The Tracking Code RF-Track

Andrea Latina, CERN

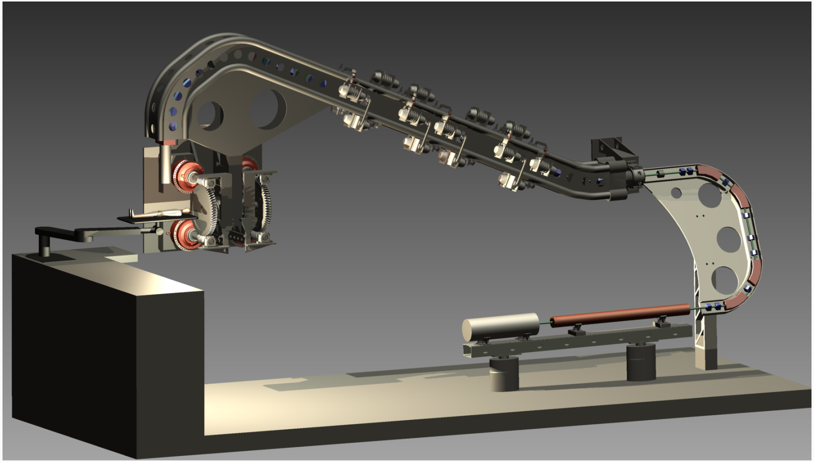
andrea.latina@cern.ch

Contents

Introduction and Highlights

Motivation for developing RF-Track: the TULIP project

A linac for hadron therapy featuring high-gradient S-band backward travelling-wave structures



S. Benedetti, A. Grudiev, and A. Latina, "High gradient linac for proton therapy", Phys. Rev. Accel. Beams 20, 040101 [2017]

RF-Track requisites and highlights

- It handles $\mathbb{C}$**omplex 3D field maps** of oscillating RF electromagnetic fields:
  - *Standing-wave*; *Backward* $\ll$ and *Forward* $\gg$ travelling-wave fields
  - *Static* electric and magnetic fields
  - Robust interpolation algorithms

- It can simulate particles with **any mass** and **charge**
  - No approximations, like $\beta \simeq 1$ or $\gamma \gg 1$, are made
  - It is used to simulate: *protons, ions, electrons, positrons, photons, muons, ...* from creation to ultra-relativistic
  - It implements **photocathodes**
  - It can simulate **mixed-species beams**

- Implements **high-order adaptive integration algorithms**
  - Can do **back-tracking**

- Implements several **collective effects**

- It's **modular, flexible,** and **fast**

RF-Track: minimalistic and physics-oriented

RF-Track is a module that can be used through **two different user interfaces**: one in **Octave** and one in **Python**. Written in **parallel** and **optimised** C++, RF-Track focuses solely on accelerator simulation:

- Flexible accelerator description and beam models
- Accurate integration of the equations of motion
- Robust field map interpolation
- Collective effects
- Easy implementation of imperfections and correction algorithms

For *"all the rest"* (ODE solvers, random number generation, special functions, . . .), it relies on two robust and well-known open-source libraries:

- GSL, "Gnu Scientific Library", provides a wide range of mathematical routines such as high-quality random number generators, ODE integrators, linear algebra, and much more

- FFTW, "Fastest Fourier Transform in the West", arguably the fastest free library to compute discrete Fourier transforms

Beam Models

Two beam models: tracking in space and in time

RF-Track implements two beam models:

1. **Beam moving in space**: `Bunch6d()`
   - All particles have the same S position
   - The equations of motion are integrated in dS: $S \rightarrow S + dS$ (moves the bunch element by element)

$$(x \text{ [mm]},\ x' \text{ [mrad]},\ y \text{ [mm]},\ y' \text{ [mrad]},\ t \text{ [mm/c]},\ P \text{ [MeV/c]})$$

