Karlsruhe Institute of Technology

Challenges for longitudinal beam dynamics simulations at cSTART

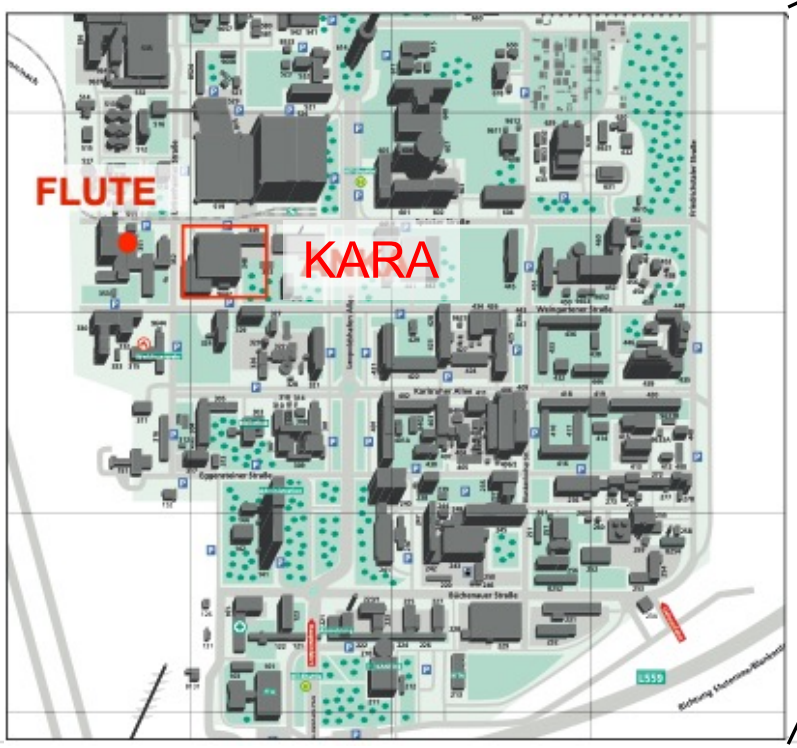
Markus Schwarz
Acknowledgement: cSTART team

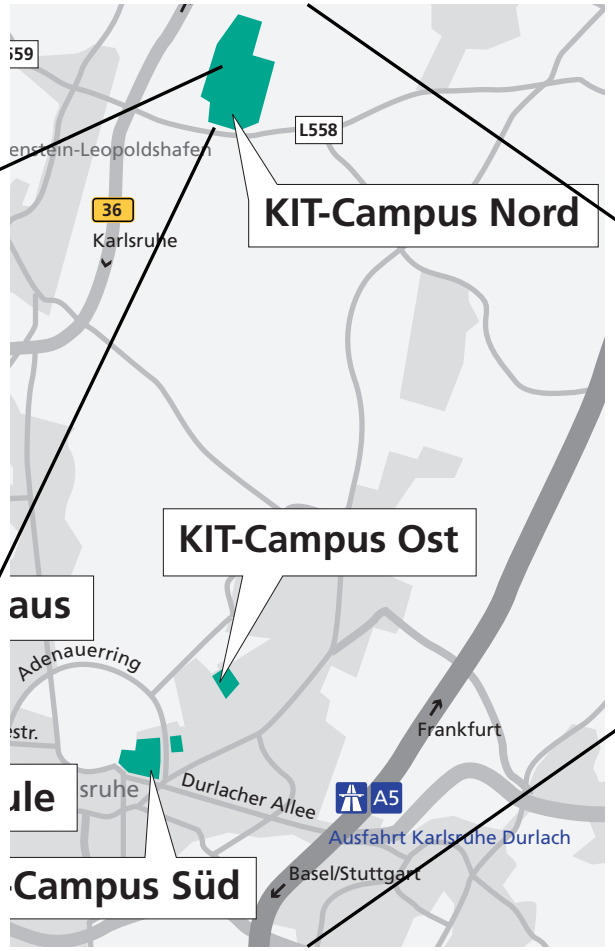
Outline

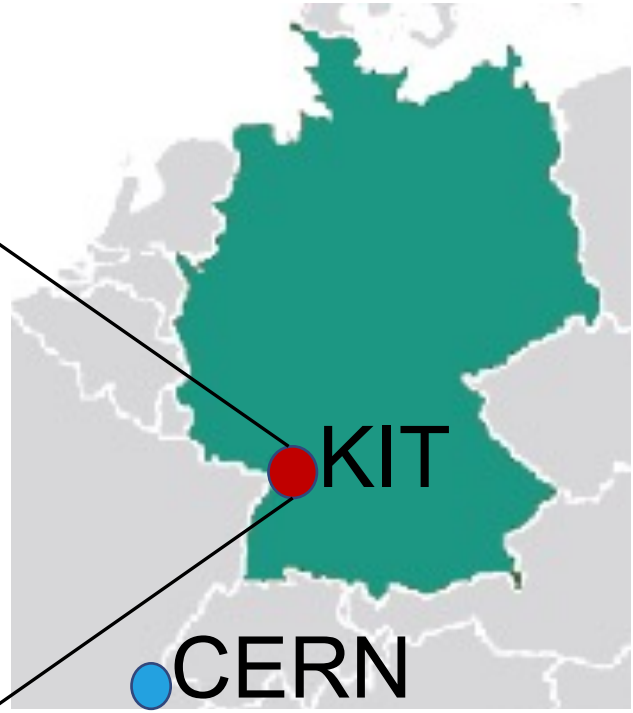
- cSTART and its beam dynamic challenges
- Solution I
- Solution II
- Example

