin(kL)}{k} & 0 & 0 \\
-k\sin(kL) & \cos(kL) & 0 & 0 \\
0 & 0 & \cosh(kL) & \frac{\sinh(kL)}{k} \\
0 & 0 & k\sinh(kL) & \cosh(kL)
\end{pmatrix}
\]

\[
\vec{z}_2 = e^{-L\frac{k}{2}(x^2-y^2)+\frac{1}{2}(x'^2+y'^2)} \vec{z}_1
\]

Absolutely not: the representation depends on the purpose, can be very different in different programs
What can $M$ be?

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!

(i.e. Transverse Dynamics, B. Holzer)
Simple examples (linear, one dimensional)
(Matrix formulation for linear* elements)

\[
\begin{pmatrix}
  x \\
  x'
\end{pmatrix}_{s_2} =
\begin{pmatrix}
  m_{11} & m_{12} \\
  m_{21} & m_{22}
\end{pmatrix}
\circ
\begin{pmatrix}
  x \\
  x'
\end{pmatrix}_{s_1}
\]

\[
\begin{pmatrix}
  \beta \\
  \alpha \\
  \gamma
\end{pmatrix}_{s_2} =
\begin{pmatrix}
  m_{11}^2 & -2m_{11}m_{12} & m_{12}^2 \\
  -m_{11}m_{21} & m_{11}m_{22} + m_{12}m_{21} & -m_{12}m_{22} \\
  m_{21}^2 & -2m_{21}m_{22} & m_{22}^2
\end{pmatrix}
\circ
\begin{pmatrix}
  \beta \\
  \alpha \\
  \gamma
\end{pmatrix}_{s_1}
\]

→ The maps become so-called ”transport matrices”

* The changes depend on \( x \) or \( x' \) only
(Interlude: Σ-matrix)

The transformation of the optical functions can also be written using the Σ-matrix formalism:

\[ \Sigma_{s_2} = M \circ \Sigma_{s_1} \circ M^T \]

i.e. for example in the linear case:

\[
\begin{pmatrix}
\beta & -\alpha \\
-\alpha & \gamma \\
\end{pmatrix}
= \begin{pmatrix}
m_{11} & m_{12} \\
m_{21} & m_{22} \\
\end{pmatrix}
\circ \begin{pmatrix}
\beta & -\alpha \\
-\alpha & \gamma \\
\end{pmatrix}
\circ \begin{pmatrix}
m_{11} & m_{21} \\
m_{12} & m_{22} \\
\end{pmatrix}
\]

→ Allows formal extension to higher order effects (e.g. synchrotron radiation)

→ Prove that it is equivalent to previous formula
Transformation of coordinates (one dimension)

Drift space of length \( L = s_2 - s_1 \):

\[
\begin{pmatrix}
  x \\
  x'
\end{pmatrix}
_{s_2}
= \begin{pmatrix}
  x \\
  x'
\end{pmatrix}
_{s_1}
+ \begin{pmatrix}
  x' \cdot L \\
  0
\end{pmatrix}
_{s_1}
= \begin{pmatrix}
  1 & L \\
  0 & 1
\end{pmatrix}
\circ
\begin{pmatrix}
  x \\
  x'
\end{pmatrix}
_{s_1}
\]
Transformation of beam ellipse

Drift space of length \( L = s_2 - s_1 \):

\[
\begin{bmatrix}
\beta \\
\alpha \\
\gamma
\end{bmatrix}_{s_2} = \begin{pmatrix}
1 & -2L & L^2 \\
0 & 1 & -L \\
0 & 0 & 1
\end{pmatrix} \circ \begin{bmatrix}
\beta \\
\alpha \\
\gamma
\end{bmatrix}_{s_1} = \begin{pmatrix}
\beta_0 - 2L\alpha_0 + L^2\gamma_0 \\
\alpha_0 - L\gamma_0 \\
\gamma_0
\end{pmatrix}_{s_2}
\]
Simple examples (one dimensional)

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

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

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

$$
\begin{bmatrix}
    x \\
    x'
\end{bmatrix}_{s_2} =
\begin{pmatrix}
    1 & 0 \\
    K^2 \cdot L (= -\frac{1}{f}) & 1
\end{pmatrix}
\circ
\begin{bmatrix}
    x \\
    x'
\end{bmatrix}_{s_1}
$$
Initial steps for optics calculation

- The optics program reads the sequence of elements of a machine (their order, their positions ..)

- It reads properties of the elements, i.e. type (dipole, quadrupole, drift ...)

- It reads strength of the elements

- It sets up the maps (matrices)

→ A ”standard” for the input language exists, plus converters (do not forget this issue !)
Simplest machine description (MADX format)

// description of elements and their strengths
// dipoles and quadrupoles only ...
mb: dipole, l=6.0, angle=0.03570;
qf: quadrupole, l=3.0, k1= 0.013426;
qd: quadrupole, l=3.0, k1=-0.013426;

// centre position of elements in the ring
start: at=0;
qf.1: qf, at=1.5000e+00;
mb: mb, at=9.0000e+00;
mb: mb, at=1.90000e+01;
qd.1: qd, at=2.6500e+01;
mb: mb, at=3.4000e+01;
mb: mb, at=4.4000e+01;
qf.2: qf, at=5.1500e+01;
...
end: at=2.20000e+03;
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):

$$\left(\begin{array}{c} x \\ x' \end{array}\right)_{s_0 + C} = M_1 \circ M_2 \circ \ldots \circ M_N \circ \left(\begin{array}{c} x \\ x' \end{array}\right)_{s_0}$$

$$\implies \left(\begin{array}{c} x \\ x' \end{array}\right)_{s_0 + C} = M_{ring}(s_0) \circ \left(\begin{array}{c} x \\ x' \end{array}\right)_{s_0}$$
Combining maps together

\[ M_{\text{ring}}(s_0) = M_1 \circ M_2 \circ \ldots \circ M_N \]