2. **Beam moving in time**: `Bunch6dT()`
   - All particles are considered at same time t
   - The equations of motion are integrated in dt: $t \rightarrow t + dt$
   - Particles can have $P_z < 0$ or even $P_z = 0$: particles can move backward

$$(X \text{ [mm]},\ P_x \text{ [MeV/c]},\ Y \text{ [mm]},\ P_y \text{ [MeV/c]},\ Z \text{ [mm]},\ P_z \text{ [MeV/c]})$$

Two beam models: tracking in space and in time

RF-Track implements two beam models:

1. **Beam moving in space**: `Bunch6d()`
   - All particles have the same S position
   - The equations of motion are integrated in dS: $S \rightarrow S + dS$ (moves the bunch element by element)

$$(x \text{ [mm]},\ x' \text{ [mrad]},\ y \text{ [mm]},\ y' \text{ [mrad]},\ t \text{ [mm/c]},\ P \text{ [MeV/c]})$$

2. **Beam moving in time**: `Bunch6dT()`
   - All particles are considered at same time t
   - The equations of motion are integrated in dt: $t \rightarrow t + dt$
   - Particles can have $P_z < 0$ or even $P_z = 0$: particles can move backward

$$(X \text{ [mm]},\ P_x \text{ [MeV/c]},\ Y \text{ [mm]},\ P_y \text{ [MeV/c]},\ Z \text{ [mm]},\ P_z \text{ [MeV/c]})$$

For each macro particle also considers

$$\mathbf{m}: \text{ mass [MeV/c}^2\text{]}, \quad \mathbf{Q}: \text{ charge } [e^+]$$

$$\mathbf{N}: \text{ nb of particles / macroparticle}, \quad \mathbf{t_0}: \text{ creation time}^{(\star)} \quad \boldsymbol{\tau}: \text{ lifetime } \textbf{[NEW!]}$$

$^{(\star)}$ only for beams moving in time.

RF-Track can simulate **multi-species beams** and the **creation** and **decay** of particles.

Two tracking environments

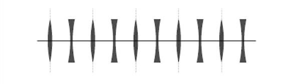
Lattice: for space integration

- A list of elements
- Tracks the particles element by element, along the longitudinal direction
- Elements can be arbitrarily misaligned

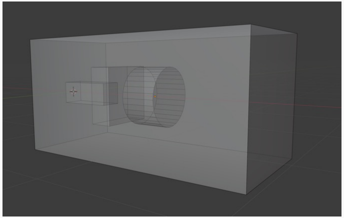
Volume: for time integration

- A portion of 3D space
- Elements can be placed anywhere
- Element misalignment via Euler angles (pitch, yaw, roll)
- Allows element overlap
- Allows creation of particles
- Can simulate cathodes and field emission
- Includes cathode mirror charges

Lattice

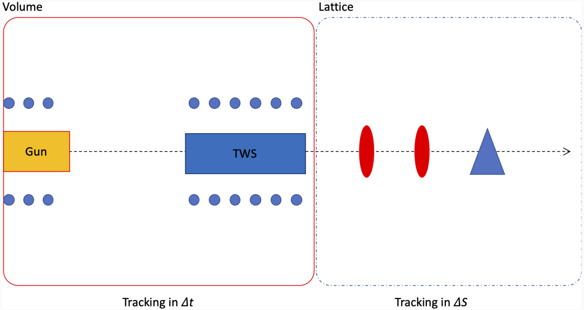


Volume



Lattice and Volume

Lattice and Volume can be used together or separately. Example: a photoinjector

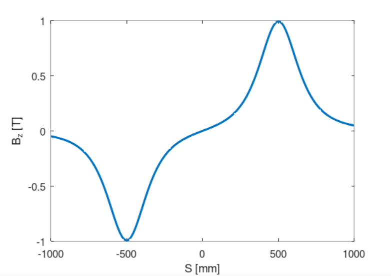


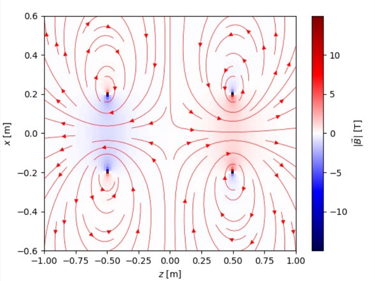
Typically, **Volume** (time integration) is suitable for **space-charge dominated regimes**, whereas **Lattice** (space integration) is suitable for **ultra-relativistic regions** of the machine.