Karlsruhe Institute of Technology

- University (22,300 Students)
- National Research Center
  - 9800 Employees

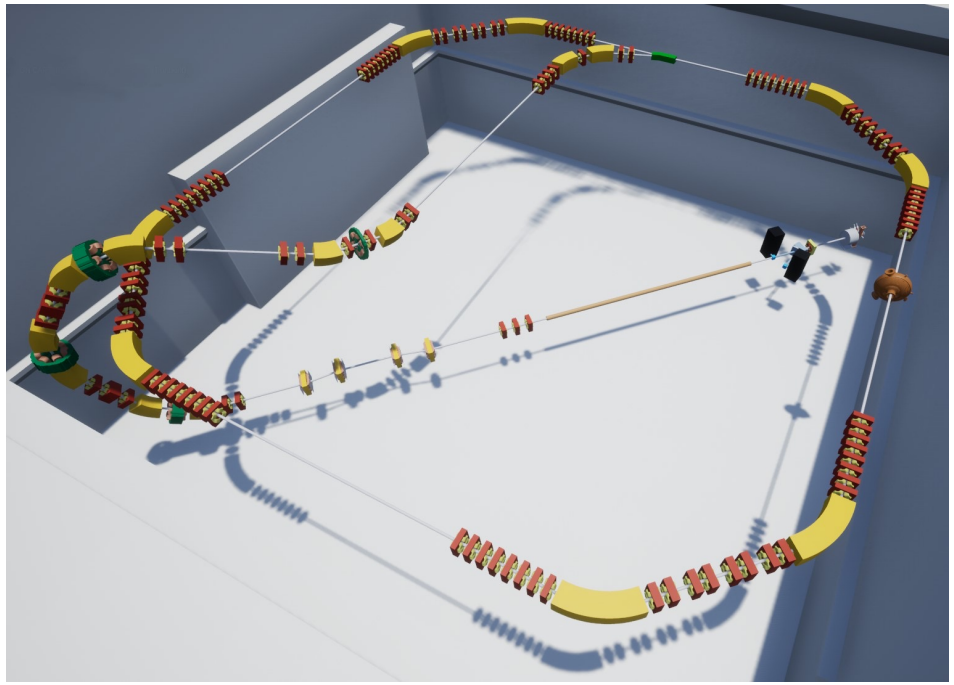






The cSTART Project

- compact Storage Ring for Accelerator Research and Technology
- Very-large acceptance storage ring to **inject LPA-like** and **store sub-ps short** electron bunches
- Two injectors
  - linac-based FLUTE
  - LPA [S. Hillenbrand et al., NIM A **740**, 2014]



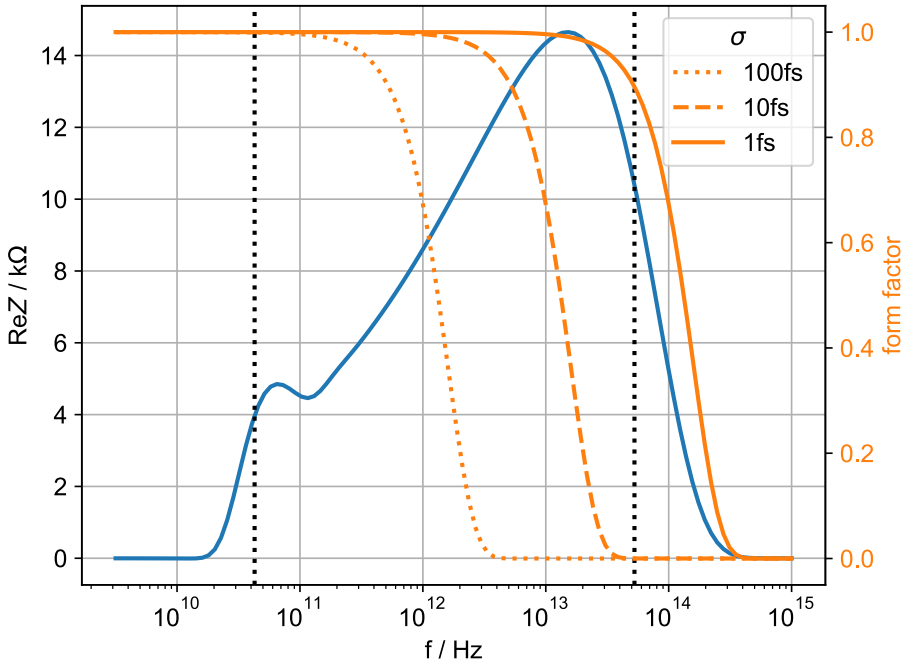
cSTART storage ring parameters

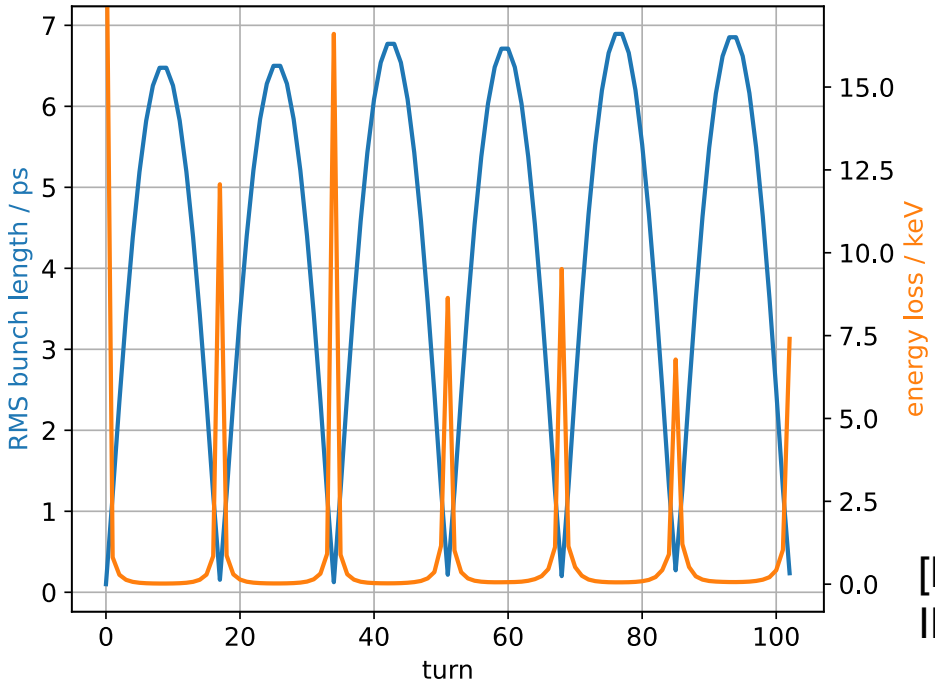
Lattice	
Circumference	43.2 m
Revolution period; freq	144 ns; 6.9 MHz
Injection type	On-axis one-turn; single bunch
Injection rate	10 Hz
Magnetic rigidity	0.133 Tm to 0.200 Tm
Incoherent energy loss	< 0.5 eV per turn
Coherent energy loss	Up to 200 keV per turn
Damping time	37 s / 33 s / 16 s

Beam parameter	
Nominal energy	50 MeV
Energy range (no ramping)	40 MeV to 60 MeV
Bunch length (duration)	3 µm to 300 µm (10 fs to 1 ps)
Bunch charge (# electrons)	1 pC to 1 nC (7e6 to 7e9)
RF System	
Frequency; harmonic number	500 MHz; 72
# RF stations	1
RF Voltage	500 kV

(Some) Beam dynamics challenges

- Bunch length varies between 10 fs to 10 ps
  - Profile needs to cover **~100 ps with ~1 fs resolution**
- Always in **non-equilibrium** state → How to cross-check simulation?