- Need to combine the N maps together
- The results is a map for one turn
- How to combine maps?
- Start with the simplest example:
Composition of elements (FODO cell) (here: simple matrix multiplications)

\[
\mathcal{M}_{cell} = \begin{pmatrix}
1 & 0 \\
-\frac{1}{f} & 1
\end{pmatrix} \circ \begin{pmatrix}
1 & L/2 \\
0 & 1
\end{pmatrix} \circ \begin{pmatrix}
1 & 0 \\
\frac{1}{f} & 1
\end{pmatrix} \circ \begin{pmatrix}
1 & L/2 \\
0 & 1
\end{pmatrix}
\]
Composition of elements (FODO cell)  
(here: simple matrix multiplications)

\[
M_{cell} = \begin{pmatrix}
1 + \frac{L}{2f} & L + \frac{L^2}{4f}
\end{pmatrix}
\begin{pmatrix}
\frac{L}{2f} & \frac{L^2}{4f^2}
\end{pmatrix}
\begin{pmatrix}
1 & 1 - \frac{L}{2f} - \frac{L^2}{4f^2}
\end{pmatrix}
\]
Combining maps together

For our FODO ring with N cells:

\[ M_{\text{ring}}(s_0) = M_1 \circ M_2 \circ \ldots \circ M_N = M_{\text{cell}}^N \]

- Simple matrix multiplications for regular ring
- If maps are more complicated? (see later)
- What is the physical picture of \( M_{\text{ring}}(s_0) \)?
Why are we interested in $\mathcal{M}_{\text{ring}}$?

- The map $\mathcal{M}_{\text{ring}}$ is extremely important:
  - A computer does not know Hill’s equation
  - Courant-Snyder ansatz (formalism) assumes motion is linearly stable, periodic, confined, and has a closed orbit.
  - A priori we do not know that ...
  - The OTM $\mathcal{M}_{\text{ring}}$ contains all information about global behaviour in the ring, i.e. stability, tune, $\beta$, closed orbit etc.

- No need for assumptions
What else can we do with $\mathcal{M}_{ring}$?

- $\mathcal{M}_{ring}$ (or $\mathcal{M}_{part}$) allow to derive global quantities
- In the Courant-Snyder analysis of linear systems $\mathcal{M}_{ring}$ corresponds to a matrix
- ”Straightforward” to formally extend it to complicated (e.g. non-linear) problems
- Allow the analysis of imperfections (and their correction !)
- Additional tools and concepts needed (invariants, fixpoints, normal forms etc.)

Demonstrate the Courant-Snyder analysis first
(Interlude: Fixed Points)

- Some points in phase space \( \vec{z}_1 \) are repeated after \( n \) completed turns (remember resonances!)

\[
\mathcal{M}^n_{\text{ring}} \circ \vec{z}_1 = \vec{z}_2 \equiv \vec{z}_1
\]

- Defines a Fixed Point of order \( n \)

- Fixed Point of order 1 is the closed orbit

- Stability requires existence of such a fixed point

- Closed orbit is found (or not!) in optics programs by searching for the first order fixed point (i.e. \( \vec{z}_1 \) at \( s_0 \))
Analysis of the One-Turn-Map (Matrix)

- Start simple: all maps are matrices (i.e. only linear elements)
- Usually the case for initial design

→ The One-Turn-Map is a One-Turn-\textbf{Matrix}:

\[
\begin{pmatrix}
 x \\
 x'
\end{pmatrix}
_{\text{s}_0 + C} =
\begin{pmatrix}
 m_{11} & m_{12} \\
 m_{21} & m_{22}
\end{pmatrix}
\circ
\begin{pmatrix}
 x \\
 x'
\end{pmatrix}
_{\text{s}_0}
\]

→ After all multiplications we get the One-Turn-\textbf{M}atrix which depends on the starting point \text{s}_0.
Find the tune $Q$

We can find the tune $Q$ from the One-Turn-Matrix $\mathcal{M}_{ring}(s_0)$ by computing the eigenvalues of $\mathcal{M}_{ring}(s_0)$:

$$\det(\mathcal{M}_{ring}(s_0) - \lambda) = 0$$

gives

$$\lambda = \cos(2\pi Q) \pm i \cdot \sin(2\pi Q)$$

(verify with the One-Turn-Matrix you know from previous lecture !)
Analysis of the One-Turn-Matrix

What else can we do with the One-Turn-Matrix?

We can express the One-Turn-Matrix $M_{ring}(s_0)$ in terms of Courant-Snyder parameters:

We know that $M_{ring}(s_0)$ for one dimension must be:

$$
M_{ring}(s_0) \equiv \begin{pmatrix}
\cos \mu + \alpha(s_0) \sin \mu & \beta(s_0) \sin \mu \\
-\gamma(s_0) \sin \mu & \cos \mu - \alpha(s_0) \sin \mu
\end{pmatrix}
$$

and we also know that (for a ring):

$$
\alpha(s_0 + C) \equiv \alpha(s_0), \quad \beta(s_0 + C) \equiv \beta(s_0), \quad \gamma(s_0 + C) \equiv \gamma(s_0)
$$
Get optical functions

