Integrated beam beam modeling with Xsuite

G. Iadarola, **R. De Maria, S. Łopaciuk,**
A. Abramov, X. Buffat, D. Demetriadou, L. Deniau, P. Hermes, P. Kicsiny, P. Kruyt, A. Latina,
L. Mether, K. Paraschou, J. Salvesen, G. Sterbini, F. Van Der Veken, CERN, Geneva, Switzerland

P. Belanger, TRIUMF, Vancouver, Canada

D. Di Croce, T. Pieloni, L. Van Riesen-Haupt, M. Seidel, EPFL, Lausanne, Switzerland

P. Niedermayer, GSI, Darmstadt, Germany

Work supported by: CHART EPFL

https://xsuite.web.cern.ch

Outline

- **Introduction**
- **Architecture and main capabilities**
  - Lattice modeling
  - Single-particle tracking
  - Twiss module
  - Optimized
  - Particle-matter interaction
  - Synchrotron radiation
  - Collective effects (wakefields, spacecharge, IBS)
- **Beam-beam capabilities**
  - Weak-strong
  - Strong-strong (soft-Gaussian)
  - Beamstrahlung and Bhabha effect
  - Strong-strong (Particle In Cell)
- **Summary and final remarks**

Xsuite

Xsuite project was launched in **2021**

- **Main goal:** bring into a **modern Python toolkit** the know-how built at CERN in developing MAD, Sixtrack, COMBI, PyHEADTAIL
  - **Cover with one toolkit** applications ranging from **low-energy hadron rings** to **high-energy lepton colliders**
- Designed for **seamless integration of components** and for **extendibility**
- Support **different computing platforms**, including **multicore CPUs and GPUs** from different vendors

Design constraints:

- Need **to grow the code in a "sustainable" way**, being managed and maintained by a **small core team** integrating (in a clean way!) contributions by a wide developer community
- Need **user and developer learning curve** to be short as possible

  → Field specific features developed **directly by field experts**

The Xsuite community

After three years the software has grown very rapidly, thanks **to many people contributing code and expertise…**

and more…

The Xsuite community

After three years the software has grown very rapidly, thanks **to many people contributing code and expertise…**

Response went well beyond our expectation:

- **>30 colleagues from CERN and other labs contributed by developing new features and debugging issues**
  - Leveraging their python skills and the short tool-specific learning curve
- Xsuite was **adopted by a large and diverse user community** (>100 users!!!)
  - Very **lively community** providing mutual support, advice, lots of feedback to developers (very precious!)
  - For first time at CERN we are using the same software tool for **optics, dynamic aperture studies, collimation, beam-beam, space-charge, instabilities, lepton machines, extraction and beam transfer studies and more…**
    - Already **profited from lots of synergies**

and more…

Xsuite simulations have been already used for **studies covering a variety of rings**:

CERN
- ELENA
- LEIR
- PSB
- PS
- SPS, TI2, TI8
- LHC
- FCC-ee, FCC-hh
- Muon collider
- LEP

GSI
- SIS-18
- SIS-100

Medical facilities
- HIT (Heidelberg)
- MEDAUSTRON
- PIMMS
- NIMMS

BNL
- RHIC
- Booster
- EIC

Fermilab
- Main injector
- Recycler
- Booster
- IOTA

Light sources and damping rings:
- PETRA
- DESY injector ring
- ELETTRA
- BESSY III
- CLIC-DR

… and more

Each of these tought us something and contributed to extend and improve the software!

- Introduction
- **Architecture and main capabilities**
  - Lattice modeling
  - Single-particle tracking
  - Twiss module
  - Optimized
  - Particle-matter interaction
  - Synchrotron radiation
  - Collective effects (wakefields, spacecharge, IBS)
- Beam-beam capabilities
  - Weak-strong
  - Strong-strong (soft-Gaussian)
  - Beamstrahlung and Bhabha effect
  - Strong-strong (Particle In Cell)
- Summary and final remarks

Xsuite – architecture

Physics modules

Xpart
generation of particles distributions

Xtrack
single particle tracking engine

Xfields
computation of EM fields from particle ensembles

Xdeps
Dependency manager, deferred expressions

Xcoll
Particle-matter interaction and collimation

Xobjects
interface to different computing plaforms
(CPUs and GPUs of different vendors)

CFFI PyOpenCL CuPy

Lower level libraries
(external, open source)

intel AMD NVIDIA

Hardware

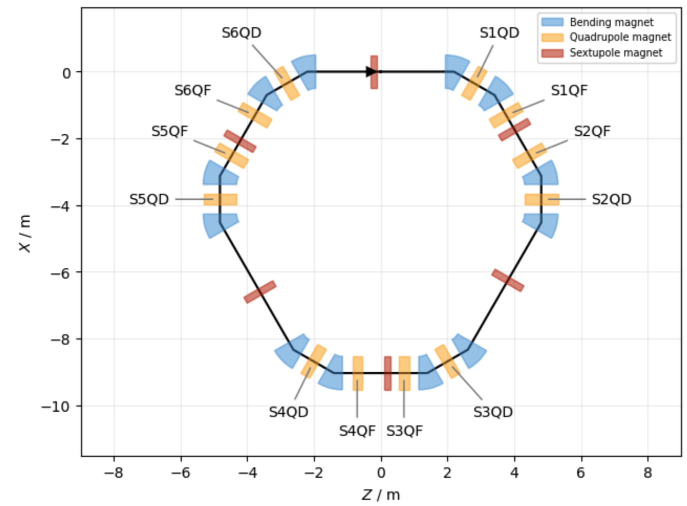
Lattice modelling

- The **beam line** is represented as a **sequence of Python objects**, each corresponding to an accelerator element or to other physical processes (e.g. magnets, cavities, aperture restrictions, etc.).
  - Can be **defined manually** or **imported from MAD-X**
    - Including **tilts**, **misalignments** and **multipolar errors**

Xsuite model of a ring
(represented with the Xplt package)

```
survey = tracker.survey()
```

```
plot = xplt.FloorPlot(survey, line, labels="S.Q.")
plot.legend()
```



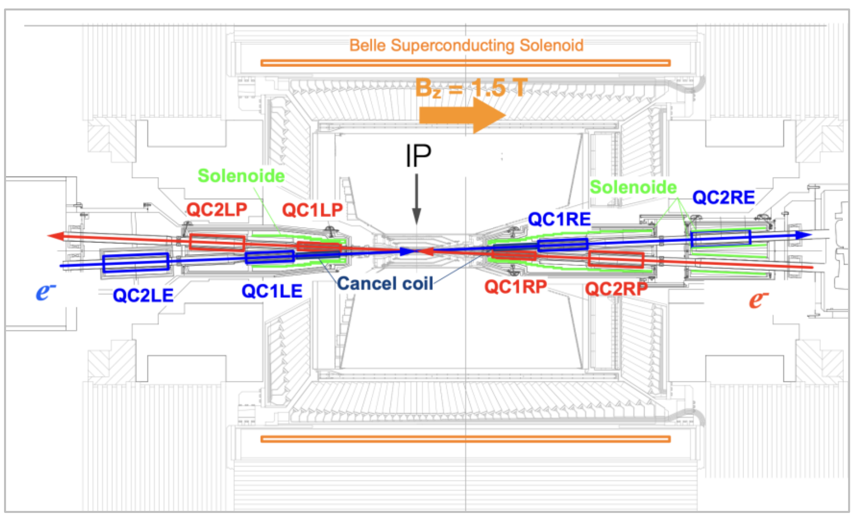
We provide:

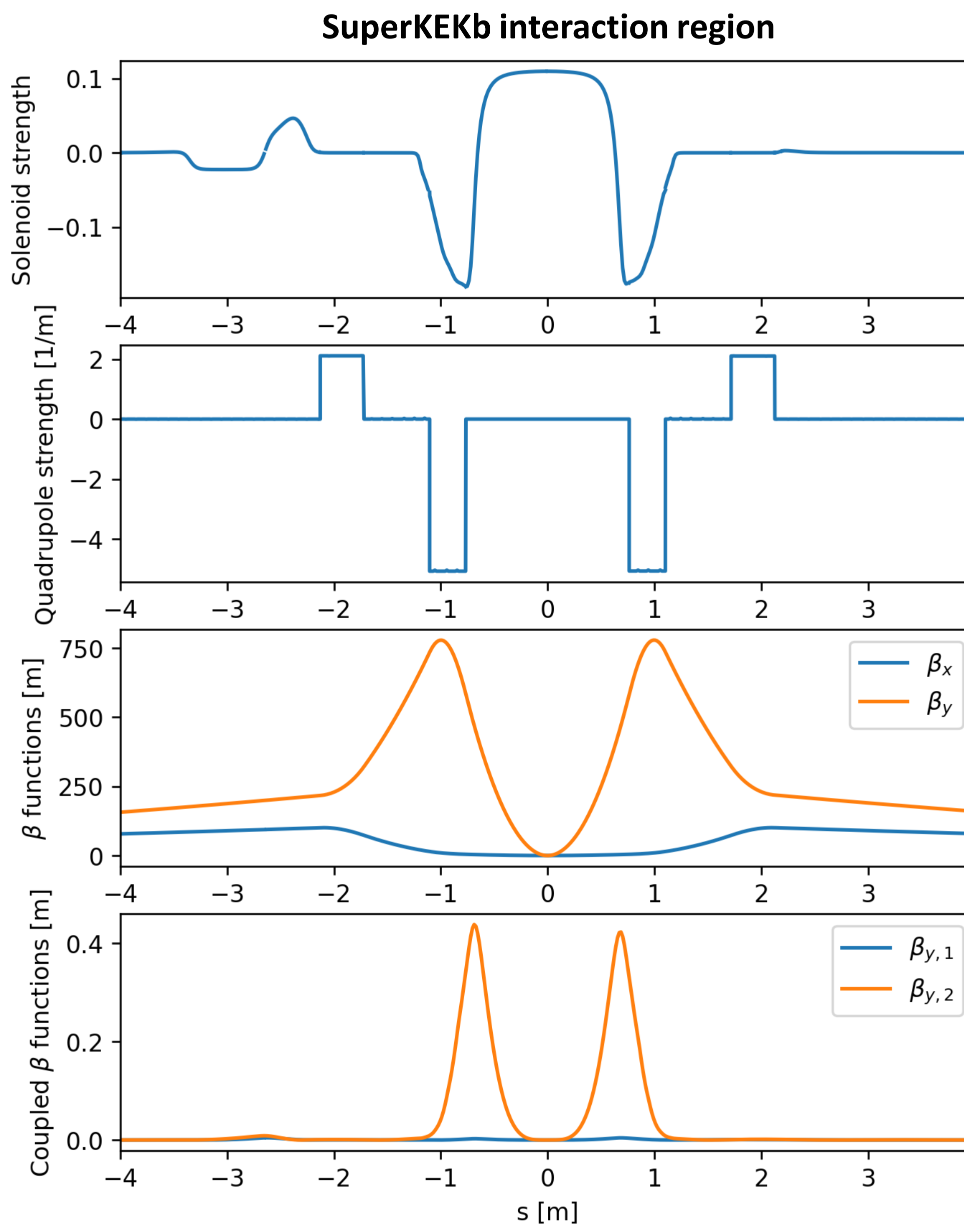
- **“Thin” lattice integration**, largely based on the Sixtrack and Sixtracklib experience
- **“Thick” maps** for bending and quadrupole magnets
- **Dipole edge effects** including **fringe fields** can be modeled either in their **linearized form** or as **full non-linear maps** (same fringe model as in MAD-NG and PTC).

Solenoids and overlapping elements

Recent developments allow modelling of **experimental solenoids** of lepton colliders also in the presence of **overlapping multipole fields**

- Tested on **FCC-ee** and **SuperKEKb** models





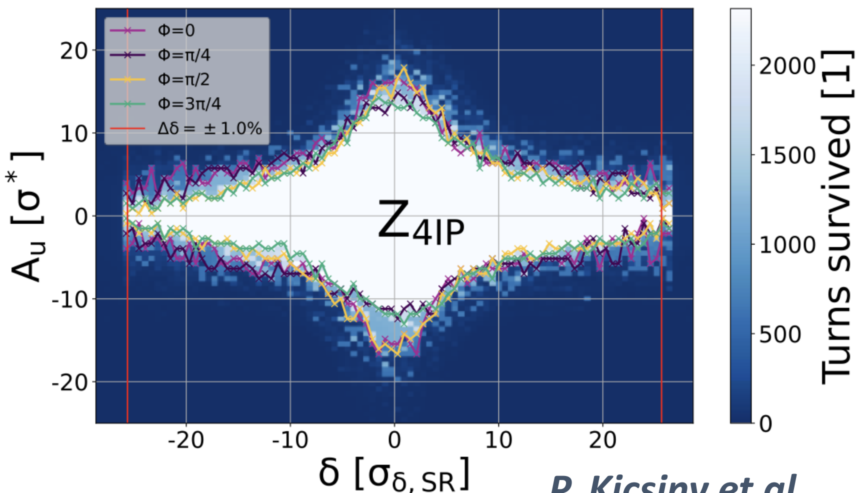
Many thanks to G. Broggi, J. P. Salvesen, KEK experts

- Multiturn **element-by-element tracking speed** is critical for several application
- To **speed up tracking simulations**, Xsuite assembles and compiles a **C kernel** (callable from Python) **optimized for the given beamline** and **specialized for the chosen platform** (CPU or GPU)
    - The **tracking speed** is found to be **similar to Sixtrack** for single-core CPU and **about two orders of magnitudes faster than that on high-end GPUs**