[M. Schwarz et al., IPAC'21 TUPAB255]

Free-space SR and Gaussian bunch

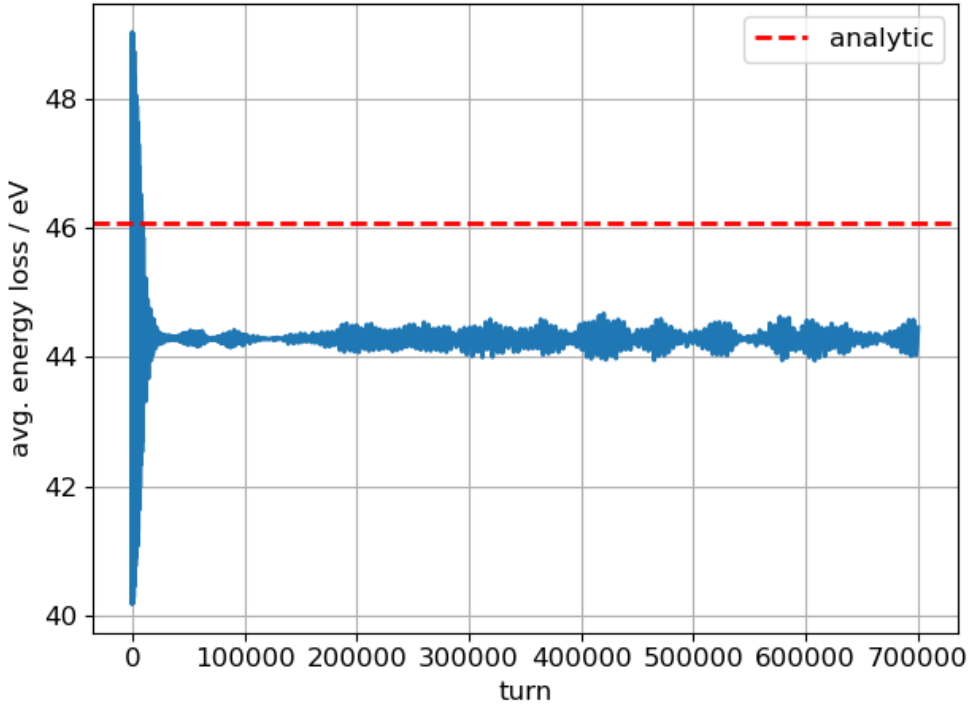
- Cross-checking simulation
  - Simplification I: free-space, $f \ll f_c \Rightarrow$ impedance $Z(f) \propto f^{1/3}$
  - Simplification II: Gaussian bunch
  - $\Rightarrow$ Analytical expression for average energy loss:

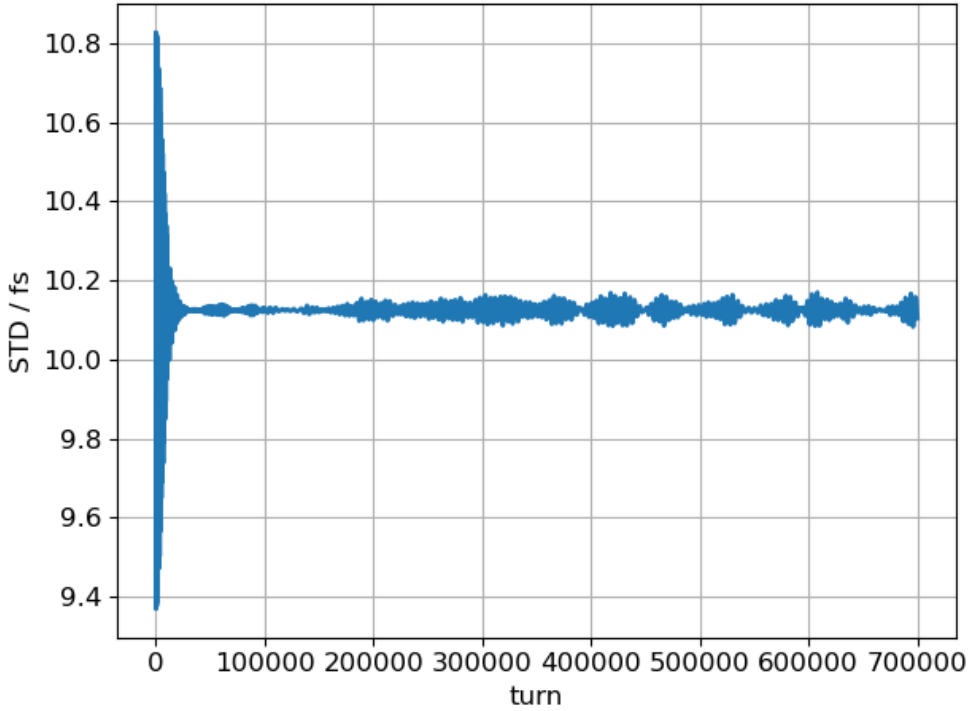
$$\bar{U} = \frac{3^{1/6}\Gamma^2(2/3)}{4\,2^{1/3}\pi^{4/3}} \frac{Z_0\, e^2 N_e^2}{(f_0\,\sigma^4)^{1/3}}$$

- Simplification II valid for first turn or low intensity beam

Example: 10 ps bunch

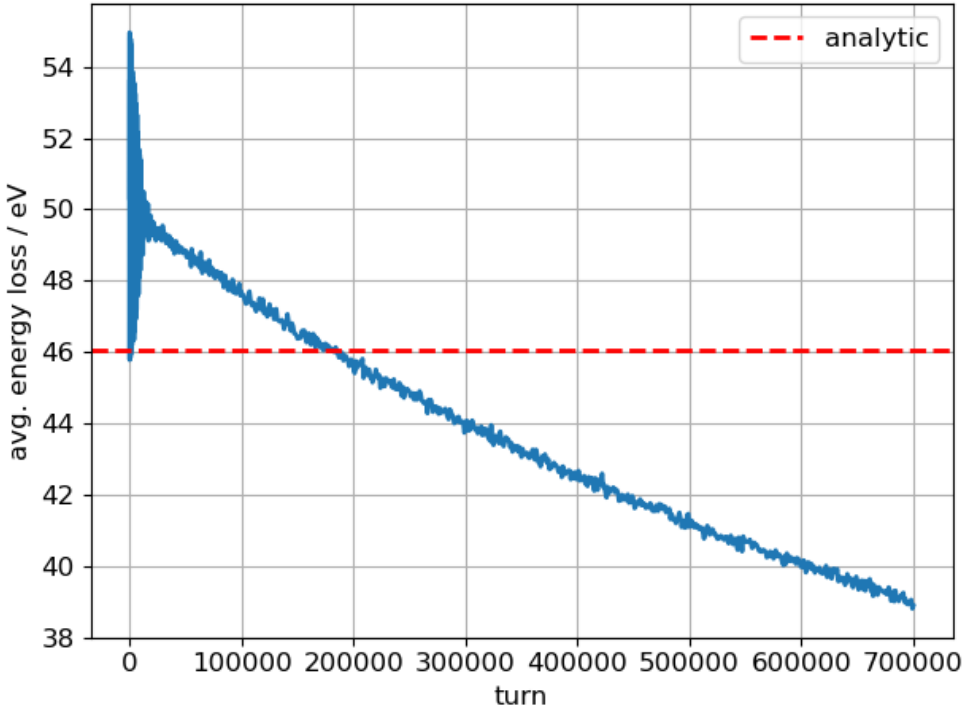
- Profile: **256 bins**, `Profile.bin_size` 1.88 ps ⇒ reasonable agreement
- But: need fine sampling for 10 fs bunch, `Profile.bin_size` ~ 1 fs…