Comparison of:

\[
\begin{pmatrix}
m_{11} & m_{12} \\
m_{21} & m_{22}
\end{pmatrix} = M_1 \circ M_2 \circ \ldots \circ M_N
\]

and:

\[
M_{\text{ring}}(s_0) = \begin{pmatrix}
\cos \mu + \alpha(s_0) \sin \mu & \beta(s_0) \sin \mu \\
-\gamma(s_0) \sin \mu & \cos \mu - \alpha(s_0) \sin \mu
\end{pmatrix}
\]

gives optical functions at position \( s_0 \):

\[\rightarrow \beta(s_0), \ \alpha(s_0), \ \gamma(s_0) \text{ (depend on position } s_0)\]

\[\rightarrow \mu \text{ is independent of } s_0: \ (2\pi Q)\]
We have now:

- Values for $\beta_x$, $\beta_y$, $\alpha_x$, ... etc. at the position $s_0$
- Tunes for both planes, closed orbit

The next step:

- Starting from initial optical (Twiss) functions at $s_0$, transforming $\beta_x$, $\beta_y$, $\alpha_x$, ... through the lattice gives functions at all positions $s$.

Question: what are the $\beta$-functions etc. of a linear accelerator or a beam line ???
Get the optical functions around the ring

\[
\begin{pmatrix}
\beta \\
\alpha \\
\gamma
\end{pmatrix}_s = \begin{pmatrix}
m_{11}^2 & -2m_{11}m_{12} & m_{12}^2 \\
-m_{11}m_{21} & m_{11}m_{22} + m_{12}m_{21} & -m_{12}m_{22} \\
m_{21}^2 & -2m_{21}m_{22} & m_{22}^2
\end{pmatrix} \circ \begin{pmatrix}
\beta \\
\alpha \\
\gamma
\end{pmatrix}_{s_0}
\]

Successive application of matrices give Twiss functions at each element around the ring and at each position $s \rightarrow$
Optical functions (horizontal $\beta$):

Horizontal beta function

$\beta (m)$

$s (m)$

Momentum offset = 0.00 %
Optical functions (vertical $\beta$):
Extension to two dimensions

Can be written as separate equations:

\[
\begin{pmatrix}
    x \\
x'
\end{pmatrix}_{s_2} =\begin{pmatrix}
    m_{11} & m_{12} \\
m_{21} & m_{22}
\end{pmatrix} \circ \begin{pmatrix}
    x \\
x'
\end{pmatrix}_{s_1}
\]

\[
\begin{pmatrix}
    y \\
y'
\end{pmatrix}_{s_2} = \begin{pmatrix}
    m_{33} & m_{34} \\
m_{43} & m_{44}
\end{pmatrix} \circ \begin{pmatrix}
    y \\
y'
\end{pmatrix}_{s_1}
\]
Extension to two dimensions

- Extend vectors for coordinates or optical parameters
- Extend transfer maps/matrices

$$\begin{pmatrix} x \\ x' \\ y \\ y' \end{pmatrix}_{s_2} = \begin{pmatrix} m_{11} & m_{12} & 0 & 0 \\ m_{21} & m_{22} & 0 & 0 \\ 0 & 0 & m_{33} & m_{34} \\ 0 & 0 & m_{43} & m_{44} \end{pmatrix} \circ \begin{pmatrix} x \\ x' \\ y \\ y' \end{pmatrix}_{s_1}$$
**Extension to two dimensions (coupling)**

The horizontal and vertical motion can be coupled:

*→ Additional elements in matrix*

\[
\begin{pmatrix}
   x \\
   x' \\
   y \\
   y'
\end{pmatrix}_{s_2} =
\begin{pmatrix}
   m_{11} & m_{12} & m_{13} & m_{14} \\
   m_{21} & m_{22} & m_{23} & m_{24} \\
   m_{31} & m_{32} & m_{33} & m_{34} \\
   m_{41} & m_{42} & m_{43} & m_{44}
\end{pmatrix}
\circ
\begin{pmatrix}
   x \\
   x' \\
   y \\
   y'
\end{pmatrix}_{s_1}
\]

*Q: what about the $\beta$-functions now??
Going to three dimensions

Formally extended by adding two new variables:

\[ (x, x', y, y', \Delta s, \frac{\Delta p}{p}) \]

\[ \Delta s = c\Delta t: \text{ longitudinal displacement with respect to} \]

reference particle

\[ \frac{\Delta p}{p}: \text{ relative momentum difference with respect to} \]

reference particle
Going to three dimensions

With \((x, x', y, y', \Delta s, \frac{\Delta p}{p})\)

\[
\begin{pmatrix}
  x \\
  x' \\
  y \\
  y' \\
  c\Delta t \\
  \frac{\Delta p}{p}
\end{pmatrix}
\begin{pmatrix}
  m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} \\
  m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} \\
  m_{32} & m_{32} & m_{33} & m_{34} & m_{35} & m_{36} \\
  m_{42} & m_{42} & m_{43} & m_{44} & m_{45} & m_{46} \\
  m_{52} & m_{52} & m_{53} & m_{54} & m_{55} & m_{56} \\
  m_{62} & m_{62} & m_{63} & m_{64} & m_{65} & m_{66}
\end{pmatrix}
\begin{pmatrix}
  x \\
  x' \\
  y \\
  y' \\
  c\Delta t \\
  \frac{\Delta p}{p}
\end{pmatrix}
\]
Going to three dimensions

Formally extended by adding two new 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
Off momentum effects

Correct Hamiltonian for a magnetic element:

\[ \mathcal{H} = -(1 + \frac{x}{\rho})\sqrt{(1 + \delta)^2 - x'^2 - y'^2} + V(x, y) \]

- Non-linear terms due to kinematics: \((1 + \delta)^2\),
  (even with quadrupoles only !)