Tracking time for a typical LHC simulation

Platform	Computing time
CPU (single core)	190 (μs/part./turn)
GPU (NVIDIA V100, cupy)	0.80 (μs/part./turn)
GPU (NVIDIA V100 pyopencl)	0.85 (μs/part./turn)

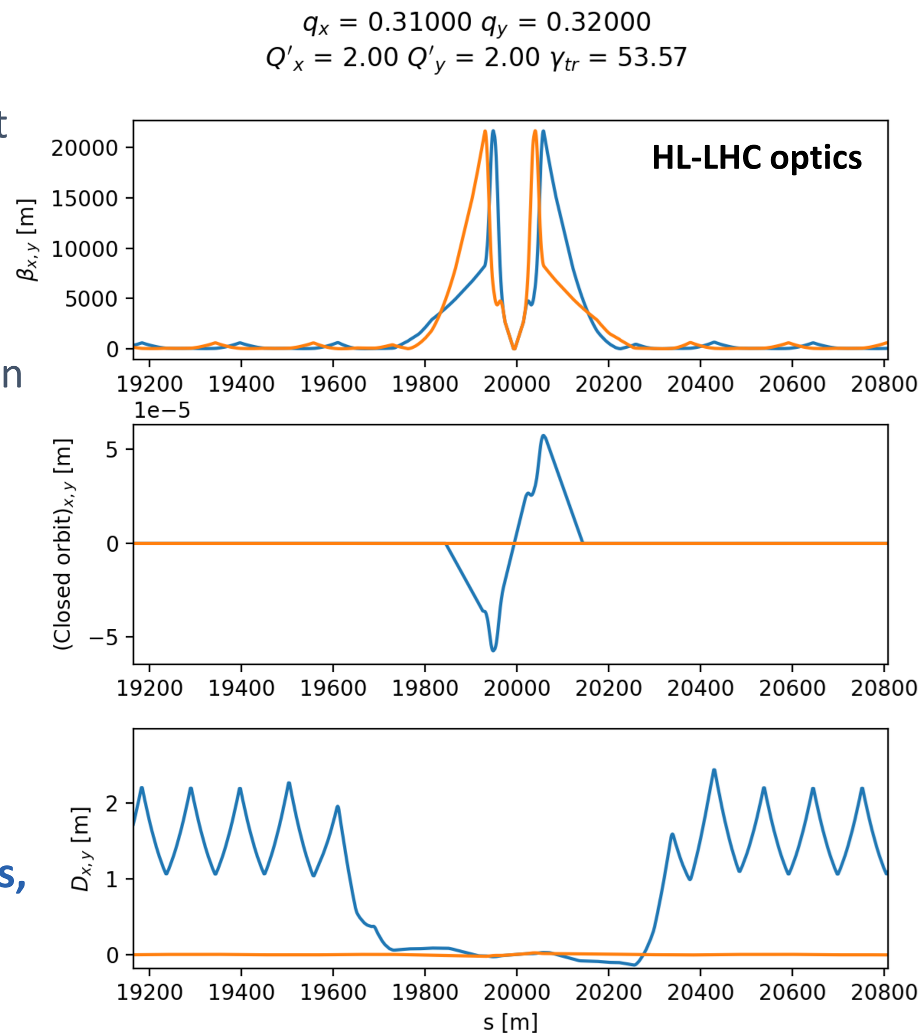
FCC-ee DA studies (with bb)



P. Kicsiny et al.

Twiss module

- The **Xsuite Twiss** module can be used to extract **lattice functions** of a ring or a beamline
- The calculation **probes the lattice simply by tracking particles**:
  - **Closed orbit** obtained by applying a Python root finder on the tracking
  - The **Jacobian** matrix obtained by **tracking (central differences)**
  - Compute **"Linear Normal Form"** of the Jacobian matrix (diagonalization)
  - **Propagate eigenvectors** by **tracking**
  - Obtain from the eigenvectors **Twiss parameters (α, β, γ) , dispersion functions, phase advances, coupling coefficients**
- Computation can be done with **assigned beam momentum** to get **off-momentum beta-beating, non-linear chromaticity, non-linear dispersion,** etc.



Accuracy compared to MAD-X: $\Delta\beta$ / β << 10^{-4}
Computation time is very similar

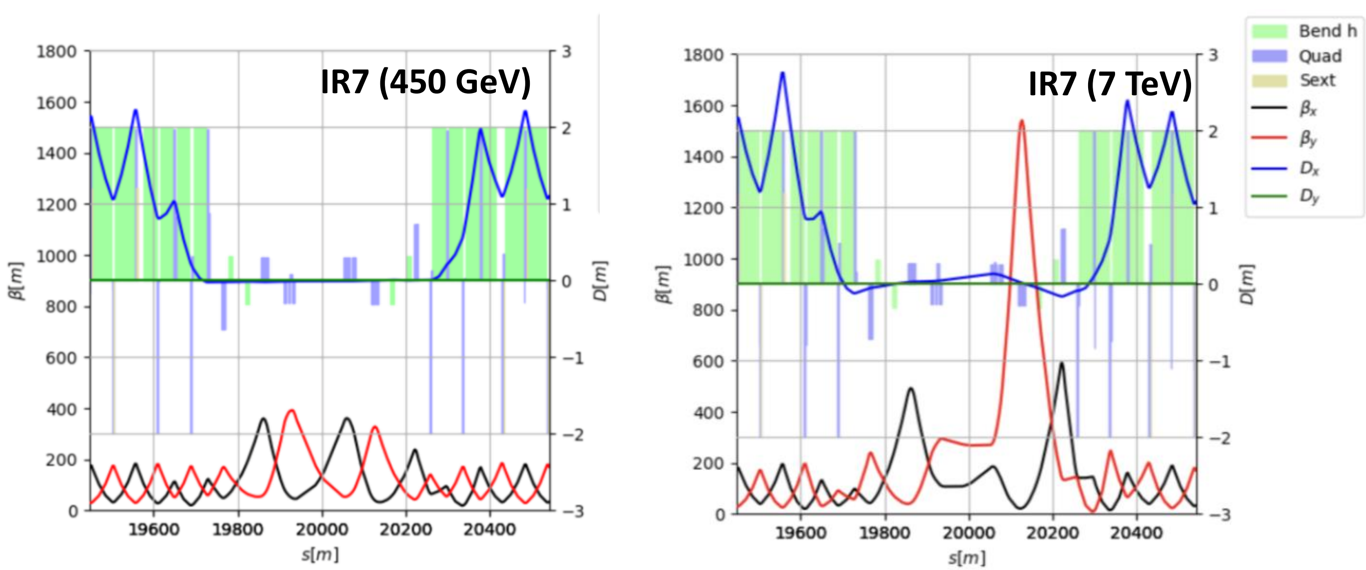
Example

```
In [37]: tw.bety[0] # xsuite
Out[37]: 149.4305507849305

In [38]: mad.table.twiss.bety[0] # madx
Out[38]: 149.43055000962505

In [39]: t_mad_ms
Out[39]: 202.0

In [40]: t_xsuite_ms
Out[40]: 185.0
```

