Linac4's RFQ Simulations in RF-Track

Andrea Latina (BE-ABP-LAF)

28 April 2023

RF-Track

RF-Track is a tracking code that exists as module loadable within Octave or Python

Written in parallel C++, it uses the opensource libraries: GSL and FFTW

- GNU Scientific Library, GSL, numerical algorithms
- FFTW, Fast Fourier Transforms

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

https://zenodo.org/record/4580369

Currently used for the simulation and optimisation of:

- medical linacs: DEFT for electron FLASH therapy, and also light ions
- Inverse-Compton scattering sources
- The CLIC and FCC-ee positron sources
- The electron and positron linacs of the FCC-ee injector chain
- The cooling channel of muon collider
- More…

Previously published…

ADAM's RFQ Commissioning

«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.»

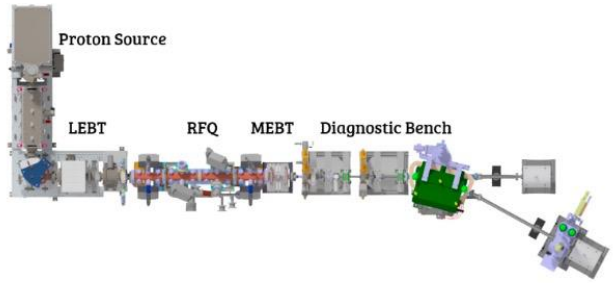


Figure 1: Layout of the LIGHT structures during the beam commissioning at 5 MeV.

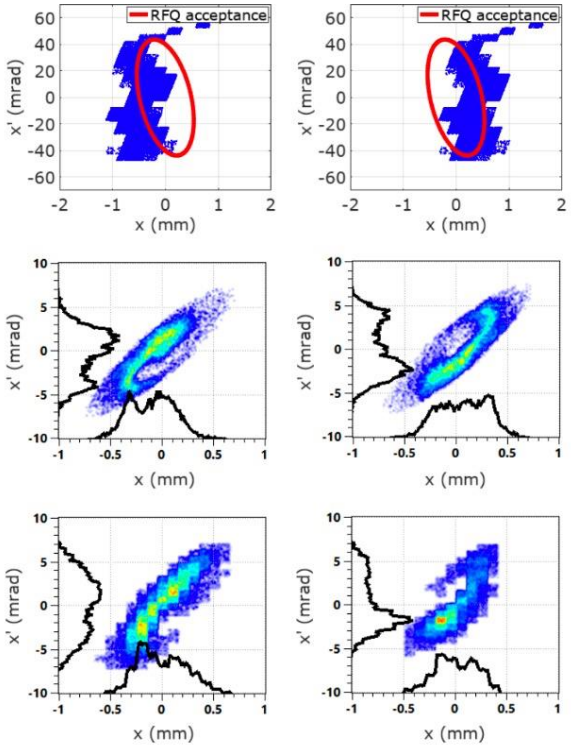


Figure 7: Horizontal phase space plots of the RFQ input beam when steered in the negative and positive x directions (first row), expected (second row) and the measured (third row) phase space plots after the RFQ for each case.

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

Beam Models

Beams can contain particles of any species, at any energy

Each particle carries a current

Two beam models can be used

1. *Bunch6d*

   (x, x', y, y', t, P)

   - Integration in space

2. *Bunch6dT*

   (x, P_x, y, P_y, z, P_z)