- After expansion and keep second order (e.g. large machines):

\[ \mathcal{H} = \frac{x'^2 + y'^2}{2(1 + \delta)} + V(x, y) \]
Off momentum effects

Strengths $k$ of elements modified by non-zero $\delta = \frac{\Delta p}{p}$:

$$k \implies k/(1 + \frac{\Delta p}{p}) = k/(1 + \delta)$$

- Closed orbit and tune are usually different for non-zero $\frac{\Delta p}{p}$

- Dispersion

- Chromatic effects

- Beware of small rings or large $\delta$ ....
Matching optical functions

- Modify machine optics to get desired properties around the machine or in specific places.
- For example, you may want special conditions for equipment: RF, collimators, diagnostics.
- For experiments: in colliding beam machines.
- Algorithms to adjust parameters and layout.

→ This process is called MATCHING!
→ Available in most optics programs (for lines and circular machines).
General purpose optics programs

Always allow to:

- Compute optical parameters (Twiss functions)
- Match the required properties

Often allow to:

- Simulate machine imperfections
- Correct imperfections
Popular Optics Programs

- BeamOptics (based on Mathematica)
- TURTLE (Beam lines)
- WINAGILE (WINDOWS, interactive, originally for teaching)
- TRANSPORT (General purpose, third-order matrix)
- DIMAD (Second-order matrix, tracking)
- TEAPOT (General purpose, thin element approximation)
Popular Optics Programs (cont.)

- COSY (Multi purpose, high order maps, differential algebra)
- SYNCH (General purpose)
- MAD (versions: 8,9,X) (Multi purpose optics and tracking)
- SAD (Multi purpose optics and tracking)
- MARYLIE (Lie algebra, tracking)
- PTC (MAP based, object oriented)
- MADX-PTC (combined MADX-PTC)
Which Optics Program should I use?

Very application dependent, you have:

- Beam line
- Large ring
- Small ring
- Large momentum offset or changing momentum (e.g. FFAG, acceleration)
- Linear accelerator
- Unconventional geometry
- Collider (one or more rings/lines)
Which Optics Program should I use?

Very application dependent, you want to do:

- Design linear optics
- Linear optical matching
- Introduce and correct imperfections
- Non-linear optical matching
- Particle tracking
- Evaluate the dynamic aperture
- Study collective effects
(Interlude: Course on optics design)

Intermediate level CAS 2003, 2005, 2007 (and maybe 2009) offers a course on optics design

- Purpose is to develop a realistic accelerator optics
- Includes correction elements, optical matching, dispersion suppressors ... 
- MAD is used for practical implementation

The course is available on CD-ROM (on request) or from the web
(Reminder: 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 \textit{linearly} on amplitude (re-written from the matrix form):

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

\[ \implies \tilde{z}(s_2) = M \cdot \tilde{z}(s_1) \]

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

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

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

\[\tilde{z}(s_2) = \mathcal{M} \circ \tilde{z}(s_1)\]

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

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

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

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

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

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

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

Higher orders can be defined as needed ...

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

Assume now 2 maps of second order:
\[ \mathcal{A}_2 = [R^A, T^A] \quad \text{and} \quad \mathcal{B}_2 = [R^B, T^B] \]

the combined second order map
\[ \mathcal{C}_2 = \mathcal{A}_2 \circ \mathcal{B}_2 \quad \text{is} \quad \mathcal{C}_2 = [R^C, T^C] \quad \text{with:} \]
\[ R^C = R^A \cdot R^B \]

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

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

$$\mathcal{J}_{ik} = \frac{\delta z^i_2}{\delta z^k_1}$$

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

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

try truncated Taylor map in 2D, second order:

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

The Jacobian becomes:

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

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:

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

- 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
Using higher order MAPS

- Needed to study long term stability and derive dynamics quantities

- Basic questions:
  - Is it sufficiently high order (i.e. accurate) ?
  - Is it sufficiently symplectic ?
  - How do we generate a high order map efficiently (accurate and symplectic) ?

- Many programs and algorithms available, not all are always appropriate
Various types of higher order MAPS

Choice depends on the application

- Taylor maps (large number of coefficients)
- Lie transformations (minimum number of coefficients)
- Truncated power series algebra (TPSA), can also generate Taylor map from tracking
- Symplectic integration techniques (see later)
- ...

⚠️ The catch: which is the truncation order ??
Single particle dynamics in a nutshell

- Try to compute a (one turn) MAP
- It contains everything
- Its analysis will tell you what you need to know
- It doesn’t matter how you got it (analytically, tracking ..)!
- Benchmark against simulation, i.e. element by element tracking, to ensure that approximations do not affect physics
You can derive:

- Tunes (we already did)
- Betatron functions (we already did)
- Stability borders, dynamic aperture (within limits)
- Detuning with amplitudes
- Invariants of the motion
- Fixpoints, closed orbit, resonance strength
- More difficult: long term stability of the beam!
Steps of accelerator design

- Define basic parameters
- Design machine layout and optics
  - Analysis of local and global properties
    - Evaluation of performance
  - Stability of beams
    - Geometry and construction
Simulation of an accelerator

- Purpose is to imitate the behaviour of a particle or a beam in an accelerator
- Use local properties of the machine element to describe its interaction with a particle
- The aim is to derive the global behaviour
  - Long term stability
  - Lifetime
  - ....
- An exact simulation code is the most reliable ”map”
Evaluation by simulation (1)

- Single particle behaviour
- Usually concerns long term behaviour such as beam loss, effect of the accelerator components on a single particle
  - Non-linear elements
  - Machine imperfections (e.g. field, alignment errors)
  - External distortions (e.g. scattering)
Evaluation by simulation (2)

- Multi particle behaviour

- Usually concerns collective behaviour: coherent motion, emittance increase, damping etc.
  - The effect of the accelerator components on an ensemble of particles (e.g. impedances)
  - Interactions of particles between each other. These are usually dictated by the properties of the accelerator (e.g. space charge, beam-beam effects ...)
The motion of a test particle through the elements of a machine is simulated for a large number of turns → ”tracking”

Use appropriate coordinates, start with $\tilde{z}_0$.