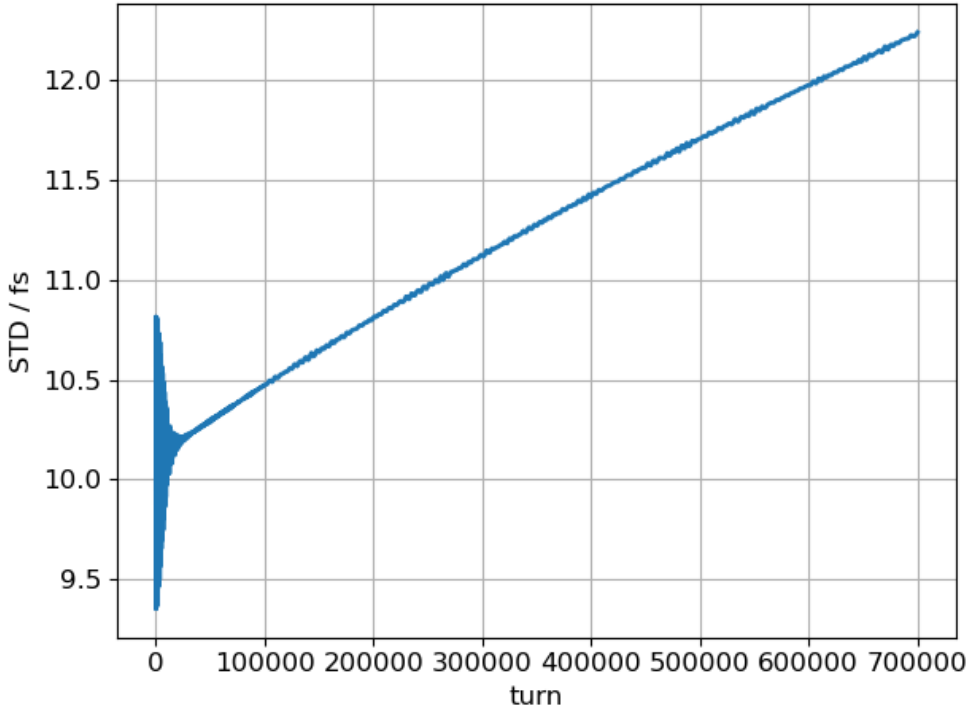




Example: 10 ps bunch

- Profile: **65,000 bins**, `Profile.bin_size` 4.5 fs
  ⇒ **spurious emittance growth** (noise amplification at large frequencies?)





Outline

- cSTART and its beam dynamic challenges
- **Solution I: variable bin size**
- Solution II
- Example