   - Integration in time

   - *Particles can be created at any time*

   - *Particles can be created with 0 initial momentum*

   - *Particles can move backward*

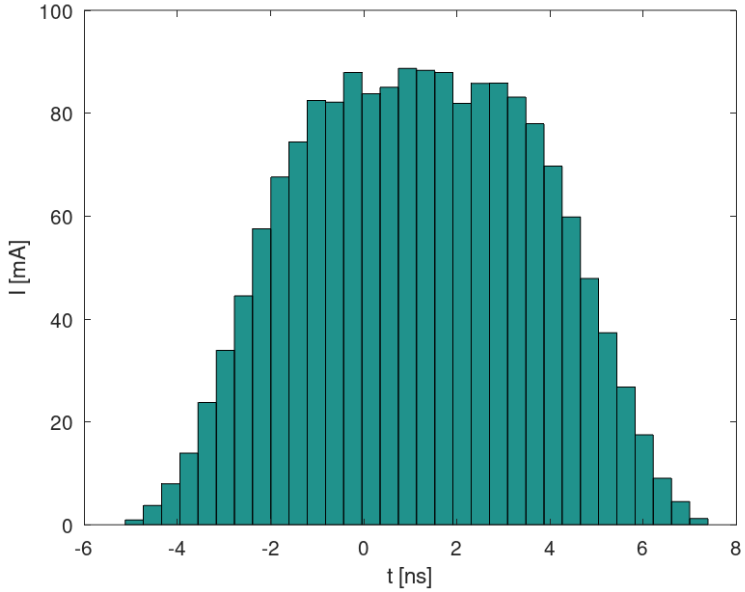
Example of simulation script

```
%% Load RF-Track
RF_Track;

%% Load the RFQ field map and create the RFQ element
RFQ = load_RFQ (0.0);
RFQ.set_tt_nsteps (100);

%% Load the beam
B0 = load_beam_TRAVEL ('~/linac4/beam_inputRFQ.txt');

%% Create lattice
L = Lattice ();
L.append (RFQ);

%% Perform tracking
tic
B1 = L.track (B0);
toc
```

It can import lattice files from MAD-X

Simulation setup

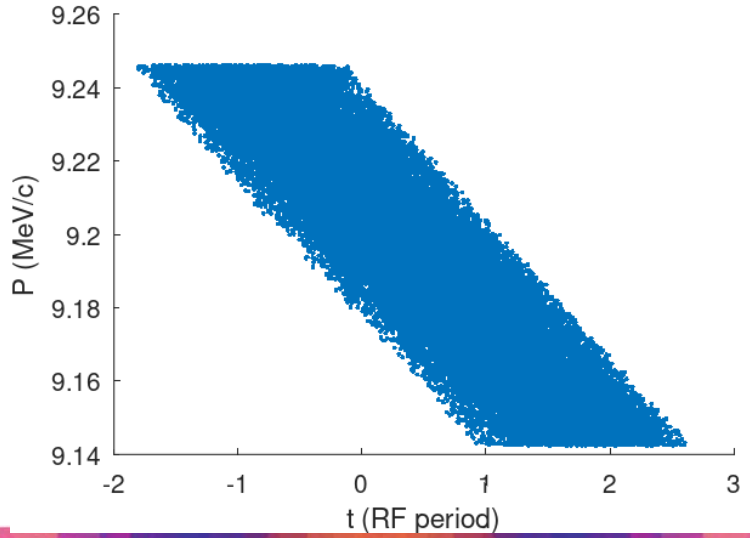
- Input distribution:
  - 47’549 macro-particles
  - Giulia: 88 mA
  - I considered 666 pC total charge