Xsuite provides a **multi-objective optimizer** to "match" model parameters to assigned constraints (e.g. control tunes, chromaticity, build orbit bumps, design the optics)

- Based on the **extensive experience of MAD-X** → Uses the **same optimization algorithm** (Jacobian, proven robustness)
- Interface **designed for usage flexibility**. User can **intervene in the optimization** by:
  - Enabling/disabling targets or knobs, rolling back optimization steps, changing knob limits, target values, convergence tolerances
- Used for **optics matching of the LHC and of FCC-ee colliders**

First full cycle designed with Xsuite tested at the LHC in 2024
(including combined ramp&squeeze for all insertions)



Courtesy R. De Maria and B. Lindstrom

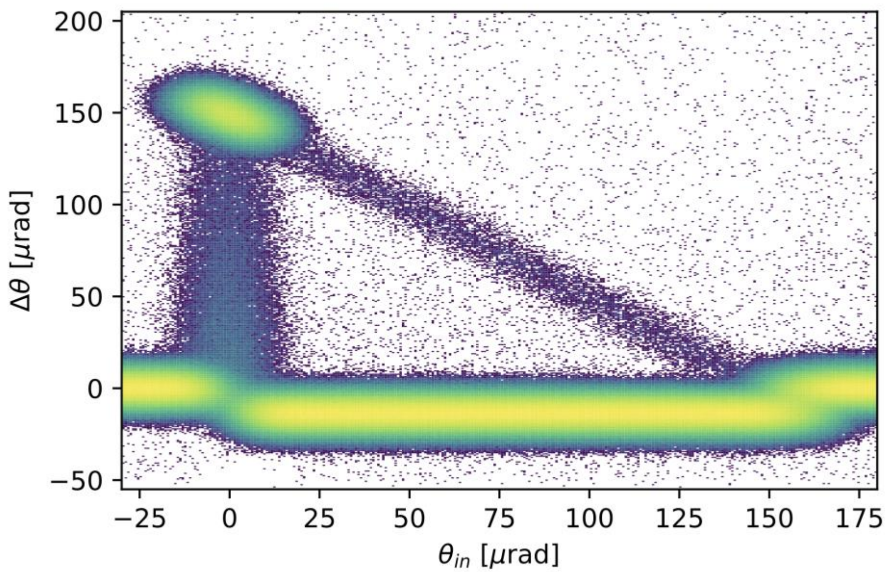
For collimation studies, the **Xcoll module** provides **three particle-matter sim. engines:**

- The **"Everest" engine** embedded in Xcoll (evolution of K2 module from Sixtrack)
- The **"Geant 4" engine**, based on an **interface with BDSIM-Geant4**
  - **Used for FCC-ee collimation studies** (see presentation by G. Broggi)
- The **"FLUKA" engine**, based on an interface with the **FLUKA** Monte Carlo code

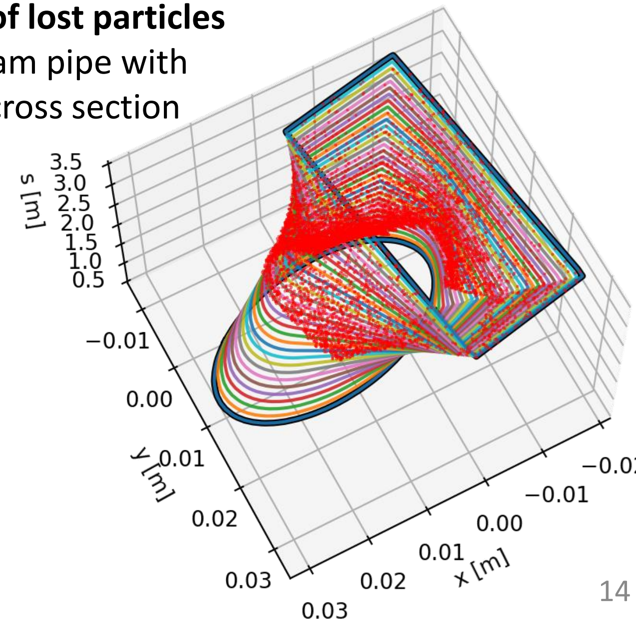
To support collimation studies, Xsuite provides:

- Tools to **automatically install and configure collimators** in the simulation model
- Support for **complex aperture modelling** and **accurate localization of the lost particles along** the beam line (typically within 1-10 cm)

Particle deflection from a bent crystal (Everest engine)

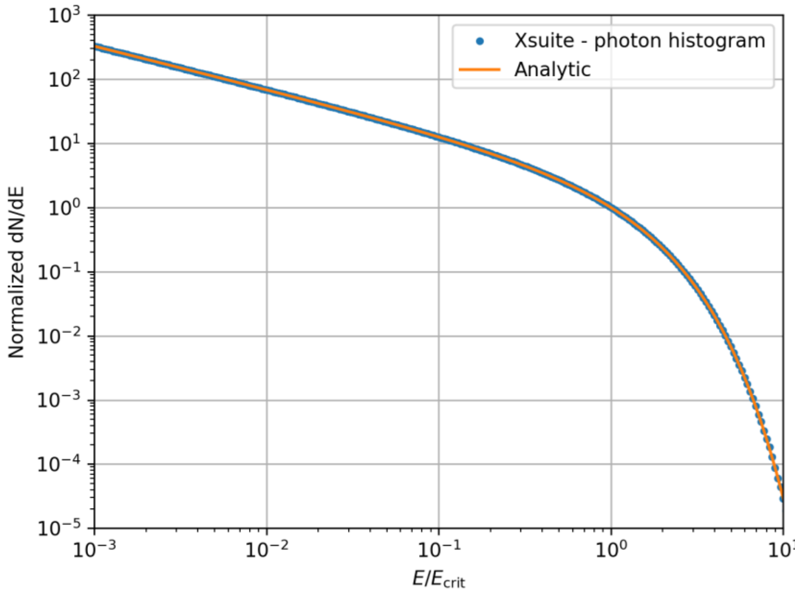


Localization of lost particles along a beam pipe with changing cross section



Supported by: CHART EPFL

Validation against analytical photon spectrum



The effect of **synchrotron radiation** can be included in Xsuite tracking simulations. Two models available:

- The **"mean" model**, for which the energy loss from the radiation is applied particle by particle without accounting for quantum fluctuations;
- The **"quantum" model** for which the actual photon emission is simulated[1].