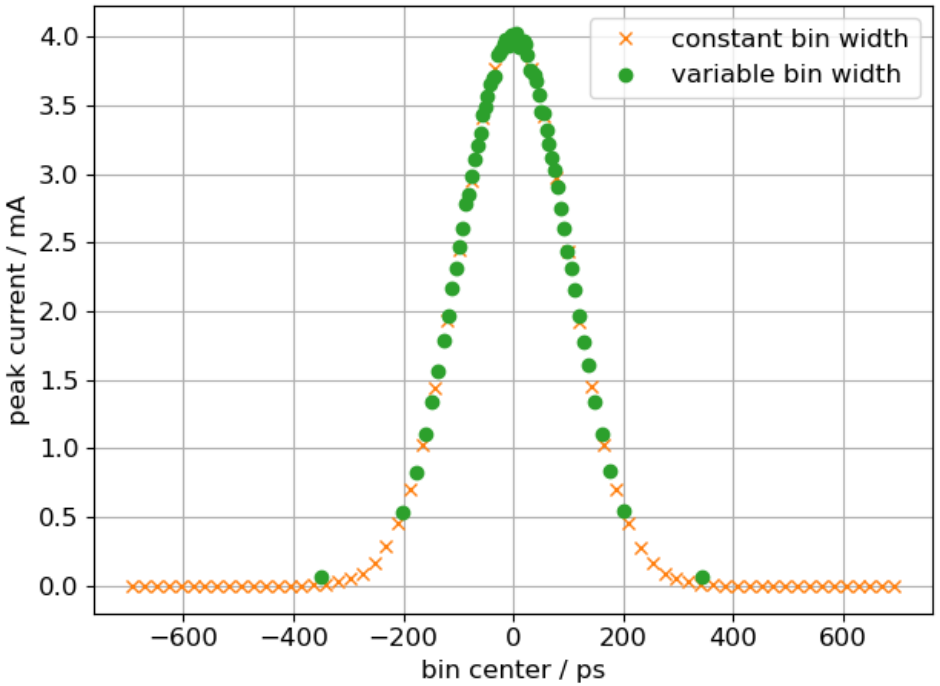
Solution I: variable bin size

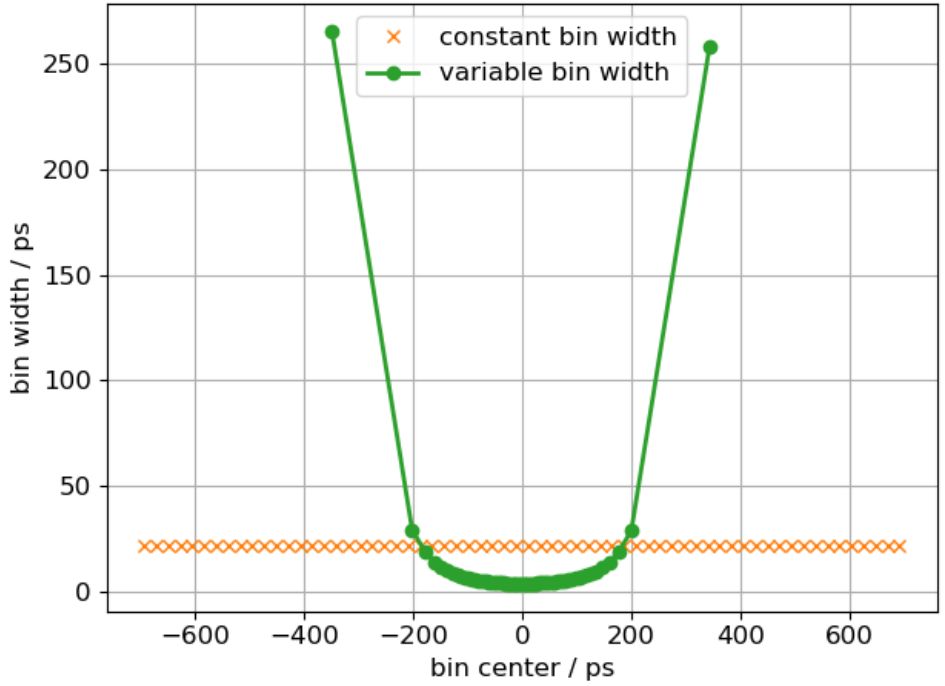
- Adjust bin size to have equal # of MP per bin
- **q-quantiles** divide probability density in areas of equal probability (ex: 2-quantile = median)
- Use `scipy.stats.mstats.mquantiles` to compute empirical quantiles for `beam.dt`

```
for tun in range(ring.n_turns):

    bin_edges = stats.mstats.mquantiles(non_beam.dt, np.linspace(0,1,n_edges))
    bin_edges[-1] *= 1+1e-6  # slightly increase last edge, so digitize works correctly

    rf_voltage = bm.rf_volt_comp(tracker.voltage, tracker.omega_rf, tracker.phi_rf, bin_edges)

    nonuni_linear_interp_kick(beam.dt, beam.dE, rf_voltage, bin_edges, beam.Particle.charge, 0)

    tracker.drift(tracker.beam.dt, tracker.beam.dE, turn+1)
```

Solution I: profile

- 100 ps Gaussian bunch; profiles with 64 bins each
- Small bin width near bunch center; large bin width at tails





Solution I: RF voltage kick

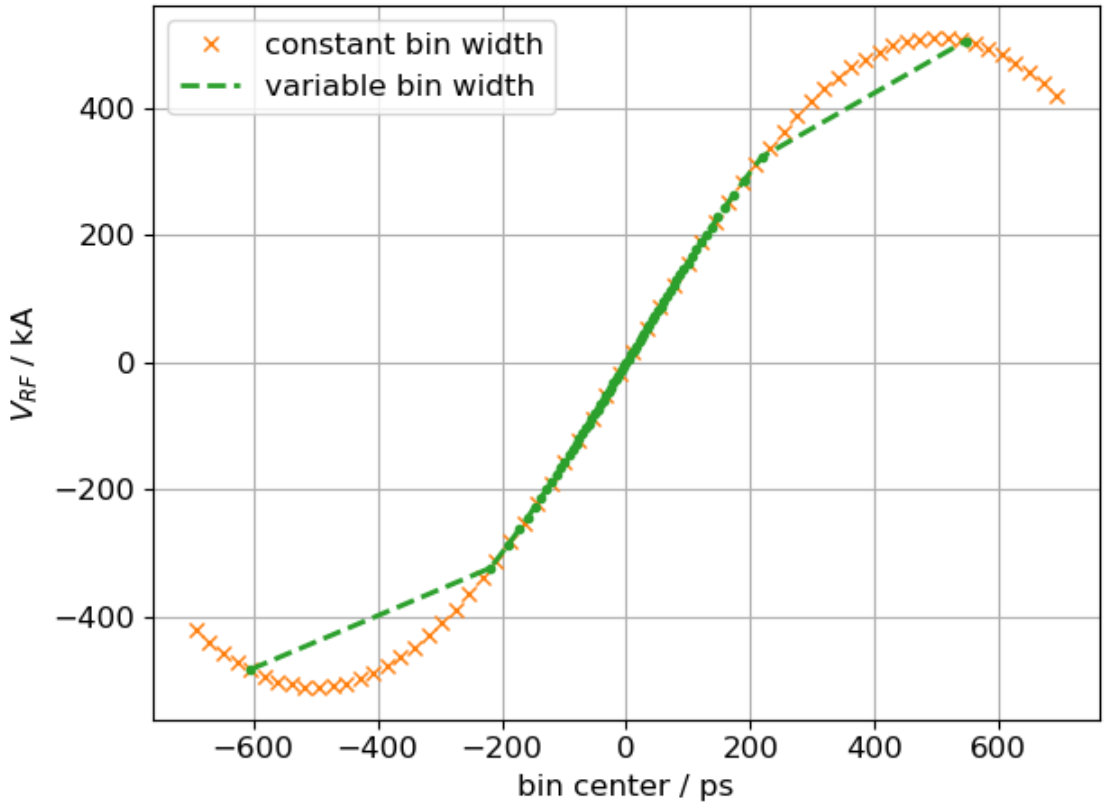
```
nonuni_linear_interp_kick(beam.dt, beam.dE, rf_voltage, bin_edges, beam.Particle.charge, 0)
```

- Same idea as `linear_interp_kick()`, i.e. approximate voltage by linear segments
- Need `np.digitize()` (quick-sort algorithm) to assign bin to particle
  ⇒ time consuming

```
def nonuni_linear_interp_kick(beam_dt, beam_dE, voltage_array, bin_edges,
                              charge, acc_kick):

    slope =  charge * np.diff(voltage_array) / np.diff(bin_edges)
    intercept = (charge * voltage_array[:-1] - bin_edges[:-1] * slope) + acc_kick

    fbin = np.digitize(beam_dt, bin_edges) - 1

    beam_dE += beam_dt * slope[fbin] + intercept[fbin]
```