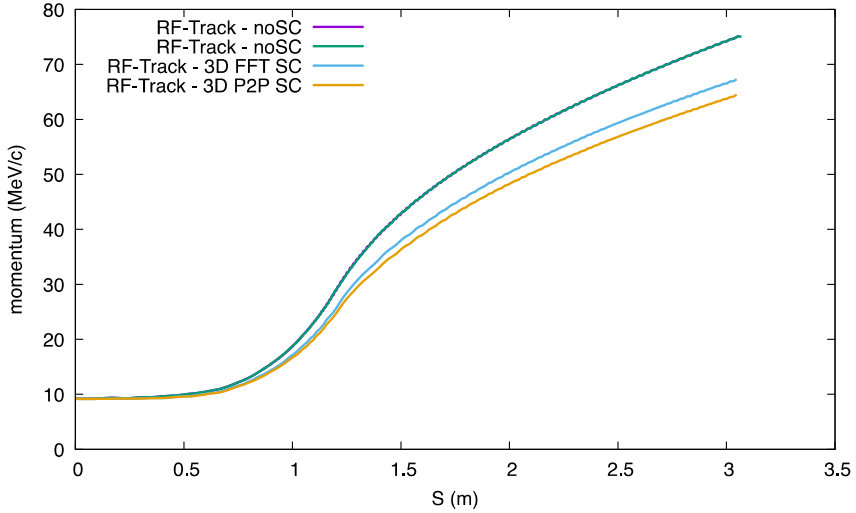
f = 352.2 MHz

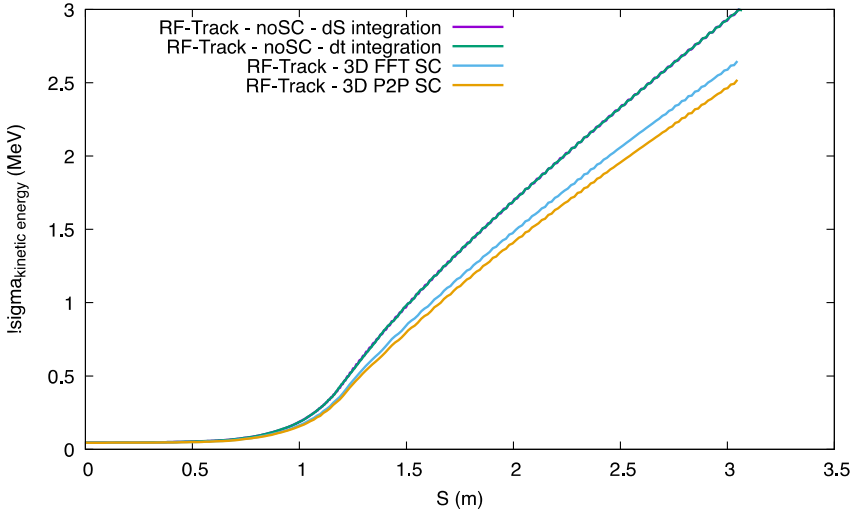
1 rf period = 2.83 ns

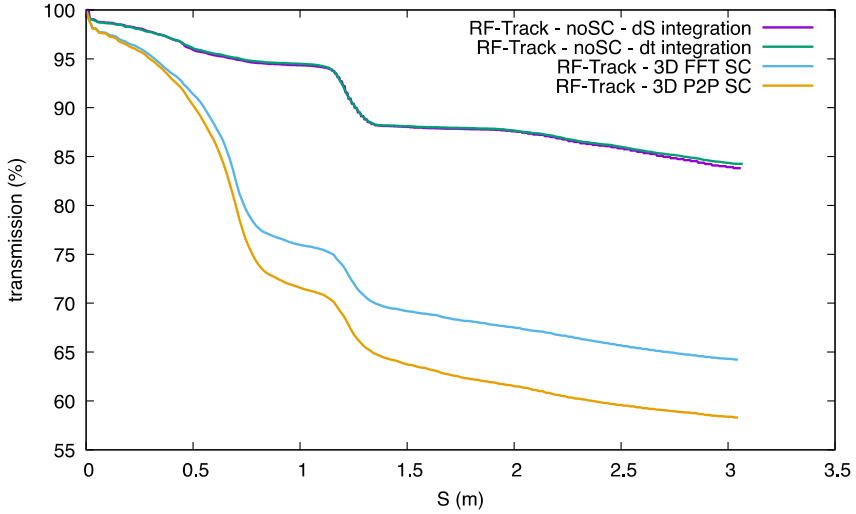
- RF-Track, three cases
  - No SC
  - 3D FFT SC
  - 3D P2P SC



