The Tracking Code RF-Track: Taming High-Intensity Beams

Andrea Latina, CERN

andrea.latina@cern.ch

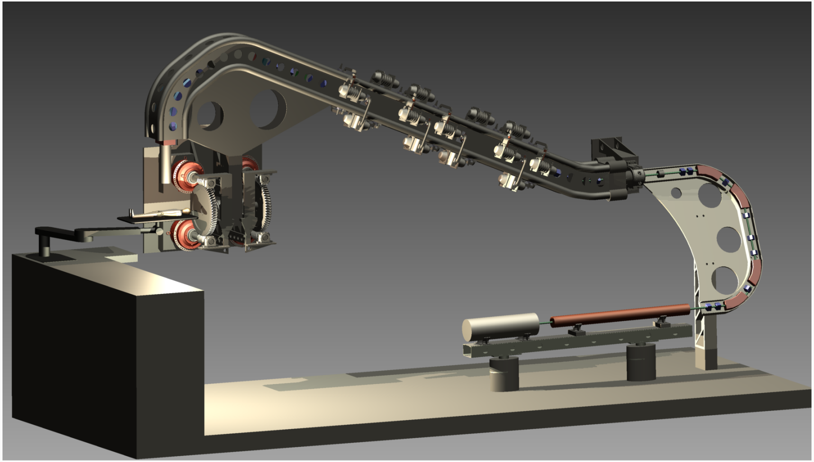
LEDS Workshop 2024 – 17-20 September, 2024 – Bern, Switzerland

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)

Requisites:

- Handle $\mathbb{C}$**omplex 3D field maps** of oscillating RF electromagnetic fields:
  - *Standing-wave*; *Backward* $\ll$ and *Forward* $\gg$ travelling-wave fields
- Provide conventional elements
- Be flexible and programmable

Highlights: The result is a code that can simulate particles with **any mass** and **charge**

- No approximations, like $\beta \simeq 1$ or $\gamma \gg 1$, are made
  - It is currently used to simulate: *protons, ions, electrons, positrons, muons, …* from creation to ultra-relativistic
- It can simulate **mixed-species beams**
- Implements **high-order adaptive integration algorithms**
  - Can do **back-tracking**
- Implements **collective effects**
- Is **modular, flexible,** and **fast**

RF-Track: minimalistic and physics-oriented

RF-Track is written in **parallel** and **optimised C++**, focusing *only* on accelerator simulation:

- Flexible accelerator description and beam models
- Accurate integration of the equations of motion
- Robust interpolation of field maps
- Collective effects
- Easy realisation 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

RF-Track provides **two alternative user interfaces**: one in Octave and one in Python.

Beam models

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

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

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: $S \to S + dS$ (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: $t \to t + dt$
   - 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

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

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

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

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

Bunch creation

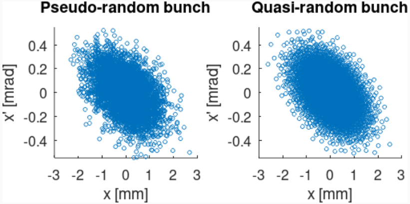
Particle bunches can be created in multiple ways:

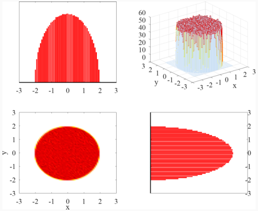
1. From **arbitrary distributions**, directly importing the phase space
2. From a set of **Twiss parameters**
3. From a **Photocathode** simulation similar to ASTRA's "Generator"

   1D distributions: '**gaussian**', '**uniform**', '**plateau**', '**parabola**'

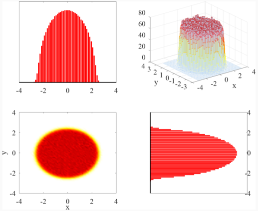
   2D distributions: '**radial-uniform**', '**radial-pleateau**', '**radial-gaussian**', '**radial-parabola**'

   3D distributions: '**ellipsoid**', '**isotropic**', '**fermi-dirac**'

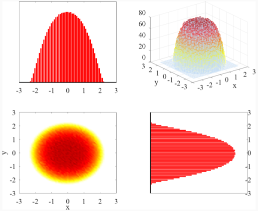




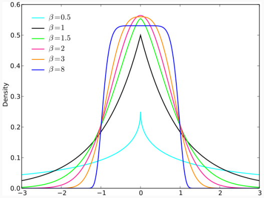
Radial-uniform



Radial-pleteau



Radial-parabola



Generalized Gaussian

Multi-bunch beams

Since version 2.3.0 (alpha release), it is possible to create multi-bunch beams.