*Notice that Volume*s can be inserted in a Lattice, and *Lattice*s can be placed in a Volume.

Example of Volume

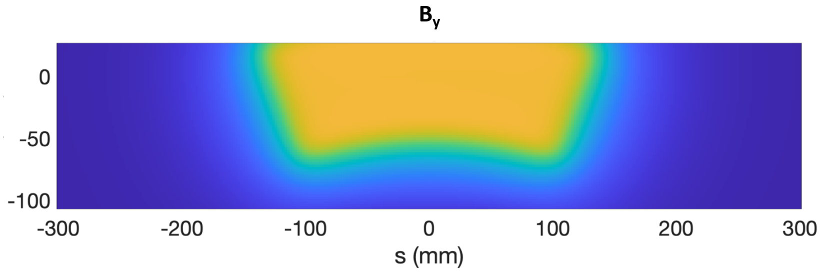
```
RF_Track;

L = 0;   % length [m]
B = 1;   % field at the center of the coil [T]
R = 0.2; % coil radius [m]

Cm = Coil(L, -B, R);
Cp = Coil(L, +B, R);

V = Volume();
V.add(Cm, 0, 0, -0.5, 'center');
V.add(Cp, 0, 0,  0.5, 'center');

figure(1)
Za = linspace(-1e3, +1e3, 1000); % mm
[E,B] = V.get_field(0, 0, Za, 0);

plot(Za, B(:,3));
xlabel('S [mm]');
ylabel('B_z [T]');
```





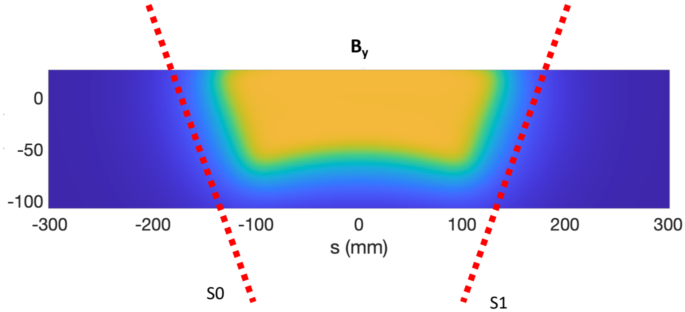
Volume as a Lattice element

Example of field map:



Volume as a Lattice element

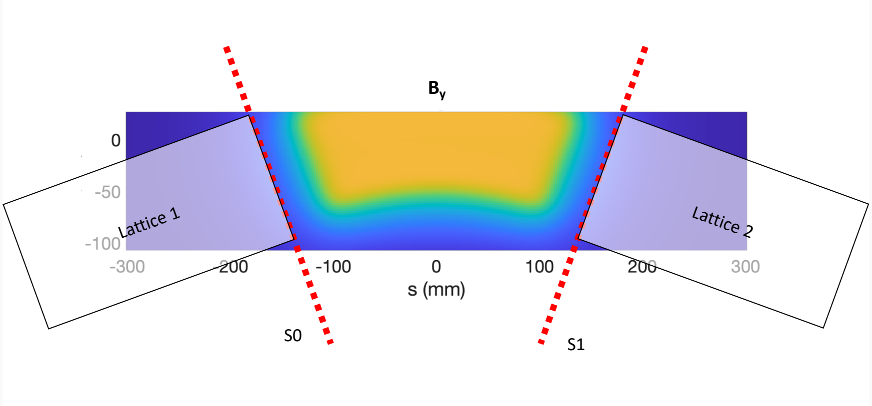
Boundaries of a Volume:



The boundaries of a Volume can have any orientation in space.

Volume as a Lattice element

Boundaries of a Volume:



A Volume can be sandwiched between two Lattices.