In each element (or part of the machine), the coordinates are transformed by $\tilde{z}_{n+1} = \mathcal{M} \circ \tilde{z}_n$

$\mathcal{M}$ must be symplectic of course

We follow the evolution of the coordinates $\tilde{z}_{n+1}$, not of the maps $\mathcal{M}$

We must distinguish **thick** and **thin** elements
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 (also 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...
Tracking through thin elements

- No change of amplitudes $x$ and $y$
- The momenta $x'$ and $y'$ receive an amplitude dependent deflection (kick)

$$x' \rightarrow x' + \Delta x' \text{ and } y' \rightarrow y' + \Delta y'$$
These programs are so-called kick-codes
Kick can be non-linear
Always symplectic! (homework)
Usually rather fast on computers
Most tracking programs are of this type
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
Simple example (1D, see B. Holzer)

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

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

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

\[
\begin{pmatrix}
  x \\
  x'
\end{pmatrix}_{s_1+L} = \begin{pmatrix}
  1 & 0 \\
  -K^2 \cdot L (= -\frac{1}{f}) & 1
\end{pmatrix} \circ \begin{pmatrix}
  x \\
  x'
\end{pmatrix}_{s_1}
\]
Thick → thin quadrupole

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

- Exact map (matrix) for quadrupole
- For efficiency:
  - Can we avoid trigonometric functions?
  - What do we lose?
- (What follows is valid for all elements)
Accuracy of thin lenses

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

Start with exact map

Taylor expansion in ”small” length \( L \):

\[ L^0 \cdot \begin{pmatrix}
1 & 0 \\
0 & 1
\end{pmatrix} + L^1 \cdot \begin{pmatrix}
0 & 1 \\
-K^2 & 0
\end{pmatrix} + L^2 \cdot \begin{pmatrix}
-rac{K^2}{2} & 0 \\
0 & -rac{K^2}{2}
\end{pmatrix} + ... \]
Accuracy of thin lenses (B)

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 $O(1)$

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

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

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

- Precise to first order \( \mathcal{O}(1) \)
- ”symplectified” by adding term \(-K^2 L^2\)

(it is wrong to \( \mathcal{O}(2) \) anyway ...)
Accuracy of thin lenses

- Keep up to second order term in \( L \)

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

- Precise to second order \( O(2) \)

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

Symplectification like:

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

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

- 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:
Thick quadrupole ..

\[ K^2 \]
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)

\[
\begin{pmatrix}
1 & \frac{1}{2}L \\
0 & 1 \\
\end{pmatrix}
\begin{pmatrix}
1 & 0 \\
-K^2 \cdot L & 1 \\
\end{pmatrix}
\begin{pmatrix}
1 & \frac{1}{2}L \\
0 & 1 \\
\end{pmatrix}
= 
\begin{pmatrix}
1 - \frac{1}{2}K^2L^2 & L - \frac{1}{4}K^2L^3 \\
-K^2 \cdot L & 1 - \frac{1}{2}K^2L^2 \\
\end{pmatrix}
\]
What is the point ???

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

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

Symplecticity: but phase space ellipse not accurate
Quadrupole symplectic $O(2)$ solution

Symplecticity: phase space ellipse accurate enough
Can we do better?

Try a model with 3 kicks:

\[ \begin{align*}
    a & \approx 0.6756, \quad b \approx -0.1756, \quad c \approx -1.7027, \quad d \approx 1.3512 \\
\end{align*} \]

With:

- we have a $O(4)$ integrator ...
- (a $O(6)$ integrator would require 9 slices ...)
Can we use this for studying long term behaviour?

- Symplecticity is important
- If map is approximate:
  - Phase space slightly distorted
  - Long term stability conserved
- Needs to be verified!
Accuracy of thin lenses

What about other (e.g. non-linear) elements?

assume a general case:

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

using the thin lens approximation 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 !!
Accuracy of thin lenses

Accuracy of "leap frog" algorithm/integration"

the Taylor expansion gives:

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

the "leap frog" algorithm gives:

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

_errors_ Errors are \( \mathcal{O}(L^3) \)
Accuracy of thin lenses

For bar/coffee discussions:

why did I write:

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

and not:

\[ x'' = f(x, x') \]
Assume a (one-dimensional) sextupole with:

\[ x'' = k \cdot x^2 \]

using the thin lens approximation (type D) gives:

\[
\begin{align*}
x(L) &= x_0 + x'_0 L + \frac{1}{2}kx_0^2 L^2 + \frac{1}{2}(\frac{1}{3})kx_0x'_0 L^3 + \frac{1}{8}(\frac{1}{12})kx_0'^2 L^4 \\
x'(L) &= x'_0 + kx_0^2 L + kx_0x'_0 L^2 + \frac{1}{4}(\frac{1}{3})kx_0'^2 L^3
\end{align*}
\]

Map for thick sextupole 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
Analysis tools

- Fourier analysis, diffusion coefficients, chaos detection ...