RF-Track tracking

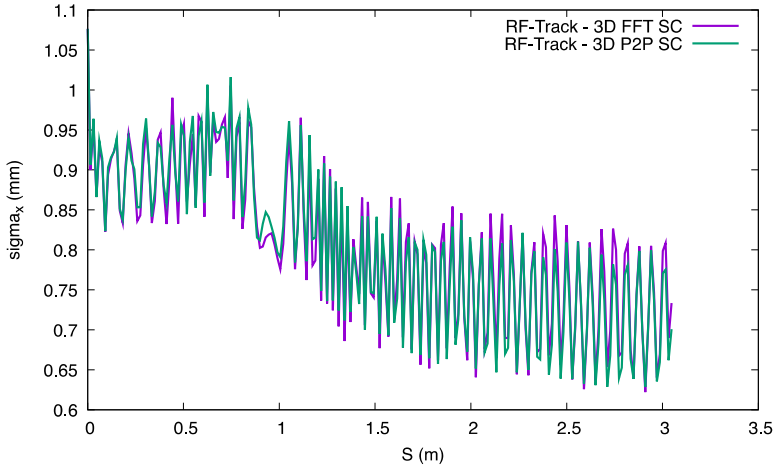


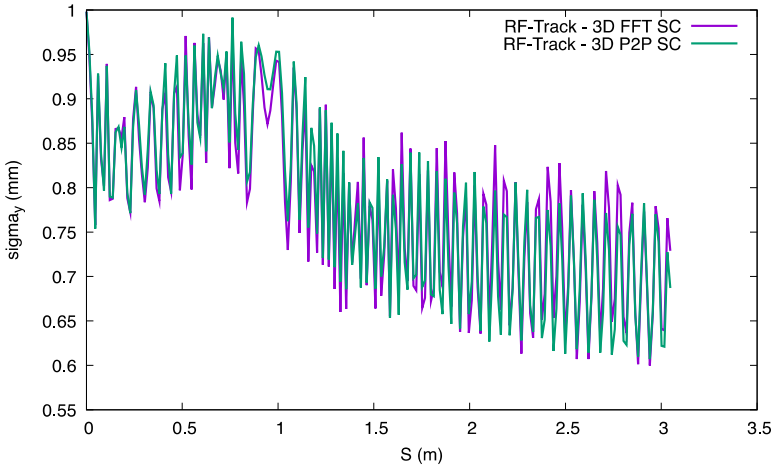


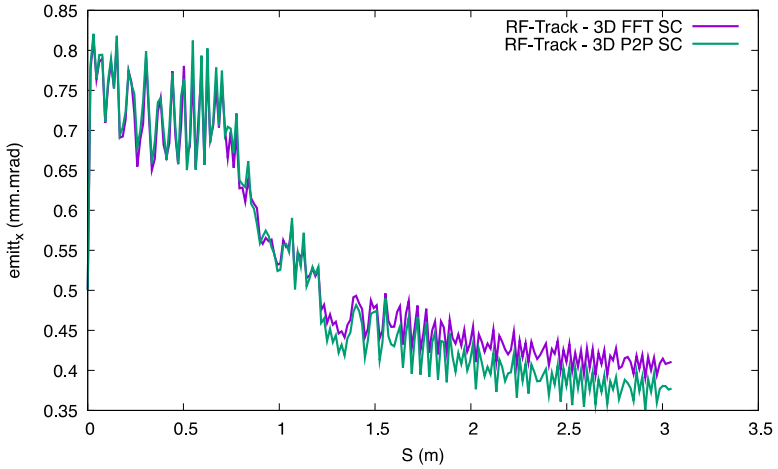


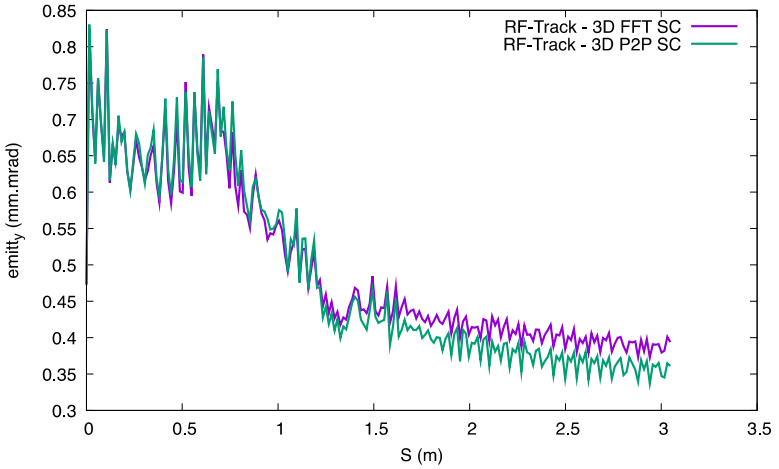
Transmission
- No SC = 84%
- 3D FFT SC = 64%
- 3D P2P SC = 58%