Beamline Elements

Overview of the beamline elements

1. **Standard set** of **matrix-based symplectic** elements:
   - **Sector bend**
   - **Quadrupole**
   - **Drift** (with an optional constant electric and magnetic fields, can be used to simulate e.g., rectangular bends)

Overview of the beamline elements

1. **Standard set** of **matrix-based symplectic** elements:
    - **Sector bend**
    - **Quadrupole**
    - **Drift** (with an optional constant electric and magnetic fields, can be used to simulate e.g., rectangular bends)

2. **Field maps** (see next slides)

Overview of the beamline elements

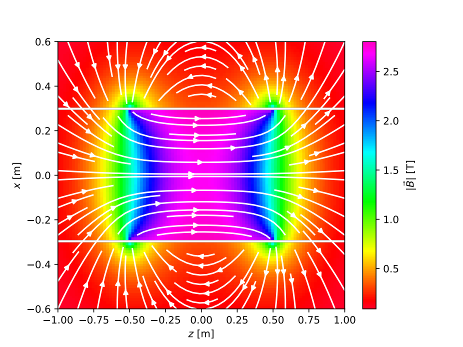
1. **Standard set** of **matrix-based symplectic** elements:
   - **Sector bend**
   - **Quadrupole**
   - **Drift** (with an optional constant electric and magnetic fields, can be used to simulate e.g., rectangular bends)

2. **Field maps** (see next slides)

3. **Special elements**:
   - **Absorber** (predefined materials: air, water, beryllium, lithium, tungsten, ...)
   - 3D analytic fields**: Coil** and **Solenoid**, **Undulator**, **Standing-wave** and **Traveling-wave** structures, **Adiabatic matching device**, **Toroidal Harmonics**
   - **LaserBeam** for Inverse Compton Scattering simulations
   - **Electron Cooler**
   - **Transfer Line**: tracks through an arbitrary lattice given in form of Twiss table (phase advances, momentum compaction, 1^{st} and 2^{nd} order chromaticity are considered)
   - **Screens**: to capture the phase space at any point with any orientation in space

The element Solenoid

In Lattice, a Solenoid is a standard **symplectic transfer matrix**. In Volume, a Solenoid is the three-dimensional field generated by an **arbitrary number of thin sheets of current**:



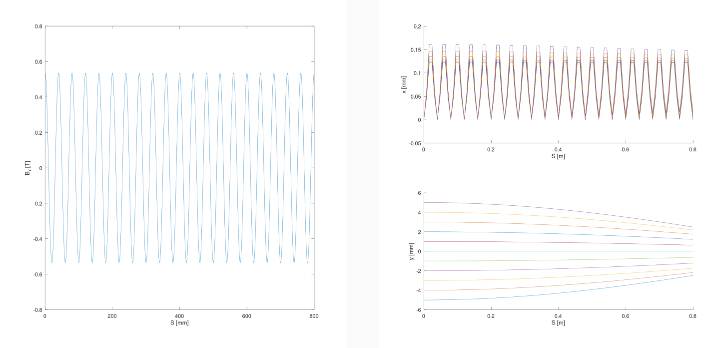
Solenoid field computed by RF-Track.

[Plot courtesy of Bernd Michael Stechauner, CERN]

The element Undulator

A symplectic implementation of an Undulator, using a 3D field.

```
U = Undulator (lperiod, K, nperiods, kx2=0);
```



Undulator field and single-partcle trajectories as computed by RF-Track.

Field maps

RF-Track can import several types of oscillating RF field maps, which are interpolated *linearly* or *cubically*

- **1D field maps** (on-axis field)
  - It uses Maxwell's equations to reconstruct the 3D fields off-axis, assuming cylindrical symmetry
- **2D field maps**: given a field on a plane, applies cylindrical symmetry
- **3D field maps** of oscillating electro-magnetic fields
  - Accepts 3D meshes of complex numbers
  - Accepts quarter field maps and automatically performs mirroring
  - For RF fields, allows specifying the input power supplied to the structure
  - Treats "NaN" in the field map as walls for accurate 3D loss maps