Example of multi-bunch beam definition:

```
% Define a bunch
mass = electronmass;
charge = 1 * nC;
q = -1;
bunch = Bunch6d(mass, charge, q, phase_space);

% Define the train structure
num_of_bunches = 30; % train length
bunch_spacing = 1/3 * ns; % bunch spacing

% Define a beam
B0 = Beam(bunch, bunch_spacing, num_of_bunches);
```

A native multi-bunch implementation ease and speed up the computation of bunch-to-bunch collective effects.

Two tracking environments

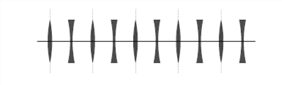
Lattice: for integration in space

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

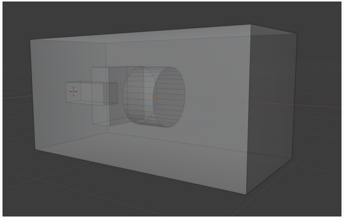
Volume: for integration in time

- 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

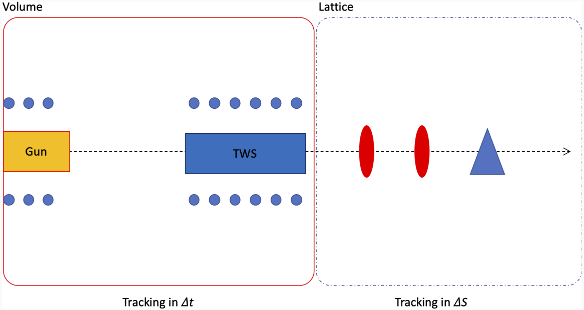


Volume



Lattice and Volume

Lattice and Volume can be used together or separately. Injector example:



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

*Volume*s can be inserted in a Lattice. And *Lattice*s can be placed in a Volume.

Example of Volume

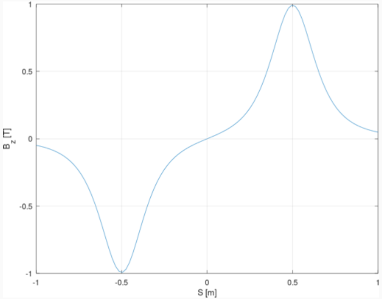
```
%% Load RF-Track
RF_Track;

%% Declare two coils
Cm = Coil(0.01, -1.0, 0.2);    % L length [m],
                               % B field at the center of the coil [T],
                               % R radius [m]
Cp = Coil(0.01, +1.0, 0.2);

%% Create a Volume
V = Volume();

% Add the two coils
V.add(Cm, 0, 0, -0.5);
V.add(Cp, 0, 0, 0.5);

% Set the boundaries
V.set_s0(-1.0); % -1 m
V.set_s1(+1.0); % +1 m
```



Example of Lattice

```
% load RF-Track
RF_Track;

% create a bunch from phase-space matrix
B0 = Bunch6d(electronmass, 200 * pC, -1, phase_space_matrix);

% create a lattice (1 FODO cell)
Lq = 0.4; % m
Ld = 0.6; % m
G = 1.2; % T/m

FODO = Lattice();
FODO.append (Quadrupole (Lq, G));
FODO.append (Drift (Ld));
FODO.append (Quadrupole (Lq, -G));
FODO.append (Drift (Ld));

% track the beam
B1 = FODO.track(B0);

% plot the phase space
T1 = B1.get_phase_space("%x %xp %y %yp");
scatter (T1(:,1), T1(:,2), "*");
xlabel ("x [mm]");
ylabel ("x' [mrad]");
```

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, or solenoids)

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, or solenoids)
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, or solenoids)

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

3. **Special elements**:
   - **Absorber** (predefined materials: air, water, beryllium, lithium, tungsten, ...)
   - 3D analytic fields**: Coil** and **Solenoid, Standing-wave** and **Traveling-wave** structures, **Adiabatic matching devices**, **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**: with any orientation in space