RF-Track tracking



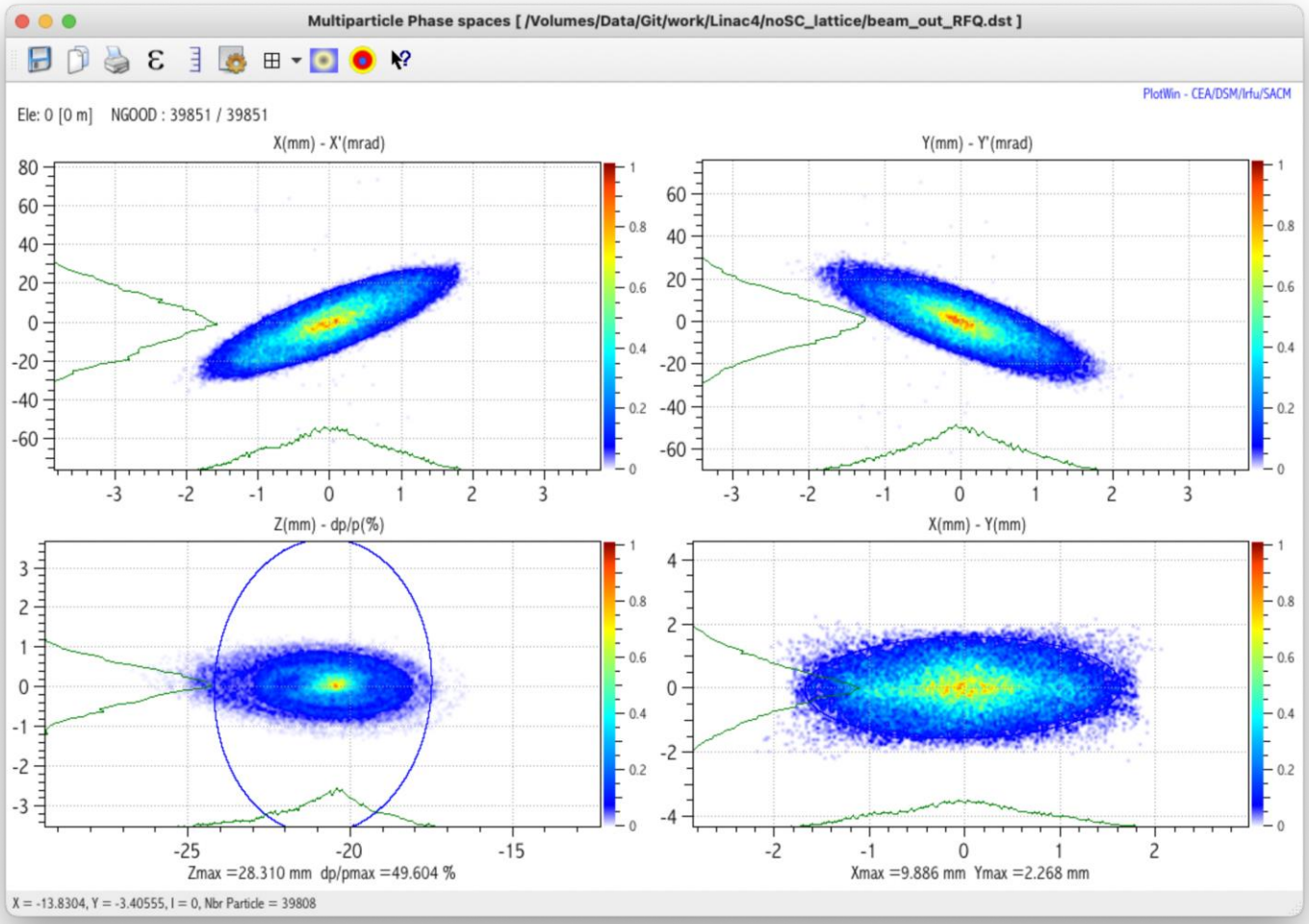




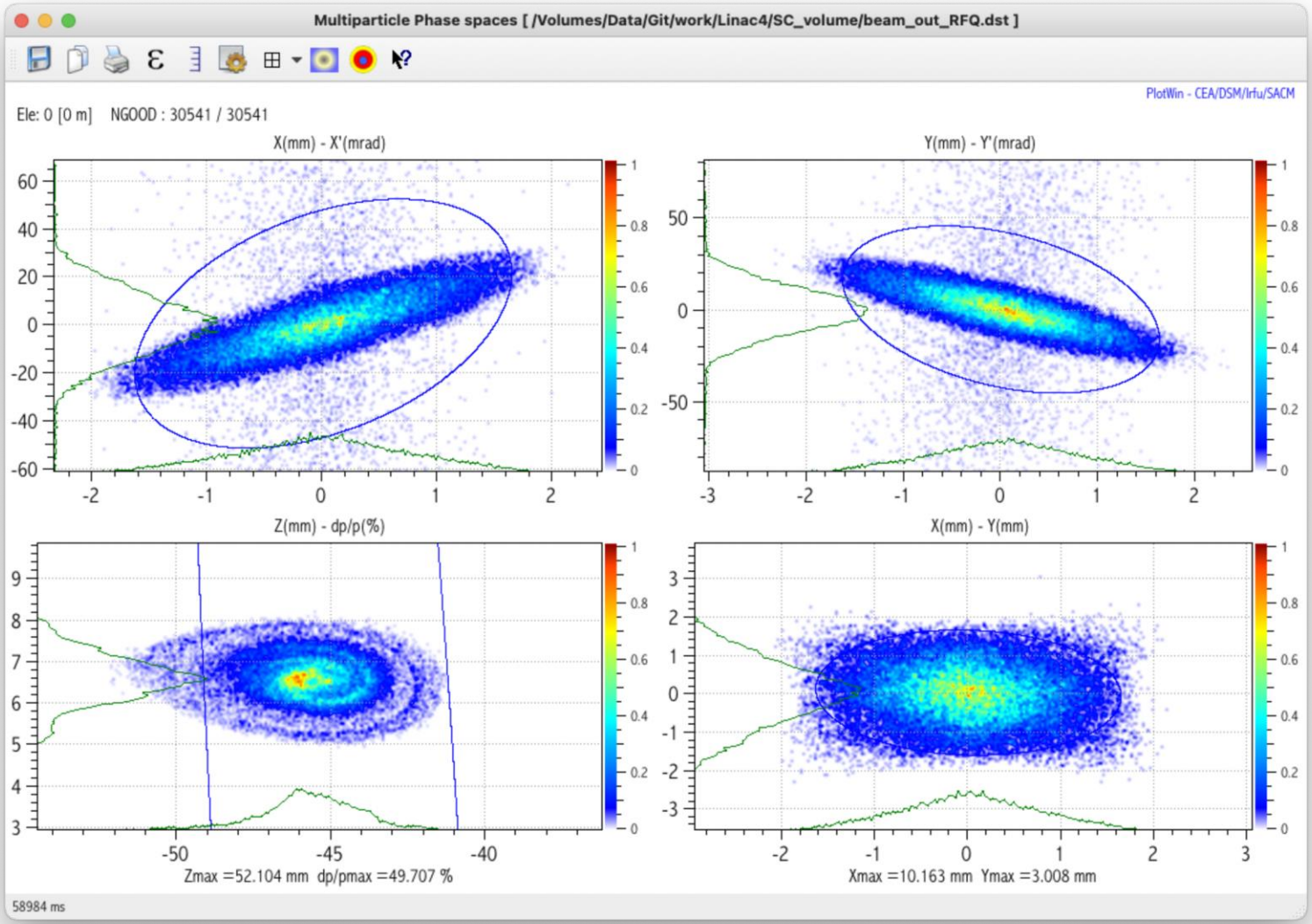