[1] *Based on H. Burkhardt, "Monte Carlo generator for synchrotron radiation", 1990. Implementation ported from PLACET (A. Latina)*
[2] *E. Forest, From tracking code to analysis: generalised Courant-Snyder theory for any accelerator model. Springer, 2016*

Synchrotron radiation

Supported by: CHART EPFL

The effect of **synchrotron radiation** can be included in Xsuite tracking simulations. Two models available:

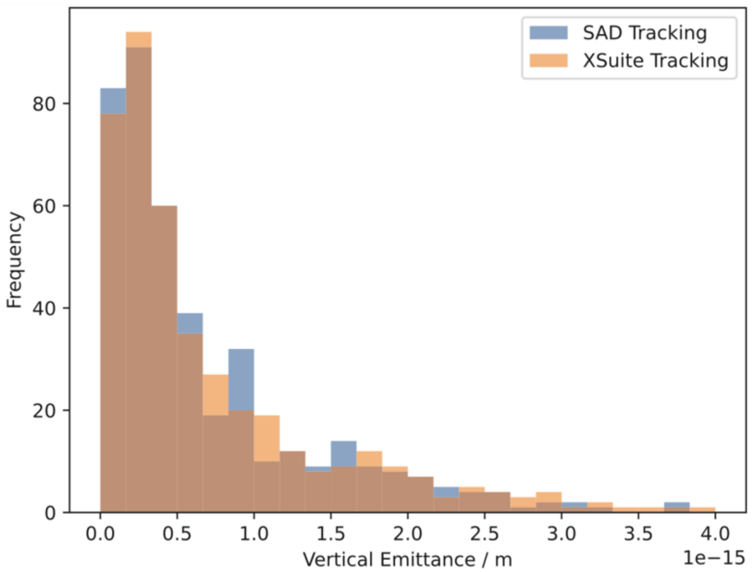
- The **"mean" model**, for which the energy loss from the radiation is applied particle by particle without accounting for quantum fluctuations;
- The **"quantum" model** for which the actual photon emission is simulated[1].

The **Xsuite Twiss** also includes:

- Dedicated algorithm for **non-symplectic one-turn map**[2]
- Computation of **radiation energy loss, damping times and equilibrium emittances**

An **automatic tool** is provided for **phasing the RF cavities** and **adjusting magnet strengths** to **compensate the radiation energy loss ("tapering")**

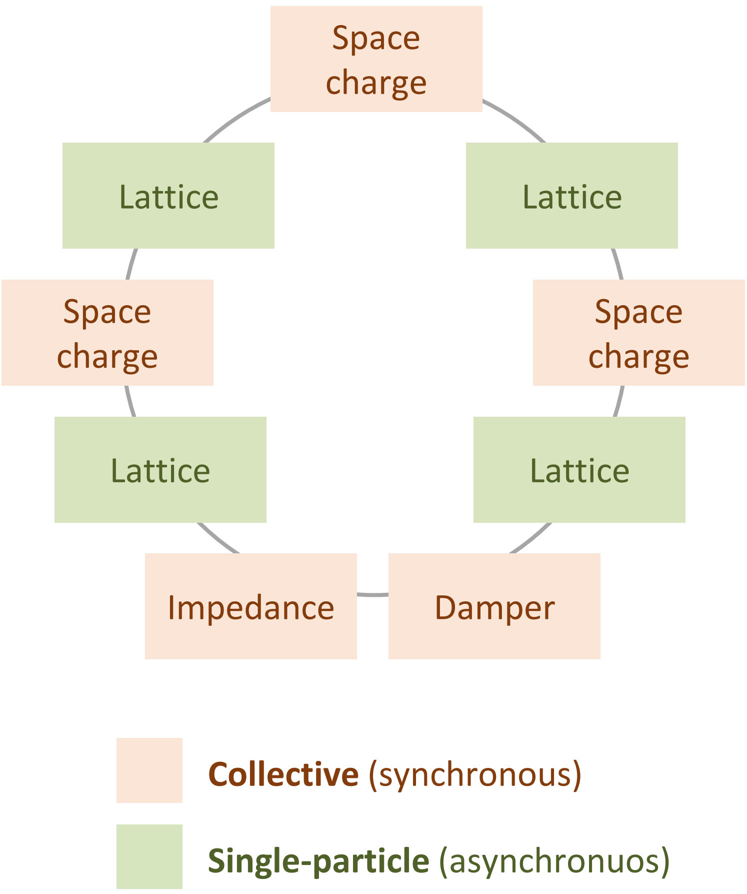
Benchmark of equilibrium emittaces from tracking (with lattice errors)



L. Van van Riesen-Haupt , T. Pieloni, et al., EPFL

[1] *Based on H. Burkhardt, "Monte Carlo generator for synchrotron radiation", 1990. Implementation ported from PLACET (A. Latina)*
[2] *E. Forest, From tracking code to analysis: generalised Courant-Snyder theory for any accelerator model. Springer, 2016*

Supported by: CHART EPFL



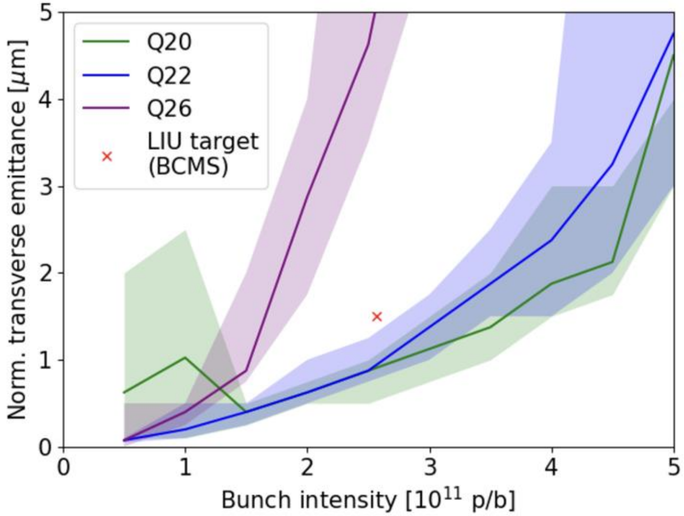
- Xsuite is designed to include **collective effects** in the simulations
- Handling of collective elements is **fully automatic** → The Xtrack module identifies the collective elements and **splits the sequence**:
  - The **non-collective** parts are handled **asynchronously** to **gain speed**
  - The simulation of the **collective effects** is **performed synchronously**
- **Space-charge**, **beam-beam**, **IBS**, **e-cloud** (weak-strong) are handled **natively**
- **Impedances** and **feedback systems** are handled through an interface with **PyHEADTAIL**
  - Native implementation coming soon

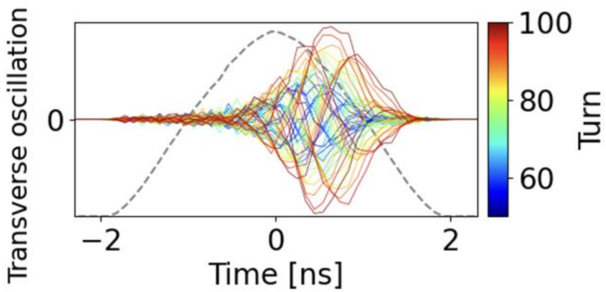
Space charge

Different **space-charge models** are implemented:

- The **"frozen" model**, in which particles interact with fixed charge distributions
- The **"quasi frozen"** model that is a variant of the frozen model in which the **beam intensity and beam sizes are recomputed at each interaction**
- The **"Particle In Cell (PIC)"** model:
  - Charge of tracked particles distributed on a **rectangular grid**
  - **Fast Poisson solver** based on **FFT** method with Integrated Green Functions
- Space charge simulations **strongly profiting from GPU acceleration**

Simulation campaign for the CERN SPS including full non-linear lattice, space charge and wakefields

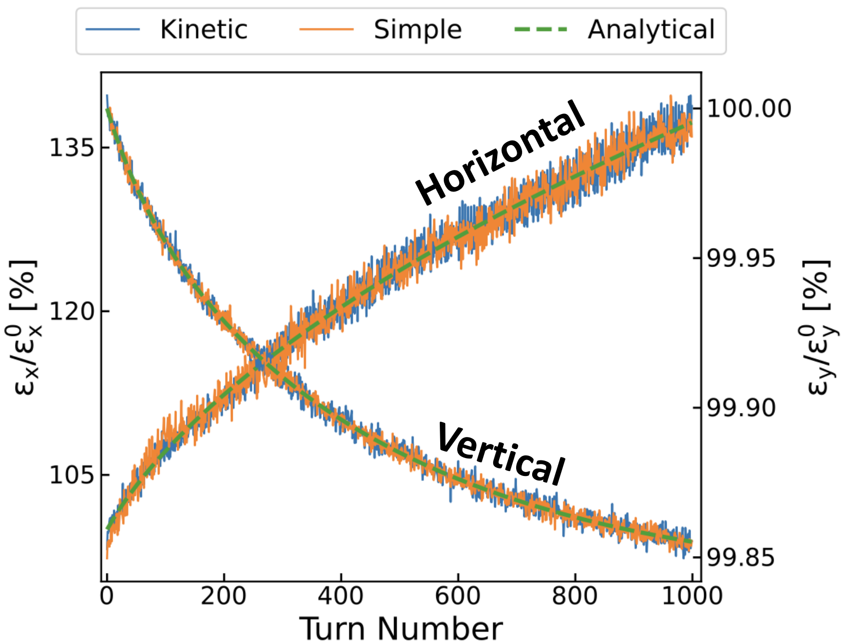




N. simulations	400
Number of PIC calculations per turn	540
Number of turns per simulation	40'000
Computing time per sim. (GPU)	~3 days
Computing time per sim. (CPU serial)	> 12 months

Courtesy X. Buffat

Benchmark case for the SPS ring (Pb ions)



Intra Beam Scattering (IBS) simulation capabilities have been recently introduced:

- **IBS growth rates computation** from beam parameters and optics. Two methods available:
  - Nagaitsev (very fast, vertical dispersion neglected)
  - Bjorken-Mtingwa (slower, D_y correctly accounted)
- Effect of **IBS can be included in multiparticle simulations** in combination with all other effects available in Xsuite. Two methods available:
  - Effective kick
  - Kinetic formalism

For more info: F. Soubelet et al., "Development of numerical tools for intra-beam scattering modelling", IPAC24

Outline

- **Introduction**
- **Architecture and main capabilities**
  - Lattice modeling
  - Single-particle tracking
  - Twiss module
  - Optimized
  - Particle-matter interaction
  - Synchrotron radiation
  - Collective effects (wakefields, spacecharge, IBS)
- **Beam-beam capabilities**
  - Weak-strong
  - Strong-strong (soft-Gaussian)
  - Beamstrahlung and Bhabha effect
  - Strong-strong (Particle In Cell)
- **Summary and final remarks**

Beam-beam capabilities in Xsuite

Xsuite offers **different capabilities for the simulation of beam-beam effects in colliders** that have become the workhorse for for LHC and FCC studies

- **Weak-strong** model assuming Gaussian Transverse Beam Profile
  - 4D interaction (lumped transverse-only kick)
  - 6D interaction (Hirata approach, energy change)
- **Strong-strong modelling**
  - 4D or 6D “soft Gaussian” approach
  - Self-consistent Particle In Cell

GPU acceleration available for all models

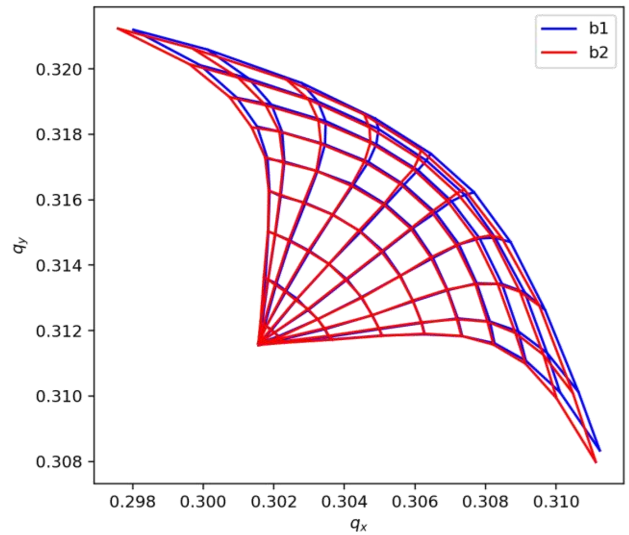
Beam-beam capabilities in Xsuite

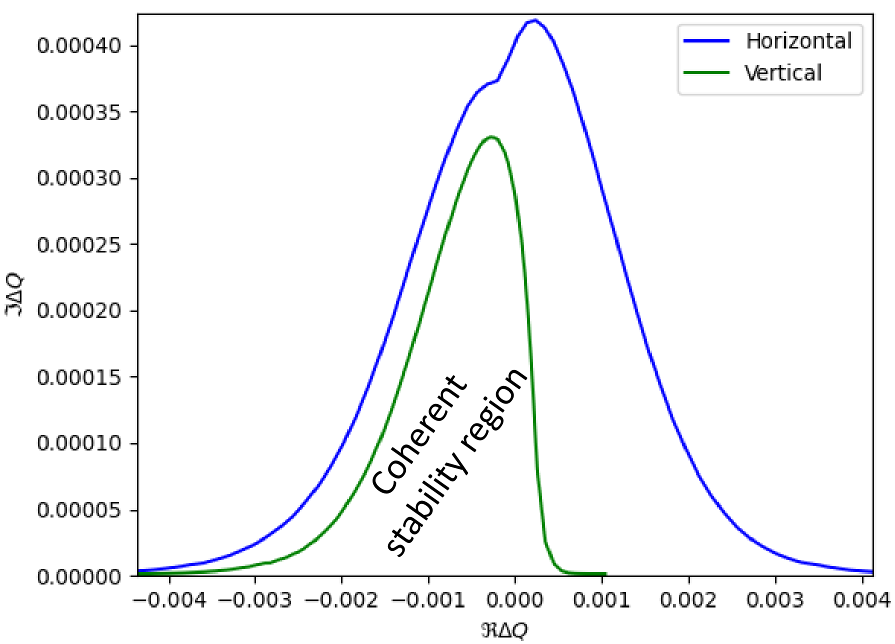
Weak-strong approach used extensively to study effect of beam-beam non linearities on single-particle dynamics:

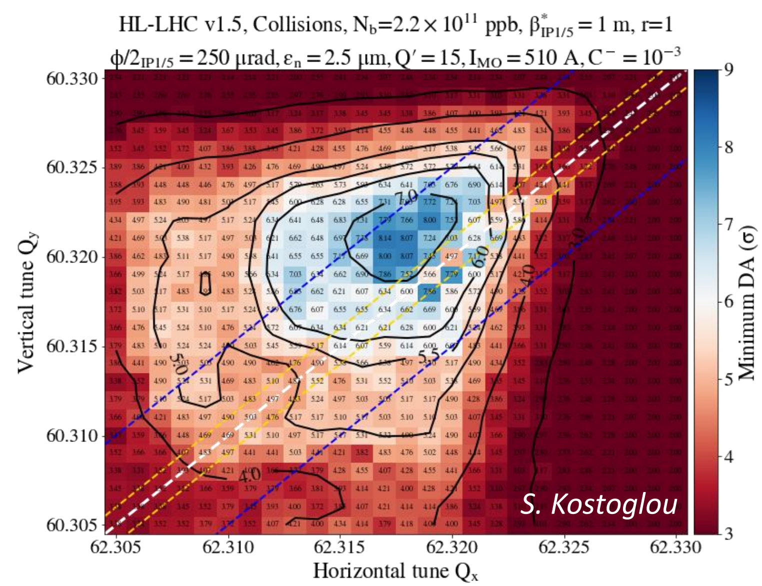
- **Amplitude detuning**

  → effect on beam stability, (stability diagram computation)

- Impact of beam-beam on **Dynamic Aperture (DA)**
- Direct simulation on **beam lifetime and emittance growth**







The simulation can be **easily configured from from the bare Xsuite collider model**.

- Required beam-beam interactions are **installed and configured** using survey, orbit and optics information computed by the **Xsuite optics engine**.
- Effect of **crab cavities** is taken into account

```
collider.install_beambeam_interactions(
    clockwise_line='lhcb1', anticlockwise_line='lhcb2',
    harmonic_number=35640, bunch_spacing_buckets=10,
    ip_names=['ip1', 'ip5'], delay_at_ips_slots=[0, 0],
    num_long_range_encounters_per_side=[5, 5],
    num_slices_head_on=11, sigmaz=0.1)
```

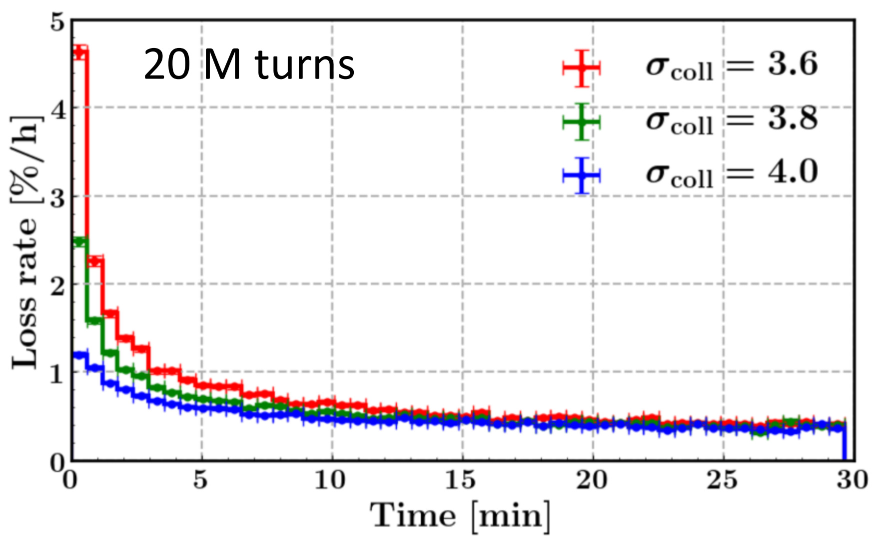
… arbitrary machine config (optics, crossing angles, etc.) …

```
collider.configure_beambeam_interactions(
                  crab_strong_beam=False,
                  num_particles=2.3e11,
                  nemitt_x=2e-6, nemitt_y=2e-6)
```

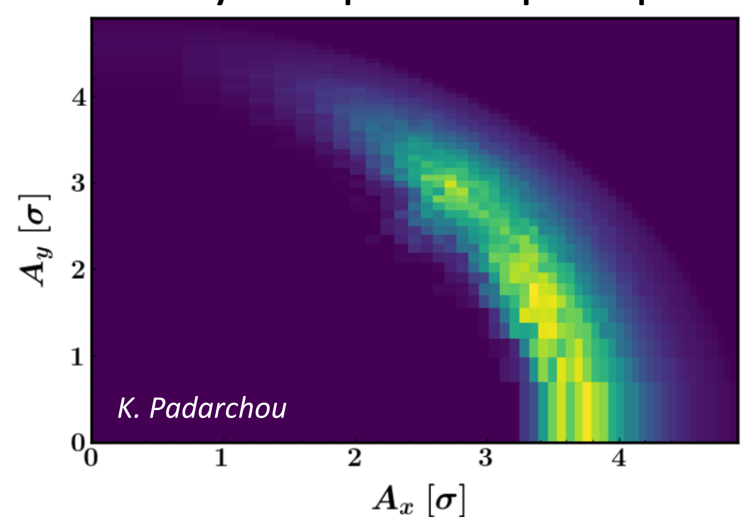
GPU acceleration allows simulation of very long time scales, of interest for the LHC.

Example for LHC:

- **Direct element-by-element simulation of the first 30 minutes** after bringing beams in collision (**20 M turns!**) to study lifetime, and tail depopulation
- Simulation of **20 000 particles**
  - On **CPU** (single process) would take **> 2 years simulation time**
  - On **GPU** (NVIDIA V100) was done in **< 3 days simulation time**