Field maps

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

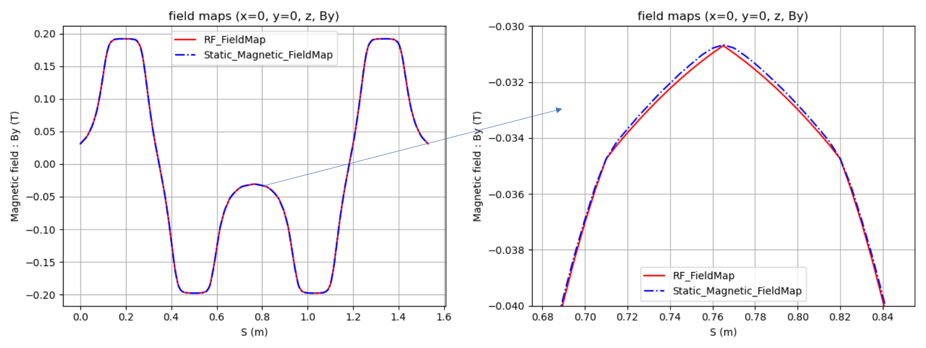
- **1D field maps** (on-axis field)
  - It uses Maxwell's equations to reconstruct the 3D fields off-axis, assuming cylindrical symmetry
- **2D field maps**: given a field on a plane, applies cylindrical symmetry
- **3D field maps** of oscillating electro-magnetic fields
  - It accepts 3D meshes of complex numbers
  - It accepts quarter field maps and performs mirroring automatically
  - For RF fields, it allows to specify the input power provided to the structure

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

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

The element “Static_Magnetic_FieldMap”

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

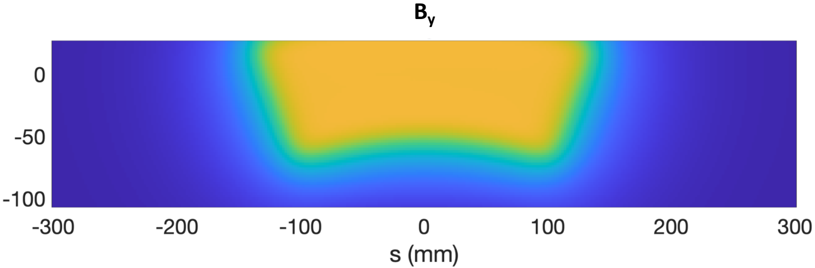


Magnetic chicane for the FCC-ee’s positron source.

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

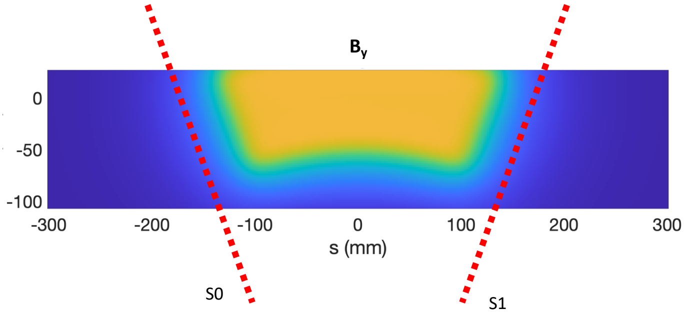
Volume as a Lattice element

Example of field map:



Volume as a Lattice element

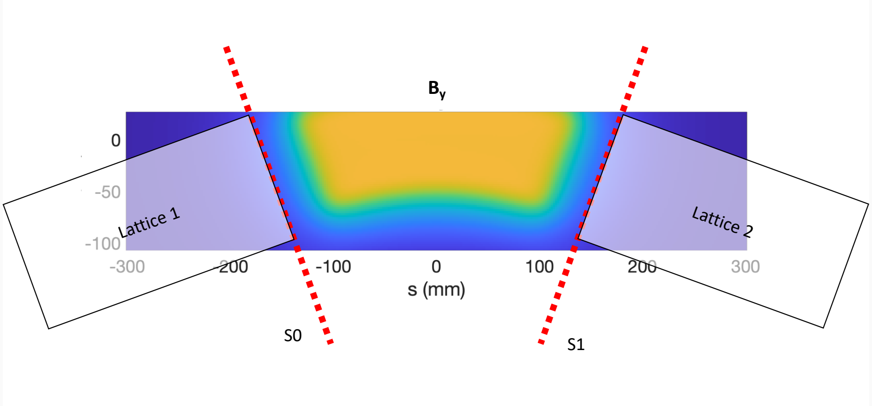
Boundaries of a Volume:



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.