- Phase space plots from (simple) tracking:
  - Start with a ”particle” with initial coordinates $x$ and $x'$ at a position $s_0$
  - Pass through the One-Turn-Map (for position $s_0$ !)
  - Plot the new $x$ and $x'$ coordinates at position $s_0$ after every turn
A simple example

Linear transformation plus a constant deflection (i.e. orbit kick from displaced quadrupole)

\[
\begin{pmatrix}
  x \\
  x' \\
\end{pmatrix}_{n+1} = \begin{pmatrix}
  \cos(\mu) & \sin(\mu) \\
  -\sin(\mu) & \cos(\mu) \\
\end{pmatrix}_{s_0} \cdot \begin{pmatrix}
  x \\
  x' \\
\end{pmatrix}_n + \begin{pmatrix}
  0 \\
  b \\
\end{pmatrix}
\]

\[
\mu = 2\pi Q_x = 2\pi \cdot 0.19,
\]

constant \( b \) is a free parameter

\[\rightarrow \text{Find the fixed point(s) (closed orbit)}\]
A simple example..

- Start at different amplitudes and "observe" x and x' at position $s_0$
A (still) simple example

Linear transformation plus a quadratic non-linearity (e.g. (thin) sextupole) plus a constant deflection

\[
\begin{pmatrix}
x \\
x'
\end{pmatrix}_{n+1} = \begin{pmatrix}
\cos(\mu) & \sin(\mu) \\
-\sin(\mu) & \cos(\mu)
\end{pmatrix}
\begin{pmatrix}
x \\
x'
\end{pmatrix}_n + \begin{pmatrix}
0 \\
ax^2
\end{pmatrix} + \begin{pmatrix}
0 \\
b
\end{pmatrix}
\]

\[\mu = 2\pi Q_x = 2\pi \cdot 0.22,\]

constants a and b are free parameters

→ Find the fixed points (how many do you see ?)
A (still) simple example..

- Motion at different amplitudes distorted: sextupole driving a 5th order resonance
- Stability region reduced by non-linear effect
Analysis of tracking results

- Inspection of phase space structure
- Regular motion and stability boundary
- Identification of chaotic motion
- Tracking results (algorithms, output) can be used to ”construct” analytical, approximate Taylor map for the ring: TPSA (Berz, Forest, 1989), Requires numerical differentiation → Differential Algebra (Berz, 1988)
Several optics programs can also perform single particle tracking

Many optics programs have thick to thin element converters

Specialized codes exist (SIXTRACK, PTC, ...)

Some codes have analysis tools (normal forms, chaos detection etc.)

For predicting reliably long term behaviour essential!
Complications: light particles

- Light particles ($e^-, e^+$ etc.) emit synchrotron radiation and motion is damped
  - Stochastic component
  - No symplecticity, no invariants (but equilibrium parameters, e.g. emittance)

- Synchrotron motion must be simulated
- Computation of damping properties
Simulation of multi particle effects

- Often requires the simulation of a **beam**: simulate many (up to $10^8$) particles simultaneously and study their behaviour:
  - Beam shape (density distribution)
  - Centre of mass motion of all particles

- Must be **self-consistent**: changes of the beams must be taken into account

- Fields generated by the beam need to be computed
Complications: many particles

- Particles have different amplitudes
- Particles have different tunes
- Particles have different momenta!
- Definition of emittance becomes more complicated
Strategy for multi-particle simulations (1)

- Generate and simulate many particles ($10^4 - 10^8$ per beam) simultaneously
- Every particle interacts with the machine elements individually
- The whole ensemble interacts with the machine elements
- Every particle interacts with other particles!
  - Feed back into motion of individual particles
All particles must be treated in parallel

For realistic LHC: $10^7$ to $10^8$ particles to simulate

Already for storage requires $\approx 10$ Gb memory

Parallel processing needed for reasonable computing time

Often requires (intelligent) simplifications
Simulation of interactions with environment

This means: interaction of the individual particles and the whole beam with:

- Machine elements (e.g. magnets, RF, ...)
- Wake fields
- Impedances
- Electron cloud
- Intercepting elements (e.g. collimators, ...)

→ Strategies have changed with fast computers ...
Particles inside a beam interact with other particles from the same beam:

- Space charge effects
- Intra-beam scattering
- Multi-bunch effects
Simulation of interactions with other beams

So-called beam-beam effects

- Other beam acts like a (very) non-linear lens
- Incoherent beam-beam effects (on each individual particle)
- Coherent beam-beam effects (on ensemble of particles)

→ Requires the knowledge of fields generated by the other beam
One-Turn-Maps can be written for two bunches or two beams (e.g. 1 and 2)

Here only 1 dimension for illustration

\[
\begin{pmatrix}
    x_1 \\
    x'_1 \\
    x_2 \\
    x'_2
\end{pmatrix}_{n+1} =
\begin{pmatrix}
    m_{11} & m_{12} & 0 & 0 \\
    m_{21} & m_{22} & 0 & 0 \\
    0 & 0 & m_{33} & m_{34} \\
    0 & 0 & m_{43} & m_{44}
\end{pmatrix} \cdot
\begin{pmatrix}
    x_1 \\
    x'_1 \\
    x_2 \\
    x'_2
\end{pmatrix}_n
\]
Matrix formulation (linear models)

One-Turn-Maps can be written for two bunches or two beams (e.g. 1 and 2)

Additional elements couple two beams

\[
\begin{pmatrix}
  x_1 \\
  x'_1 \\
  x_2 \\
  x'_2
\end{pmatrix}_{n+1} = \begin{pmatrix}
  m_{11} & m_{12} & m_{13} & m_{14} \\
  m_{21} & m_{22} & m_{23} & m_{24} \\
  m_{31} & m_{32} & m_{33} & m_{34} \\
  m_{41} & m_{42} & m_{43} & m_{44}
\end{pmatrix} \circ \begin{pmatrix}
  x_1 \\
  x'_1 \\
  x_2 \\
  x'_2
\end{pmatrix}_n
\]