It also provides elements dedicated to **StaticElectric** and **StaticMagnetic** field maps

- They ensure curl-free (electric) and divergence-free (magnetic) interpolation of the field

Integration algorithms

In field maps and analytic fields, RF-Track integrates the equations of motion numerically:

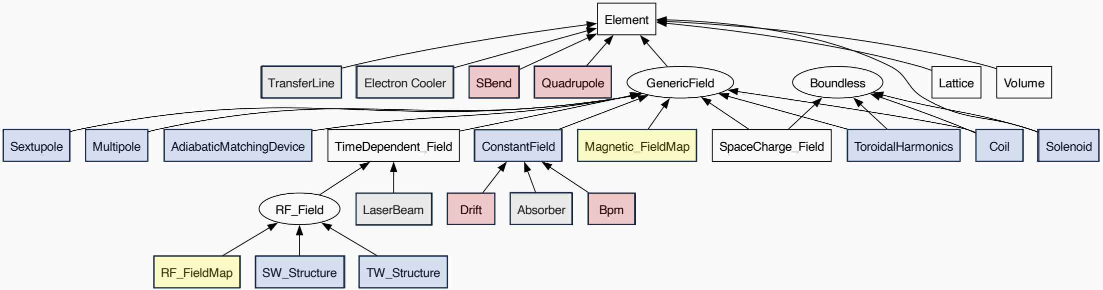
- The default is: "**leapfrog**":
  - ⋆ super fast, second-order accurate

- "**analytic**" algorithm:
  - ⋆ integration assuming a locally-constant EM field

- **Higher-order, adaptive algorithms** provided by GSL:
  - ⋆"**rk2**" Runge-Kutta (2, 3)
  - ⋆"**rk4**" 4th order Runge-Kutta
  - ⋆"**rkf45**" Runge-Kutta-Fehlberg (4, 5)
  - ⋆"**rkck**" Runge-Kutta Cash-Karp (4, 5)
  - ⋆"**rk8pd**" Runge-Kutta Prince-Dormand (8, 9)
  - ⋆"**msadams**" multistep Adams in Nordsieck form (order varies dynamically between 1 and 12)

⇒ backtracking is possible

Element hierarchy



- Field maps 1D, 2D, 3D
- Analytic fields 3D
- Matrix based
- Special elements

Collective and Single-particle Effects

Overview of the collective and single-particle effects

Collective effects:

- **Space-charge**, full 3D, Particle-in-Cell (FFT) or P2P
  - Full computation of electric and magnetic effects
  - Beam-beam effects are automatically included
  - Optionally considers mirror charges at cathode
- Two models of **Short-range wakefields**:
  1. Karl Bane's approximation
  2. 1D user-defined spline, longitudinal monopole or transverse dipole
- Two models of **Long-range wakefields:**
  1. Sum of damped oscillators. Takes modes: frequency, amplitude, and Q factor
  2. 1D user-defined spline, longitudinal monopole or transverse dipole
- Self-consistent **Beam loading** effect in TW and SW structures
  - Given: R/Q, group velocity, and Q factors along the structure, computes the beam loaded fields

Single-particle effects:

- **Incoherent Synchrotron Radiation** (from *any* fields)
- **Magnetic multipole kicks** for imperfection studies
- **Multiple Coulomb Scattering, Energy loss, Energy straggling** (recently updated)

These effects can be attached to **any element**, **simultaneously**.

Examples of Applications

Examples of applications

RF-Track is currently used for the design, optimisation, and simulation of:

- Medical applications (**DEFT facility**, collaboration CERN, CHUV, THERYQ), the **CLIC** and **FCC-ee positron sources** (CERN, IJCLab, PSI) and **FCC-ee pre-injector linacs** (CERN, PSI)

- **Linac4** (CERN), **Inverse-Compton Scattering sources** (CERN, IJCLab, INFN Ferrara, Korea University), and the **Cooling channel** of a future **Muon Collider** (CERN), etc.