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)

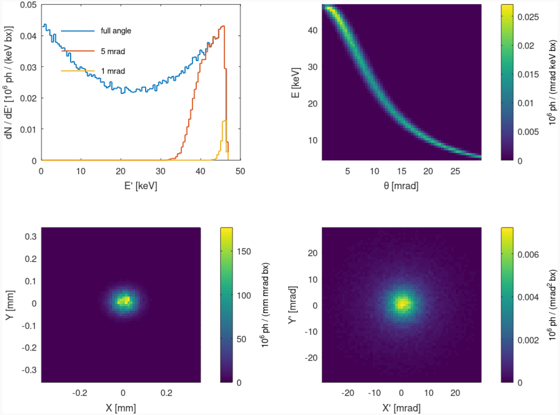
The element "LaserBeam" and Inverse Compton Scattering

A collision with a **LaserBeam** can be added to any Lattice, using the element "LaserBeam":

```
X_angle = 2; % deg, crossing angle

nX = sind(180-X_angle);
nZ = cosd(180-X_angle);

%% Define laser-beam IP region
FP = LaserBeam(); % ICS interaction point
FP.pulse_energy = 28; % mJ, laser pulse energy
FP.pulse_length = 5; % ps, laser pulse length
FP.wavelength = 1030; % nm, laser wavelength
FP.set_direction (nX, 0, nZ); % laser incoming direction
FP.length = IP_length;
FP.set_IP_position (IP_length); % m
FP.R = 0.035; % mm, laser rms radius at waist, Gaussian profile
FP.M2 = 1.1; %
```



The **photons** are added to the bunch and **transported** through the beam line.

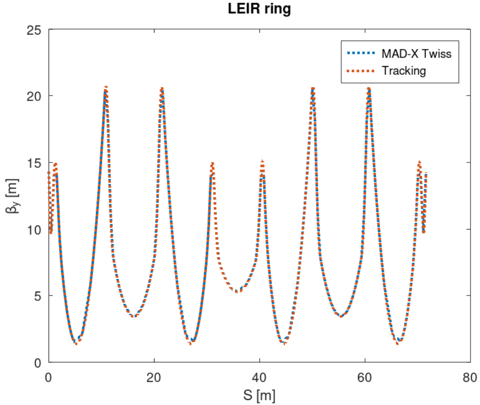
The element “TransferLine”

The element **TransferLine** allows to insert a full **MAD-X lattice** into RF-Track, just using the **Twiss file**.

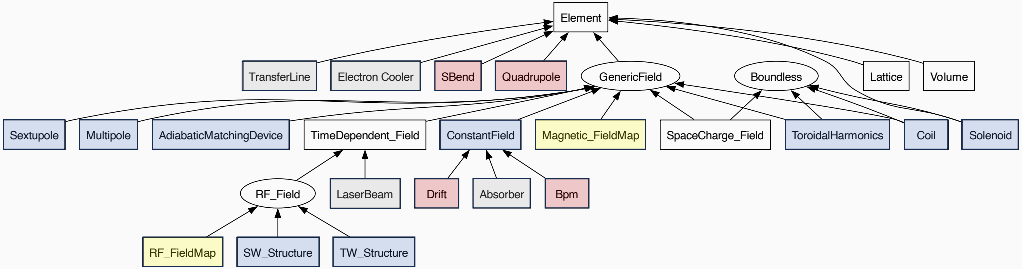
```
% Input a MAD-X twiss file
R = TransferLine ('twiss_file.tfs');

% or a TWISS matrix
R = TransferLine (RING_TWISS, DQx, DQy, momentum_compaction, P_ref);
R.set_nsteps(100);
R.set_tt_nsteps(1000);
R.set_cfx_nsteps(100);
```

Collective effects can be distributed along the lattice.



Elements 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
- **Short-range wakefields**:
  - Karl Bane's approximation
  - 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** (recently updated)

Space-charge effects (1/4)

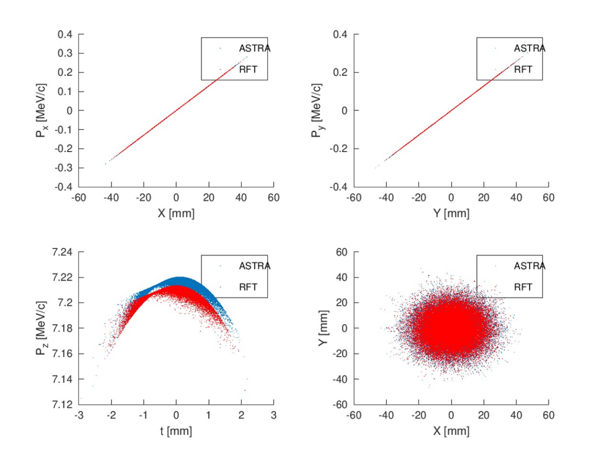
Benchmark against ASTRA:

Simulation of the **CLEAR photoinjector**:

- Q = 600 pC
- Gun, Ez = 100 MV/m, f = 3 GHz
- Peak energy phase for the reference particle
- Solenoid, Bz = 0.25 T ON | **OFF**
- Space-charge ON | **OFF**

50’000 macro particles

Solenoid **OFF** – Space-charge **OFF**



Space-charge effects (2/4)

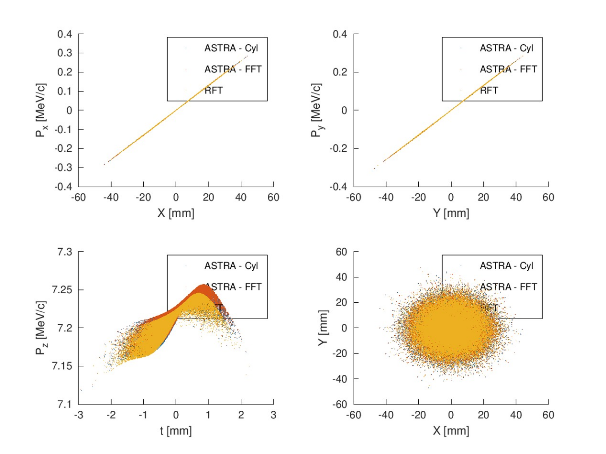
Benchmark against ASTRA:

Simulation of the **CLEAR photoinjector**:

- Q = 600 pC
- Gun, Ez = 100 MV/m, f = 3 GHz
- Peak energy phase for the reference particle
- Solenoid, Bz = 0.25 T ON | **OFF**
- Space-charge **ON** | OFF

50’000 macro particles

Solenoid **OFF** – Space-charge **ON**



Space-charge effects (3/4)

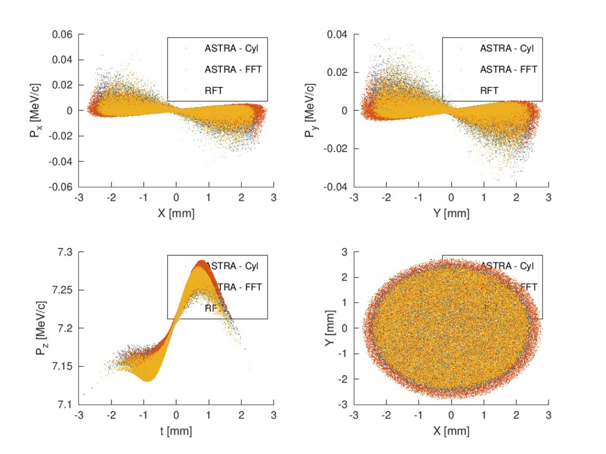
Benchmark against ASTRA:

Simulation of the **CLEAR photoinjector**:

- Q = 600 pC
- Gun, Ez = 100 MV/m, f = 3 GHz
- Peak energy phase for the reference particle
- Solenoid, Bz = 0.25 T **ON** | OFF
- Space-charge **ON** | OFF

50’000 macro particles

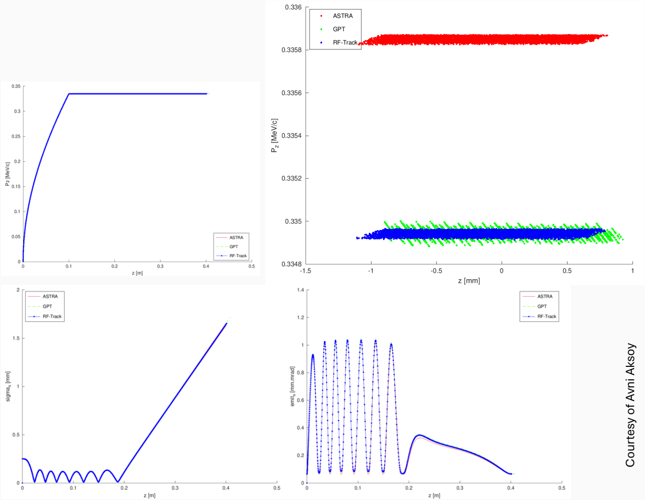
Solenoid **ON** – Space-charge **ON**



Space-charge effects (4/4)

Electrostatic Gun (Astra, GPT, RFT)

Ez = -1 MV/m
B = realistic solenoid, B max = 0.25 T
No SC



Courtesy of Avni Aksoy

Beam loading in traveling-wave structures

A **power diffusive model** computes beam loaded field in an RF structure.
Both **transient** and **steady state** can be computed.
Beam-loading **compensation** schemes can be computed.