Solution I: RF voltage kick

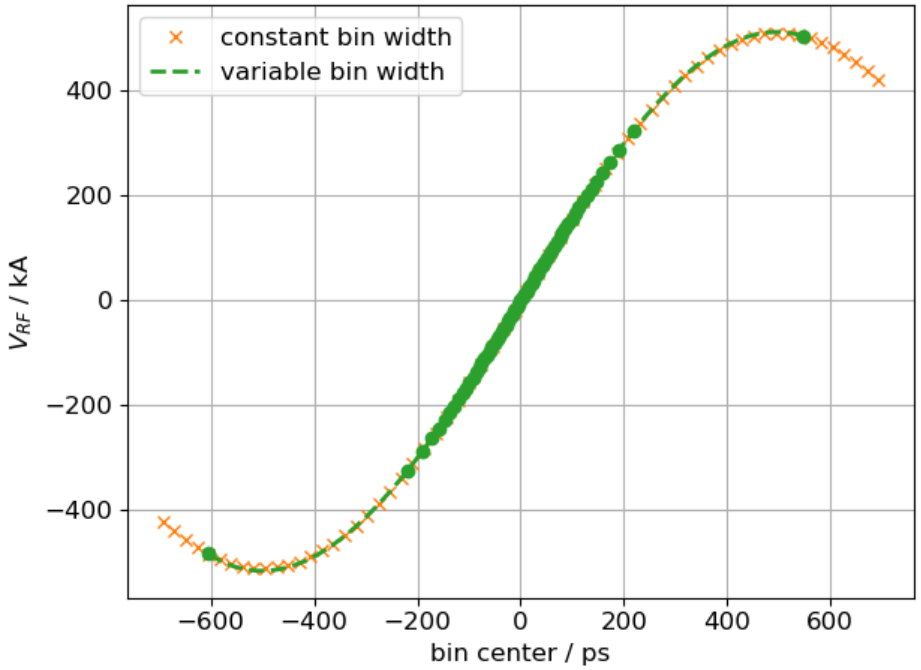
- Linear approximation **insufficient** near edge of RF bucket



Solution I: RF voltage kick

- Use **cubic** interpolation

```
def nonuni_nonlinear_interp_kick(beam_dt, beam_dE,
                                 voltage_array, bin_edges,
                                 charge, acc_kick, kind='cubic'):

    v_interp = interpolate.interp1d(bin_edges, charge * voltage_array,
                                    kind=kind, assume_sorted=True)

    beam_dE += v_interp(beam_dt) + acc_kick
```



- Result:
  - **comparable results** to constant bin-width profile
  - Tracking **100 times slower** even without intensity effects :(

Outline

- cSTART and its beam dynamic challenges
- Solution I: variable bin size
- **Solution II: variable profile cuts**
- Example

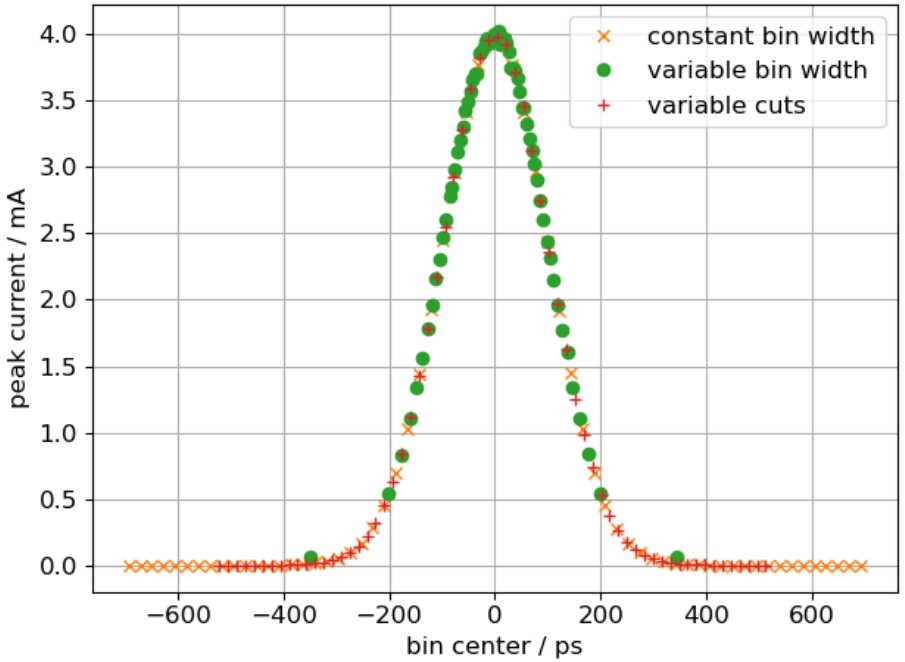
Solution II: Variable profile cuts

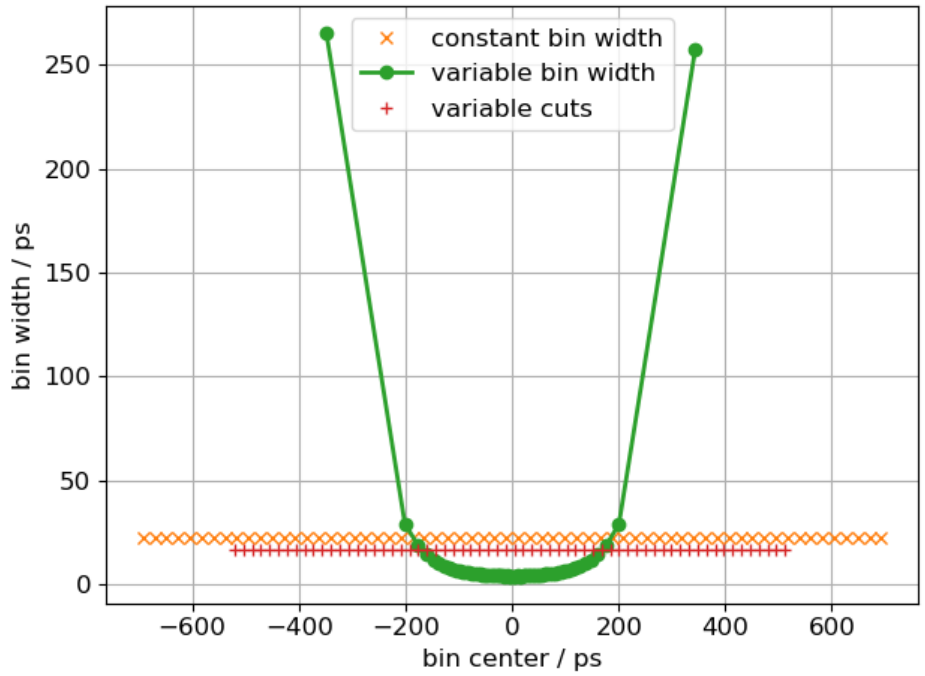
- Standard `Profile` object, but **adjust profile cuts** to `min(beam.dt)` and `max(beam.dt)` each turn
- ⇒ equidistant bins, but bin size varies each turn
- Use existing BLonD ‘infrastructure’, e.g. `Profile.set_slices_parameters()`

```
for turn in range(ring.n_turns):
    dt_max, dt_min = np.max(beam.dt), np.min(beam.dt)
    profile.cutOptions.cut_left = dt_min - 0.05 * (dt_max - dt_min)
    profile.cutOptions.cut_right = dt_max + 0.05 * (dt_max - dt_min)
    profile.CutOptions.set_cuts()
    profile.set_slices_parameters()

    profile.track()

    tracker.track()
```