Allows computation of eigenmodes, eigenfrequencies of multi bunch systems
Some simulations require the computation of fields (or forces) produced by a beam from Poisson equation (here 2-dimensional):

$$\Delta V = -4\pi \cdot \rho(x, y)$$

- The density of the beam is $\rho(x, y)$
- For simple distributions (Gaussian, uniform ...) can be solved analytically
- In general (i.e. in the interesting cases !) it is done numerically
Some basic methods

Particle - particle methods: compute field between each particle pair and add up (not practical for large number of particles, sometimes used in celestial mechanics)

Particle - mesh methods: distribute particles on a mesh (grid) and solve the Poisson equation for discrete points

Multipole methods: develop potentials/fields as multipole expansion

Choice depends on application and parameters
Example: Centre of mass motion as function of time

From beam-beam simulations: Bunch oscillations and frequency spectrum
Example: Beam size as function of time

From electron-cloud simulation (courtesy: E. Benedetto)
Alternatives

- Sometimes multi-particle simulations are too time consuming
- Numerical solution of the Vlasov-equation
  - E.g. finite difference methods
  - Intermediate level school (Sevilla, 2001)
Example: Beam size as function of time

Beam-beam simulation: numerical solution of Vlasov equation gives evolution of beam sizes.
Many codes exist, always specialized:
- Collective instabilities
- Beam-beam effects
- Electron cloud effects
- etc. ...

Often compact and linked to optics codes
(Personal) Comments on simulations:

- Here I gave only a selective overview of what can be done.
- Techniques and tools in dedicated schools and courses (some in Intermediate CAS Course).
- What can be done has changed a lot in the last decade.
- It is easy to write a program!
- Analysis and interpretation is usually the difficult part.
What can go wrong?

- Wrong or missing physics in the program
- Numerical problems
- Different results on different computers
- Programming bugs ...
- Biased analysis

→ Be aware of the limitations of the program
→ Make sure it is reproducible
Control and operation

Basic aim: optimize performance

As operator or accelerator physicist:
- Provide and improve model of machine
- Measure and interpret beam parameters
- Correct and control beam parameters
- Conduct machine experiments
Control and operation of an accelerator

Basic problem: measure and control beam parameters

- Control (orbit, chromaticities ..) depend on machine model which may be incomplete
- Feedback from measurements improves the model and simulations
- Should use the same strategies and methods as during design (Remember: matching !)

→ May be an iterative process
Control and operation of an accelerator

Very similar to simulation or design of a machine except:

- Interface to hardware and control (e.g. power converters)
- Beam instrumentation!
- Communication (networks, etc.)
- Issues such as: timing, alarms, interlocks,

Treated in dedicated workshops and schools
Summary: what is needed?

- Appropriate description of the accelerator
- Tools to get a One-Turn-Map to derive global quantities
- Simulation tools to study long term behaviour
  - Single particles
    (dynamic aperture, lifetime ..)
  - Multi particles
    (collective effects, emittance, coherent motion ...)

...
backup slides
backup slides

- Lie transforms (principles and examples)
- Normal forms (principles and examples)
- Truncated Power Series Algebra (TPSA) (principles and examples)
- Differential algebra (principles and examples)
Hamiltonian ..

\[ H = \frac{p_x^2 + p_y^2}{2(1 + \delta)} - k_0 x \delta + \frac{k_0^2}{2} x^2 + \frac{k_1}{2} (x^2 - y^2) + \frac{k_2}{3} (x^3 - 3xy^2) \]

- Useful formalism when invariants are investigated
- Perturbative formalism can lead to complete nonsense, if wrongly interpreted ...
- Very useful in the context of Lie transformations
Lie transformations

Introduce Poisson bracket:

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

Here the variables \(q_i, p_i\) are canonical variables, \(f\) and \(g\) are functions of \(q_i\) and \(p_i\).

Is \(\mathcal{H}\) is the Hamiltonian of a system, then:

\[
[f, \mathcal{H}] = 0
\]

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

This is classical mechanics ...
Lie transformations

We can define:

\[ f : g = [f, g] \]

where \( f : \) is an operator acting on the function \( g \).

We can define powers as:

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

in particular:

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

Lie transforms acting on functions like $x, p$ (canonical momentum, instead of $x'$):

$$p^2 \cdot x = -2p \quad \Rightarrow p^2 \cdot p = 0$$

or more complicated:

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

$$= x + Lp$$

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

$$= p$$

This is the transformation of a drift space of length $L$ !!
Lie transformations

\[ e^f : (x, p) _0 = (x, p) _1 \]

Lie transforms describe how to go from one point to another.

Homework try:

\[ e ^ {- \frac{1}{2} f x^2} : \]

or:

\[ e ^ {- \frac{1}{2} L(k^2 x^2 + p^2)} : \]
Why all that ???

congatenation very easy:

\[ e^{ih} = e^{if} e^{ig} = e^{f+g} \]

when \( f \) and \( g \) commute.