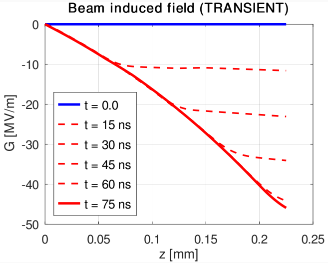
Usage example:

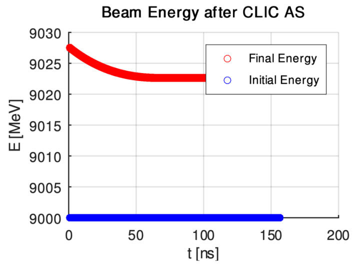
```
%% Transient Beam loading
BL = BeamLoading (Ncells, freq, phaseadvance, QQ, R_Q, VG);

%% RF-element from field map
TWS = RF_FieldMap_1d_CINT ( Ez, hz, L, freq, +1 );
TWS.set_odeint_algorithm( "rk2" );
TWS.set_P_map( Pmap );
TWS.set_P_actual( Pactual );

TWS.add_collective_effect ( BL );
TWS.set_cfx_nsteps ( 20 );
```

The plots show beam loading effects in a CLIC accelerator structure with a beam of 352 x 600 pC bunches with a 2 GHz bunch spacing.



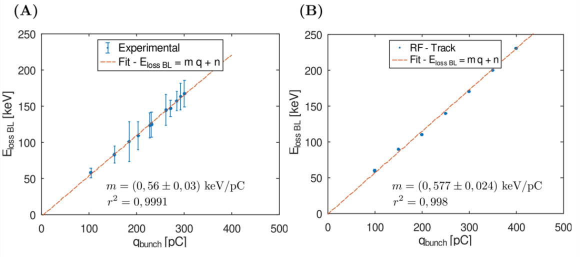


Beam loading in standing-wave structures

Beam-loading effects have also been computed in standing-wave structures.

Follow the example of the photoinjector of the CLEAR test facility at CERN. The plots show the beam-induced energy loss of a train of 150 bunches with 1.5 GHz bunch-spacing, as a function of the bunch charge.

Beam parameter (end of linac)	Value range
Energy	60 - 220 MeV
Bunch charge	0.01 - 1.5 nC
Normalized emittances	3 um for 0.05 nC per bunch 20 um for 0.4 nC per bunch (in both planes)
Bunch length	~100 um - 1.2 mm
Relative energy spread	< 0.2 % rms (< 1 MeV FWHM)
Repetition rate	0.8 - 10 Hz
Number of micro-bunches in train	1 - 150
Micro-bunch spacing	1.5 or 3.0 GHz



(A) Experimental measurements
(B) RF-Track simulation

J. H. Olivares et al., "Implementation of the beam-loading effect in the tracking code RF-track based on a power-diffusive model", Frontiers in Physics, 2024

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 five examples:

1. ADAM's RFQ
2. ThomX ICS Source
3. Electron Cooling at LEIR
4. Multiple Coulomb Scattering
5. Muon Cooling Channel

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.»

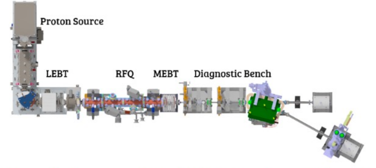


Figure 1: Layout of the LIGHT structures during the beam commissioning at 5 MeV.

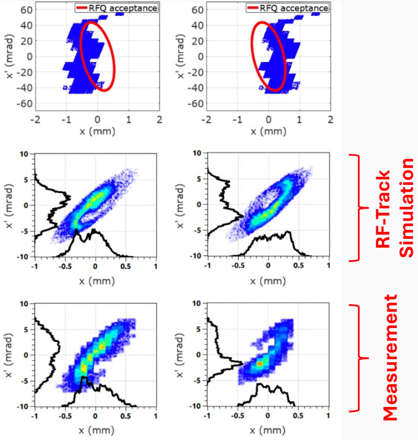


Figure 7: Horizontal phase space plots of the RFQ input beam when steered in the negative and positive x directions (first row), expected (second row) and the measured (third row) phase space plots after the RFQ for each case.