Very good agreement between 3D FFT and P2P space-charge

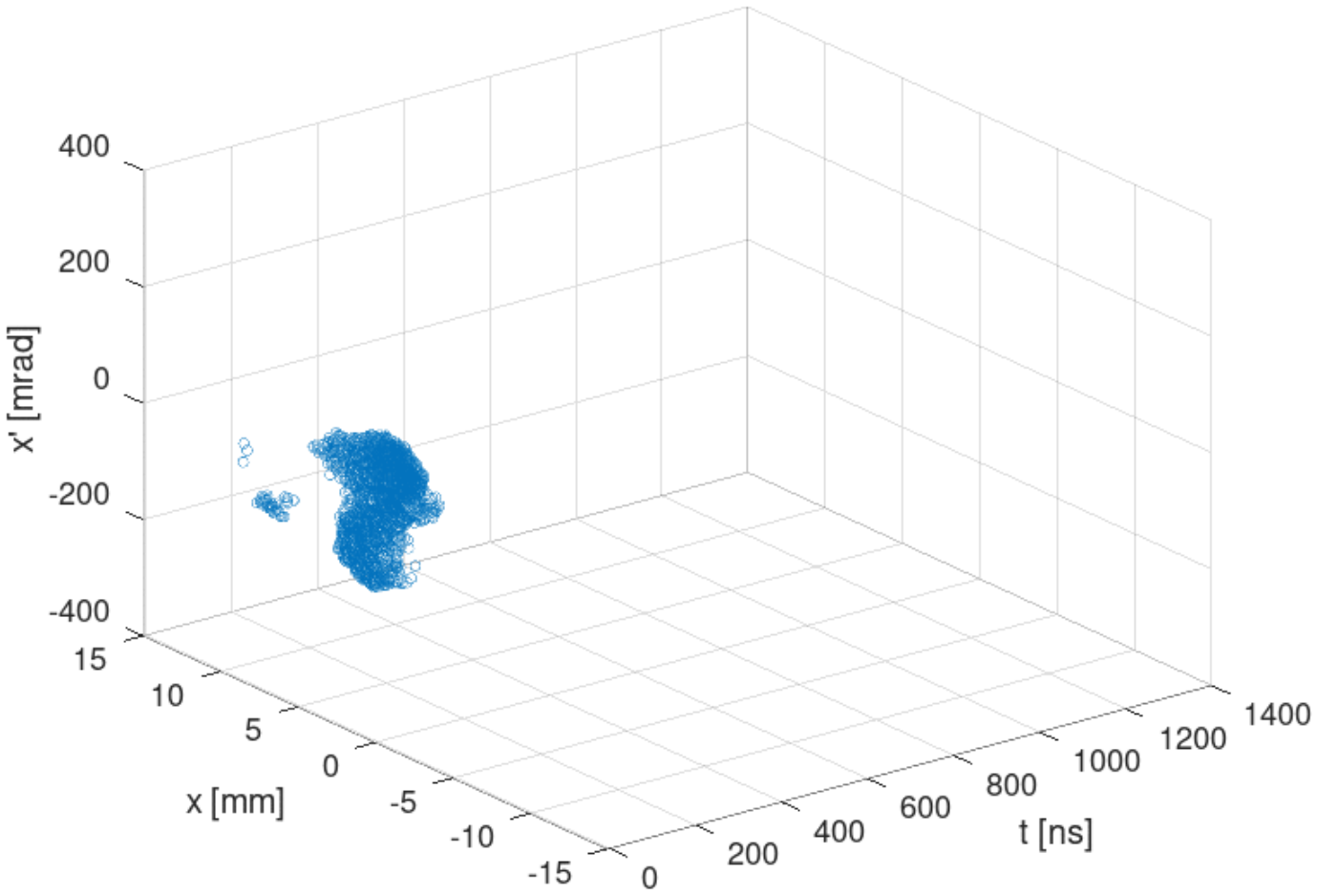
RFQ simulation – no SC
RF-Track can save as DST file directly