I'll show two examples:

1. ADAM's RFQ
2. Cooling channel for a future muon collider
3. CLIC positron source
4. FCC positron source

1. The RFQ of the ADAM linear accelerator for proton therapy

«LIGHT is a normal conducting 230 MeV medical proton linear accelerator being constructed by ADAM.

For the commissioning, RFQ beam dynamics simulations were performed with RF-Track by simulating the particles through the 3D field map.»

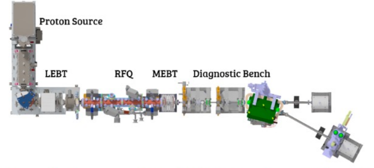


Figure 1: Layout of the LIGHT structures during the beam commissioning at 5 MeV.

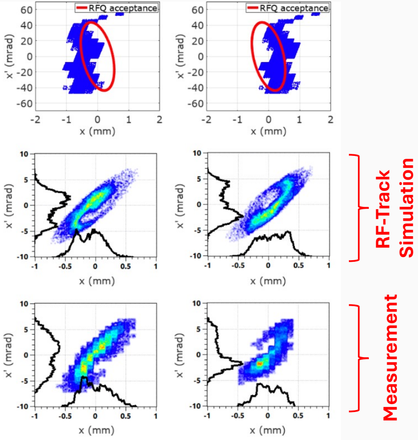
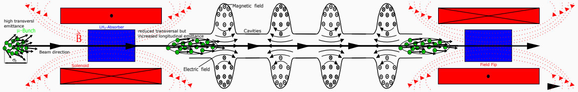
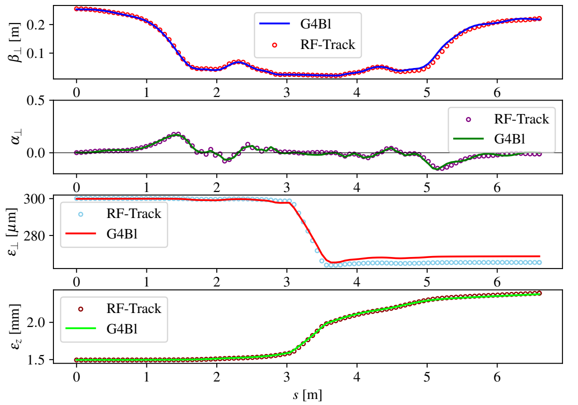


Figure 7: Horizontal phase space plots of the RFQ input beam when steered in the negative and positive x directions (first row), expected (second row) and the measured (third row) phase space plots after the RFQ for each case.

V. Dimov et al., "Beam commissioning of the 750 MHz proton RFQ for the LIGHT prototype", IPAC2018, Vancouver, BC, Canada, TUPAF002

2. Cooling channel for a future muon collider

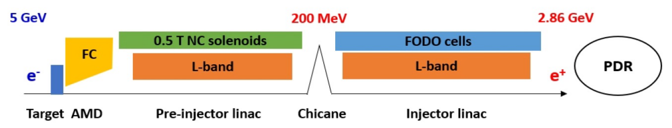




This simulation includes: 3D solenoids, standing-wave structures, absorbers – overlapping. Credits: **Bernd Stechauner** (CERN)

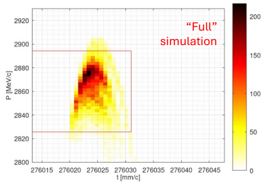
3. CLIC positron source

Start-to-end optimisation of the CLIC positron source

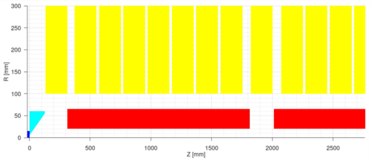


Geant4 (target simulation) + RF-Track (beam tracking)

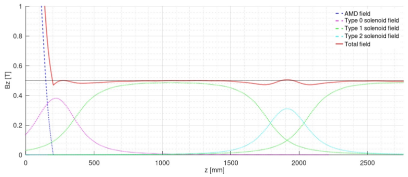
"Full simulation"