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

2. ThomX ICS source (1/4)

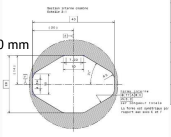
RF-Track helped to solve a serious design issue:

- 8 Dipoles
- 24 Quadrupoles
- 12 Sextupoles
- 2 Kickers
- 1 Septum
- 1 RF cavity
- 12 BPM
- 12 Correctors

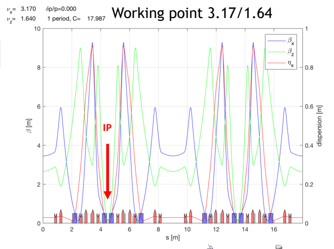
ThomX SR: L = 18 m, T = 60 ns, f_{rep} = 16.7 MHz

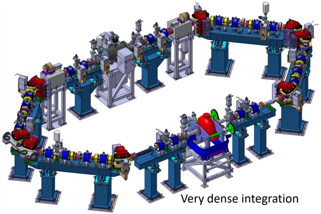
Parameter	Value/Units
Beam energy	50-70 MeV
Bunch Charge	1 nC
Bunch length (rms)	~30 ps
Circumference	18 m
Revolution frequency	**16.7 MHz**
Current	16.7 mA
RF frequency/Harmonics	**500/30 MHz**
Momentum compaction	0.0125 - 0.025
Betatron tunes	3.17/1.64
Natural chromaticity	-9/-13
Damping time trans./long.	1.2/0.6 s
Repetition frequency	50 Hz (20 ms)
Beam size at the IP	70 µm
Nominal RF Voltage/cavity	300 kV (500 kV max)
Energy loss per turn	1.57 eV

28x40 mm



Working point 3.17/1.64



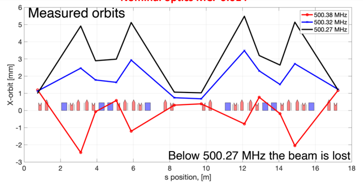


Slide courtesy of Viacheslav Kubytskyi (IJCLab)

2. ThomX ICS source (2/4)

An unexpected find: shorter circumference

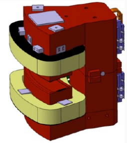
Nominal optics MCF 0.014

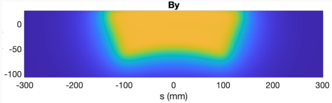


- **The RF frequency is found experimentally to be 0.3 - 0.4 MHz higher than the nominal. What is the reason?**
- **Need explicit simulation!**

Short and small-radius dipoles, long fringe fields

Features of dipoles	
Quantity	14 + 1 (pre-serie)
Radius of curvature	352 mm
Main field B_0	0.7 Tesla
Gap	42 mm
Good field region	+/- 20mm
Integral of field	184.59 mT.m
Current max.	275 Amp
Beam energy	from 50 to 70 MeV





Slide courtesy of Viacheslav Kubytskyi (IJCLab)

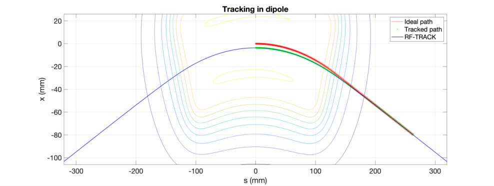
2. ThomX ICS source (3/4)

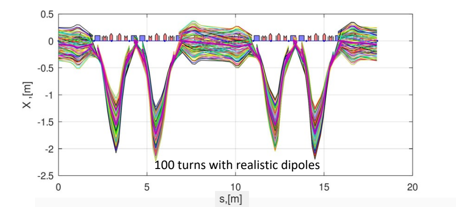
"Experiment on the table" with RF-track

First study to measure ring frequency :

- Lattice with dipoles represented by **SBEND** (usual way) : F = 500.02 MHz
- Lattice with dipoles represented by **VOLUME** with realistic magnetic field : F =500.38 MHz, dispersive orbit. The same effect as in the experiment!