- Otherwise formalism exist

- Exponential form \( e^{ih} \) is always symplectic

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

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

when \(f_2, f_3, f_4\), are 2nd, 3rd, 4th order polynomials (factorization theorem):

\[ e^\cdot f^\cdot = e^\cdot f_2^\cdot e^\cdot f_3^\cdot e^\cdot f_4^\cdot \]

- Each term is symplectic
- Truncation at any order does not violate symplecticity

Lie transformations are the natural extension of linear to non-linear formalism!
Example: thick sextupole

The map of an element of length $L$ and the Hamiltonian $H$ is:

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

The Hamiltonian of a thick sextupole is:

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

we search for:

$$e^{-L:H}x \quad \text{and} \quad e^{-L:H}p_x \quad \text{etc.}$$

$$e^{-L:H}x = \sum_{i=0}^{\infty} \frac{1}{i!} (-L : H :)^i x$$
Example: thick sextupole

we can compute:

\[ H^i x \]

to get:

\[ H^1 x = -p_x \]
\[ H^2 x = -k(x^2 - y^2) \]
\[ H^3 x = 2k(xp_x - yp_y) \]

....

and get:

\[ e^{-L:H} x = x + pxL - \frac{1}{2}kL^2(x^2 - y^2) - \frac{1}{3}kL^3(xp_x - yp_y) + \ldots \]

(more accurate than "leap frog" thin lens approximation !)
Why all that ???

Very useful to find invariants!!

Homework ➞ what is:

a) the invariant

b) the transformation

with the effective Hamiltonian $h$ (1D):

$$e^{:\cdot h} = e^{:\cdot f_2} = e^{-\frac{\mu}{2}(\gamma x^2 + 2\alpha xp + \beta p^2)}$$
Hints

with a second order polynomial as $h$:

$$e^{\cdot h} = e^{\cdot f_2} = e^{-\frac{1}{2}(ax^2 + 2bx + cp^2)}:\n$$

the solution for the matrix form is:

$$\mathcal{R} = \cos(\sqrt{ac - b^2}) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{\sin(\sqrt{ac - b^2})}{\sqrt{ac - b^2}} \begin{pmatrix} b & c \\ -a & -b \end{pmatrix}$$

(can also be used to find $f_2$ from matrix form, a quadratic form $f_2$ always corresponds to a matrix)
Normal forms

If $\mathcal{M}$ is our usual one-turn-map:

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

We can choose $A$ to get:

$$\mathcal{N} = e^{-h}:$$

with:

$$h = (w_x + w'_x p_\tau + w''_x p_\tau^2)h_x + (w_y + w'_y p_\tau + w''_y p_\tau^2)h_y + (a \cdot h_x^2 + b \cdot h_x h_y + c \cdot h_y^2 + d \cdot p_\tau^2) + ....$$

and

$$h_x = (x^2 + p_x^2)/2, \quad h_y = (y^2 + p_y^2)/2, \quad p_\tau = (p_t - p_t^0)/(p_0 c)$$
Why is THAT interesting?

- The map $\mathcal{N}$ sends circles into circles

- The function $h$ contains:
  - $w_x, w_y$ are the tunes
  - $w'_x, w'_y$ are the chromaticities
  - $w''_x, w''_y$ are the second order chromaticities
  - $a, b, c, d$ anharmonicities, i.e. amplitude dependence of tunes
Why is THAT interesting?

- The quantities \( I_{x,y} = Ah_{x,y} \) are invariant i.e.:
  - \( \mathcal{M}I_{x,y} = I_{x,y} \)
  - They generalize Courant-Snyder invariants to the non-linear case
Truncated Power Series Algebra

What if we can somehow calculate the derivatives of the output with respect to input ??

Then we have the Taylor series, i.e. the Taylor map!

Algorithm can be formula or subroutine or program..

Formula → by hand

Subroutine or program etc. → by numerical differentiation
Numerical differentiation

The problem getting the derivative 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) (Berz, 1988)
Differential Algebra

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

2. Define operations 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:

\[
(0, 0) < (0, 1) < (r, 0)
\]
Differential Algebra

This means that (0,1) is between 0 and ANY real number \( \rightarrow \) infinitely small !!!
We call this therefore ”differential unit” \( d = (0, 1) \).
Of course \( (q, 0) \) is just the real number \( q \) and we define
”real” 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**

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

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

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:

$$f(2) = \frac{9}{2}, f'(2) = \frac{15}{4}$$
**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!
Differential Algebra

If we had started with:

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

we would get:

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

If we had started with:

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

we would get:

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

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

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

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

and get (with more complicated multiplication rules):

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

\[
f(x) = (a, 1, 0, 0, 0, \ldots)
\]

Push \(f(x)\) through the algorithm (no matter what it is), using our rules and we get all derivatives around \(a\), i.e. we get the Taylor coefficients and therefore the map!

The maps are provided with the desired accuracy and to any order.
Concatenation of Lie transformations

To concatenate two Lie-transforms, use BCH (Baker-Campbell-Hausdorff) formula (without proof):

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

\[ h = f + g + \frac{1}{2} : f \cdot g + \frac{1}{12} : f^{2} \cdot g + \frac{1}{12} : g^{2} \cdot f + \frac{1}{24} : f \circ g^{2} \cdot f + \frac{1}{720} \ldots \]

when \( g \) is not too large we can write the concatenation easier as:

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

Example: linear transfer \( f_2 \) plus local, non-linear kick \( F(x) \) (in action and angle variables \( A, \Phi \)):

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

\[
F(x) \Rightarrow \sum_{n=-\infty}^{\infty} c_n(A) e^{i n \Phi}
\]

gives:

\[
h = -\mu A + \sum_n c_n(A) \frac{i n \mu}{1 - e^{-i n \mu}} e^{i n \Phi} = -\mu A + \sum_n c_n(A) \frac{n \mu}{2 \sin \left( \frac{n \mu}{2} \right)} e^{(i n \Phi + i \frac{n \mu}{2})}
\]

now \( h \) is the invariant of motion.