Solution II: Variable profile cuts

- Result:
  - comparable results to standard constant bins profile
  - Tracking 4 times slower, but **readily extents to intensity effects**





Solution II: Variable profile cuts with intensity effects

- Include intensity effects with `TotalInducedVoltage` object as usual
- Use `TotalInducedVoltage.reprocess()` to apply new profile cuts

```
for turn in range(ring.n_turns):
    dt_max, dt_min = np.max(beam.dt), np.min(beam.dt)
    profile.cutOptions.cut_left = dt_min - 0.05 * (dt_max - dt_min)
    profile.cutOptions.cut_right = dt_max + 0.05 * (dt_max - dt_min)
    profile.CutOptions.set_cuts()
    profile.set_slices_parameters()

    profile.track()

    inducedVoltage.reprocess()  # time array not updated
    inducedVoltage.induced_voltage_sum()

    tracker.track()
```

- Needed to change `_InducedVoltage.process()` when passing `frequency_resolution`

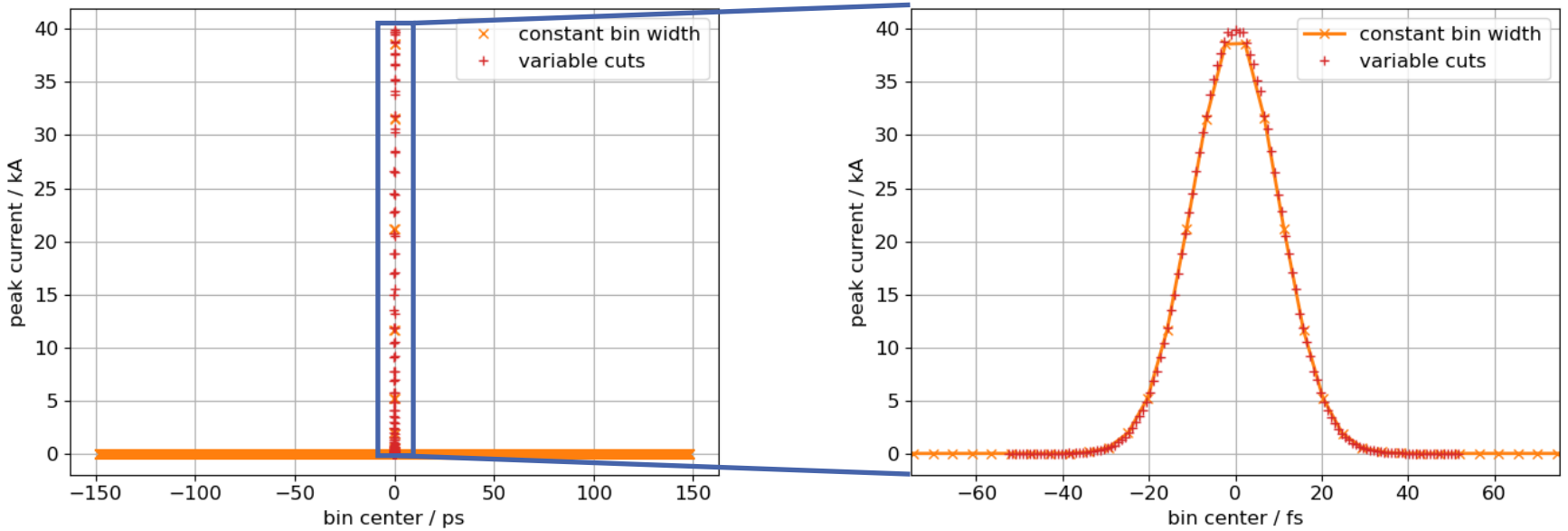
```
if self.n_induced_voltage < self.profile.n_slices:
    # FrequencyResolutionError -> adjust to default values
    self.wake_length = (self.profile.cut_right -
                        self.profile.cut_left)
    self.n_induced_voltage = self.profile.n_slices
    # raise RuntimeError('Error: too large frequency_resolution.
    #                    'Reduce it below {0:1.2e} Hz.'.format(1
```

Outline

- cSTART and its beam dynamic challenges
- Solution (attempt) I: variable bin size
- Solution II: variable profile cuts
- **Example: 10 fs Gaussian bunch**