It was found that the beam trajectory in the dipoles is shorter wrt. to the ideal path => shorter pathlength and so smaller total circumference





Big step in understanding of the problem

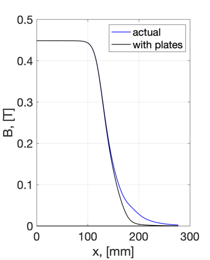
Slide courtesy of Viacheslav Kubytskyi (IJCLab)

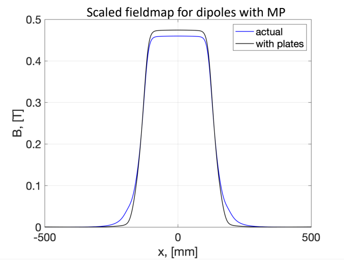
2. ThomX ICS source (4/4)

"Experiment on the table" with RF-track

Studies to compensate dipole fringing fields and retrieve nominal frequency by:

- Displacement of dipole
- Adding metallic plates to reduce fringing field
- Mechanical extension of the ring by +12mm



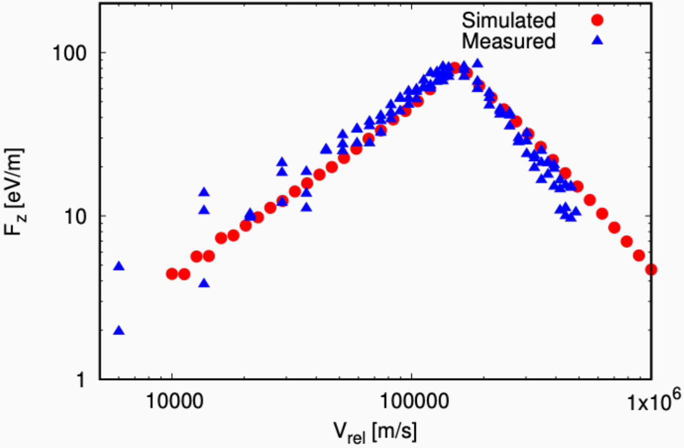


Correct scaling of magnetic field for the magnet with plates allows to recover the nominal frequency

Slide courtesy of Viacheslav Kubytskyi (IJCLab)

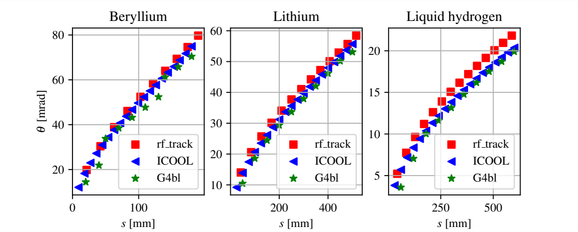
3. Electron Cooling at the CERN's Low Energy Ion Ring (LEIR)

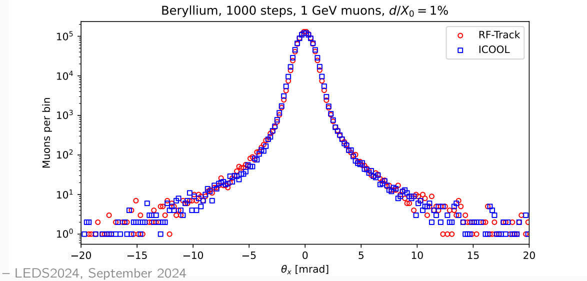
In 2019 we measured and benchmarked the cooling force as a function of ion-electron relative velocity measured at LEIR (blue) and simulated with RF-Track (red).



4. Absorber element and Multiple Coulomb Scattering

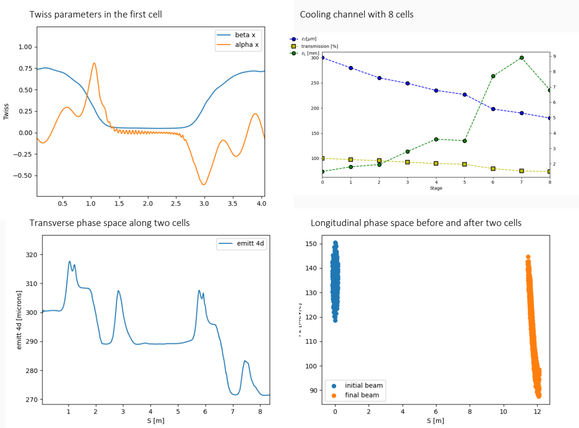
Improvement at large scattering angles [Credits: Bernd Stechauner (CERN)]





5. Muon cooling channel optimisation

Credits: Elena Fol (CERN)



This simulation includes: a constant solenoid field, realistic 3D solenoid, several standing-wave structures, and the absorber, simultaneously together and overlapping.

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:

- Implement Intra-beam scattering (Paula), 3D Coherent synchrotron radiation (ASAP)
- Interfaces 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:

- Implement Intra-beam scattering (Paula), 3D Coherent synchrotron radiation (ASAP)
- Interfaces 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 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 invlauable contributions.