Longitudinal phase space @ 2.86 GeV



Schematic layout of solenoids



On-axis Bz of different components

- **Most realistic simulation of a positron source to date**
- **Start-to-end optimisation** with **higher positron yield than any previous studies ~1.8** @ 380 GeV (**~2.4** @ 3 TeV)

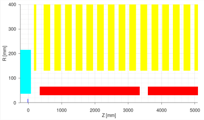
Yongke Zhao (CERN)

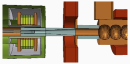
4. FCC positron source

Start-to-end optimisation of the **CLIC positron source**

Capture section

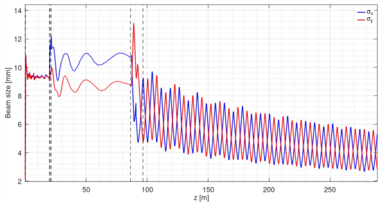
- AMD
  - HTS solenoid. 2D field. B_0 ~ 14.94 T
  - Target exit position: 40 mm (w.r.t. HTS B_0)
- Matching solenoid
  - L = 72 mm. 3D field (Maxwell3D)
  - B_0 ~ 0.245 T
  - Center position: 244 mm (w.r.t HTS B_0)
- Shielding
  - Tapered aperture (optimized by WP3). Impact on yield is negligible
- Capture Linac (CL)
  - RF structure length: 3 m. Iris radius: a_0 = 30 mm (constant)
  - N = 6, G = 13.3 MV/m, ϕ = [235 231 233 251 286 257]° (reoptimized by WP3)
  - Regular solenoids: L = 200 mm. 3D field (Maxwell3D). B_0 ~ 0.31 T. N = 9 (per structure)





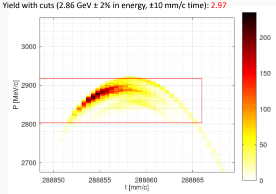
M. Daugaard

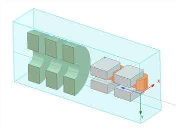
Beam spot size evolution along z

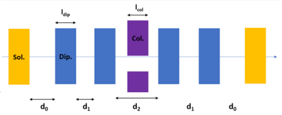


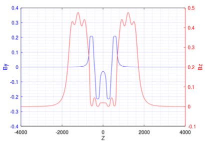
Final longitudinal phase space

- At positron linac exit
  - Total yield: **3.37**
  - Yield with cuts (2.86 GeV ± 2% in energy, ±10 mm/c time): 2.97





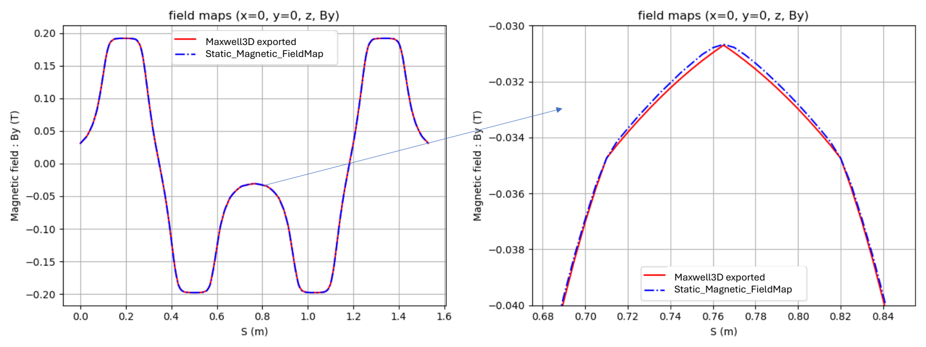




Yongke Zhao (CERN), **Riccardo Zennaro** (PSI), et al.