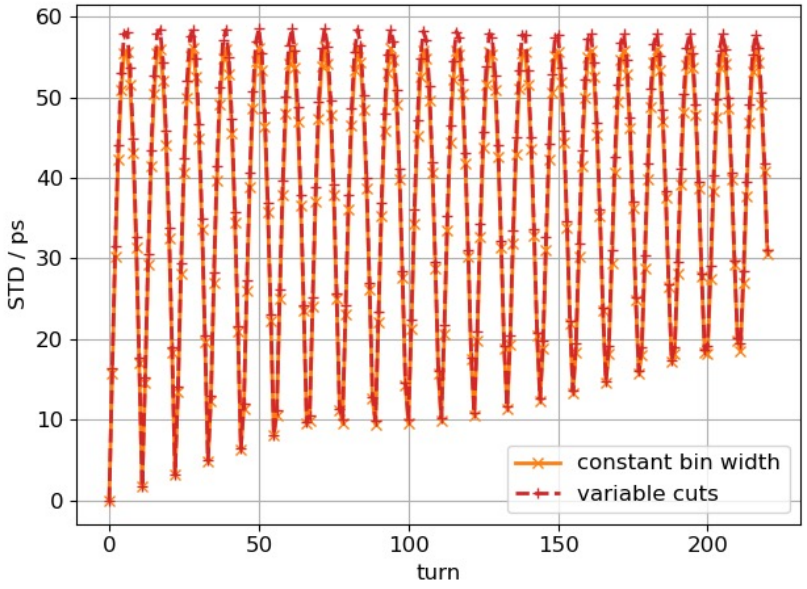
Example: 10 fs Gaussian bunch

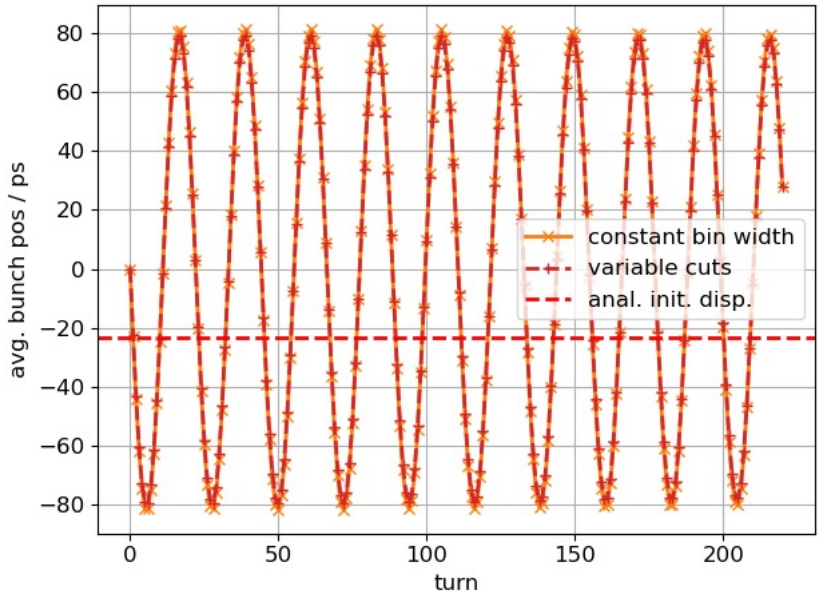
- ‘Standard `Profile`’
  - Δt ~ 4.5 fs to sample initial bunch
  - Cuts from -150 to 150 ps to ‘cover’ all MP when bunch is large ⇒ **65,000 bins**
- Profile with variable cuts needs **128 bins**

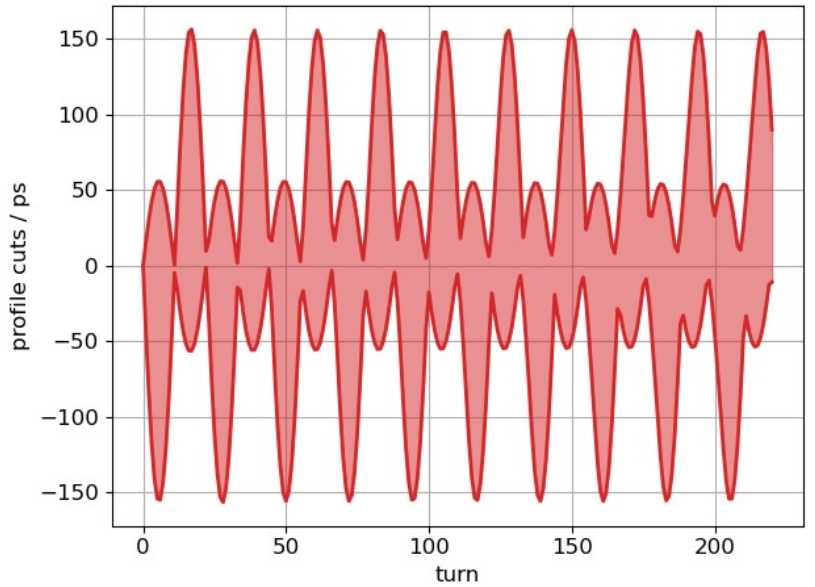


Example: 10 fs Gaussian bunch

- Good agreement between methods (for first 200 turns)
- **No spurious emittance blow-up** with variable profile cuts
- Tracking with variable profile cuts **4 times faster** :)

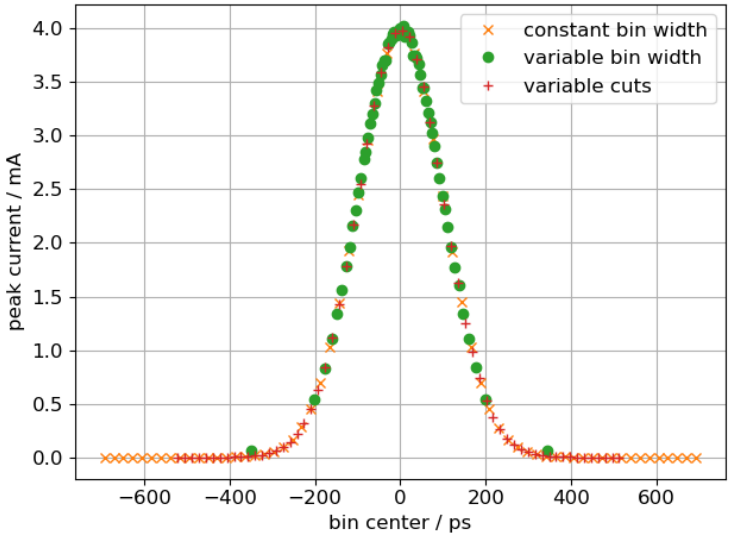


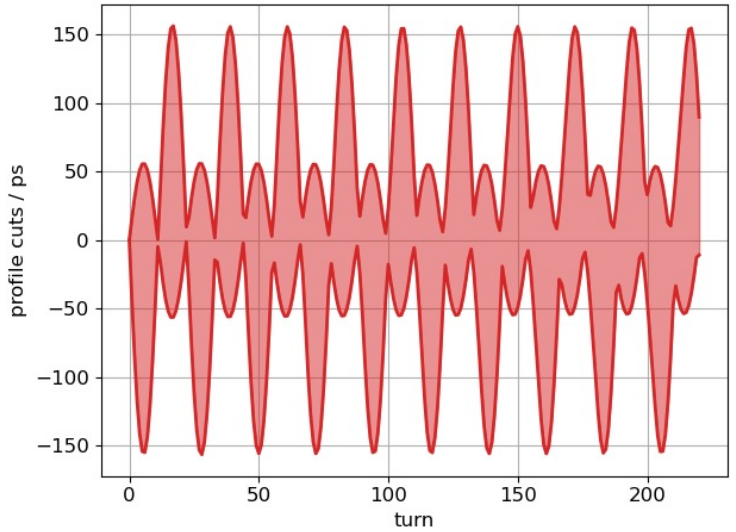


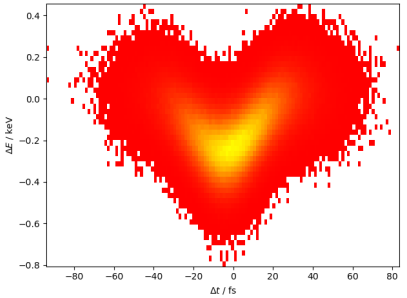


Summary

- cSTART longitudinal beam dynamics simulations challenging: **large bunch length (>10 ps) with fs resolution**
- Standard `Profile` leads to spurious emittance growth
- Profile with variable bin size possible, but computationally very expansive
- Profile with **variable profile cuts 'works' and 4 times faster**







Thank you for your attention!

Appendix

Simulating long Damping Times

- τ_z=16.7 s = 112 million turns
- Simulate with BLonD

  [www.blond.cern.ch]
- Simulation remains stable for 300 million turns and reproduces expected result