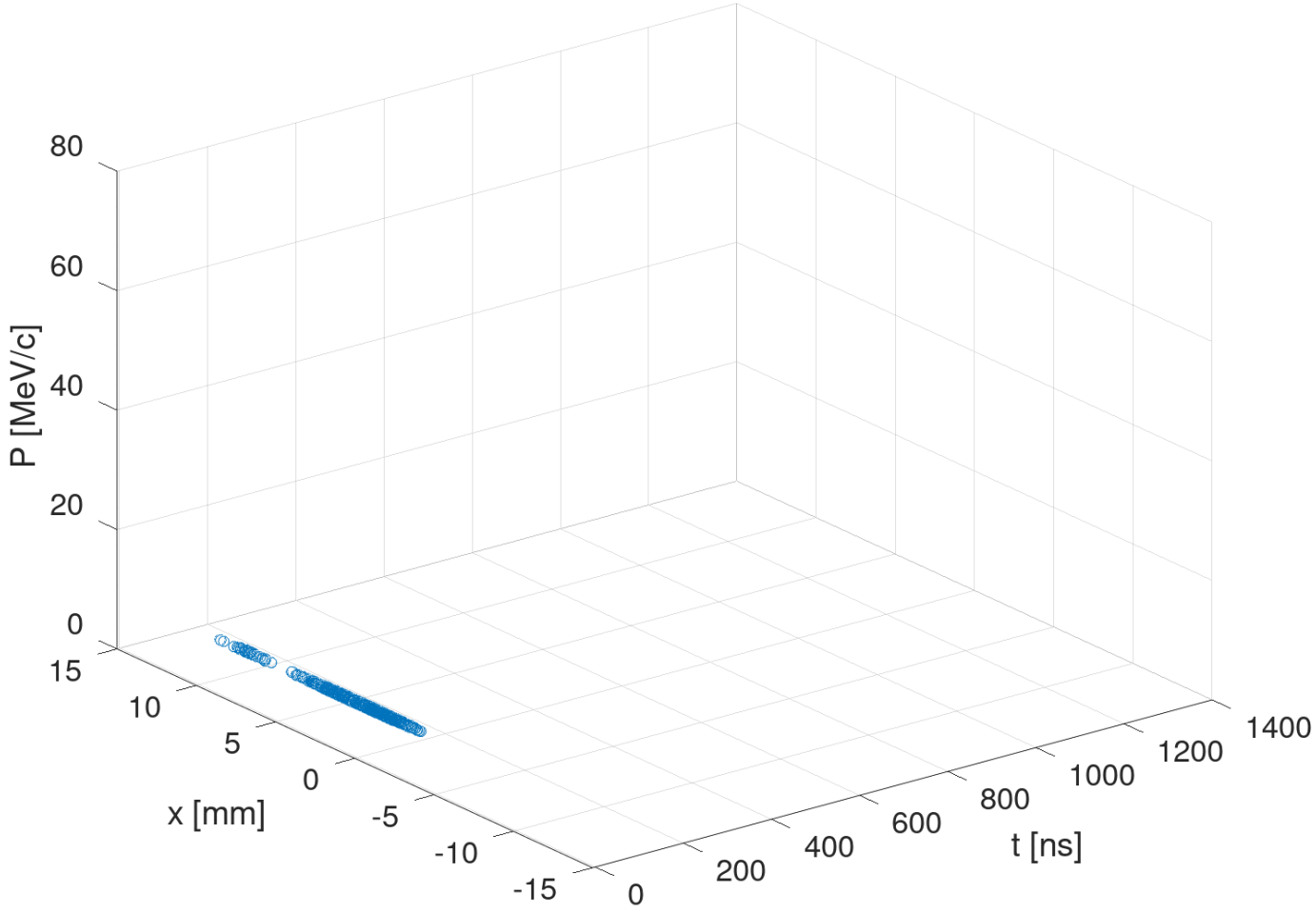
RFQ simulation – SC FFT



Transverse phase-space evolution



Longitudinal phase-space evolution



Longitudinal phase-space

- Final distribution