Field maps of magnetic elements

Static_Magnetic_FieldMap **corrects any input field map** (whether measured or computed), and makes it **physically correct**. This **ensures symplecticity**



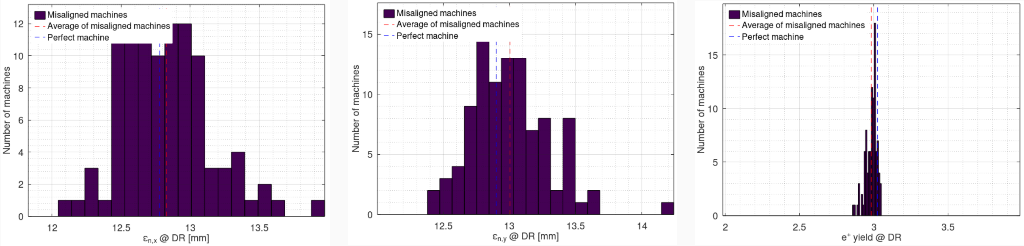
Magnetic chicane for the FCC-ee's positron source.

[Field map courtesy of **Riccardo Zennaro** (PSI); Plots courtesy of **Yuting Wang** (IJCLab)]

Static imperfections in the FCC-ee positron source

- Imperfections considered
  - **Position** error (x, y): σ = **100 um** for all elements
  - **Angular** error (roll, pitch, yaw): σ = **100 urad** for all elements, except for σ = **200 urad** for all NC solenoids and dipoles
  - Magnetic **strength** error: σ = **0.1%** for all magnets
  - RF **gradient** error: σ = **1%** for all RF structures
  - RF **phase** error: σ = **0.1°** for all RF structures
  - Beam **position jitter** (x, y): σ = **100 um** for e^+ beam from target
  - Beam **angular jitter** (x', y'): σ = **100 urad** for e^+ beam from target

- 100 random machines with imperfections
- Compared with perfect machine:
  - Average DR accepted e+ yield reduction: **1.3%** (2.97 → 2.93)
  - Average normalized X / Y emittance increase: **0.4% / 0.8%** (13.2 / 13.1 mm → 13.3 / 13.2 mm)
- Impact of considered imperfections is negligible



Yongke Zhao (CERN)

Summary and future developments

RF-Track:

- Minimalistic, parallel, fast – implements several collective effects
- Friendly and flexible, it uses Octave and Python as user interfaces
- Ideal for nontrivial optimisations and numerical experimentations
- Currently used to design and optimise: FCC-ee pre-injectors, CLIC and FCC-ee positron sources, muon cooling channel, RFQ, Linac4, ICS sources, medical accelerators...

Summary and future developments

RF-Track:

- Minimalistic, parallel, fast – implements several collective effects
- Friendly and flexible, it uses Octave and Python as user interfaces
- Ideal for nontrivial optimisations and numerical experimentations
- Currently used to design and optimise: FCC-ee pre-injectors, CLIC and FCC-ee positron sources, muon cooling channel, RFQ, Linac4, ICS sources, medical accelerators...

Next steps:

- Add Intra-beam scattering (IN TESTING PHASE), 3D Coherent synchrotron radiation (ASAP)
- Interface to SUPERFISH and CST Studio (done)

Summary and future developments

RF-Track:

- Minimalistic, parallel, fast – implements several collective effects
- Friendly and flexible, it uses Octave and Python as user interfaces
- Ideal for nontrivial optimisations and numerical experimentations
- Currently used to design and optimise: FCC-ee pre-injectors, CLIC and FCC-ee positron sources, muon cooling channel, RFQ, Linac4, ICS sources, medical accelerators...

Next steps:

- Add Intra-beam scattering (IN TESTING PHASE), 3D Coherent synchrotron radiation (ASAP)
- Interface to SUPERFISH and CST Studio (done)

Pre-compiled binaries and **more up-to-date documentation** are available here:

- `https://gitlab.cern.ch/rf-track`

Python users can visit `https://pypi.org/project/RF-Track/` and use:

- `pip install RF_Track`

Thank you for your attention!

Acknowledgements: many thanks to Dr. Avni Aksoy, Javier Olivares Herrador, Paula Desiré Valdor, Dr. Alexander Malyzhenkov, Vlad Musat, Bernd Stechauner, Dr. Elena Fol, Dr. Mohsen Kelisani, Dr. Yongke Zhao, Dr. Yanliang Han, Costanza Agazzi, Laura Gambino for their invaluable contributions.