Density of lost particles in phase space



Outline

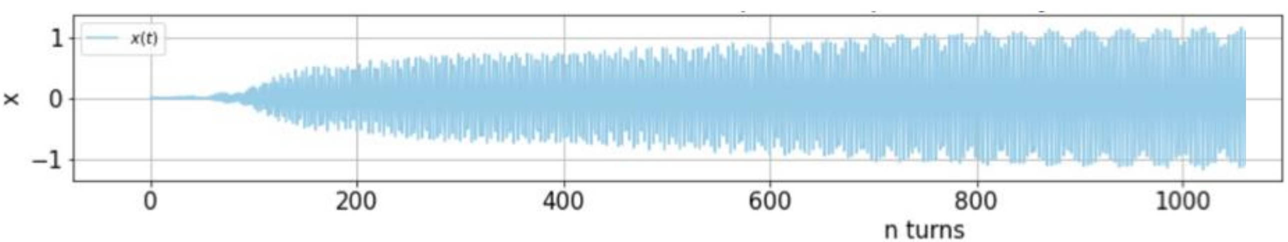
- **Introduction**
- **Architecture and main capabilities**
  - Lattice modeling
  - Single-particle tracking
  - Twiss module
  - Optimized
  - Particle-matter interaction
  - Synchrotron radiation
  - Collective effects (wakefields, spacecharge, IBS)
- **Beam-beam capabilities**
  - Weak-strong
  - Strong-strong (soft-Gaussian)
  - Beamstrahlung and Bhabha effect
  - Strong-strong (Particle In Cell)
- **Summary and final remarks**

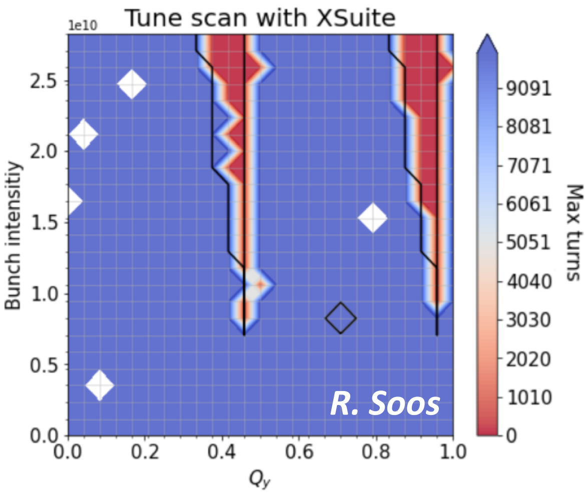
Strong-strong simulations – Soft-Gaussian approach

Supported by: CHART EPFL

With some simple additions, the weak-strong simulation engine is extended to simulate **coherent effects using the "Soft-Gaussuan approach":**

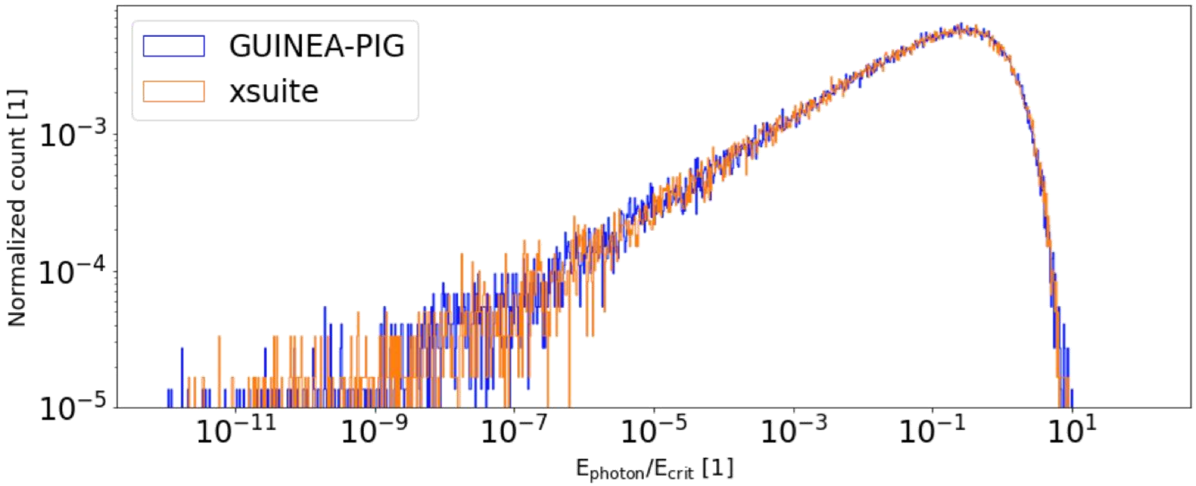
- The transverse distribution is assumed to be Gaussian
- **Both beams are tracked concurrently** (optionally on different CPU processes - MPI)
- At each beam-beam interaction, the **moments of each bunch are updated and used to compute forces on other bunch**
    - If the 6d method (Hirata) is selected such update is done slice by slice

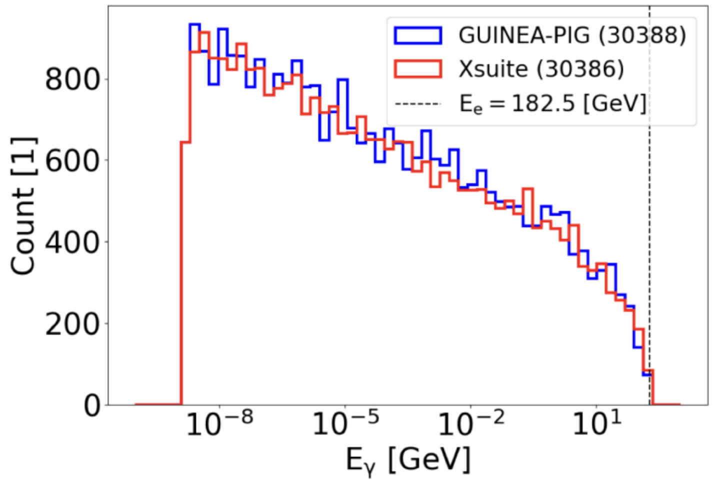




Supported by: CHART EPFL

Beamstrahlung and Bhabha effects are also modelled by the beam beam elements



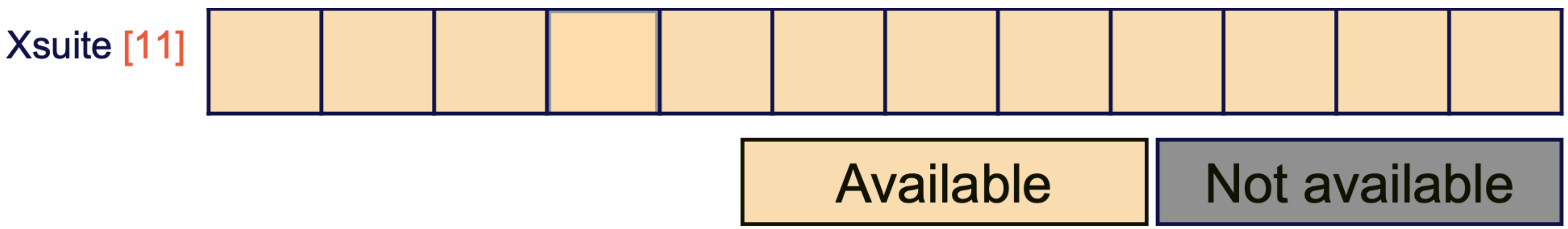


For more info see
P. Kicsiny et al. *https://indico.cern.ch/event/1160125/* *and* https://cds.cern.ch/record/2886033/

Outline

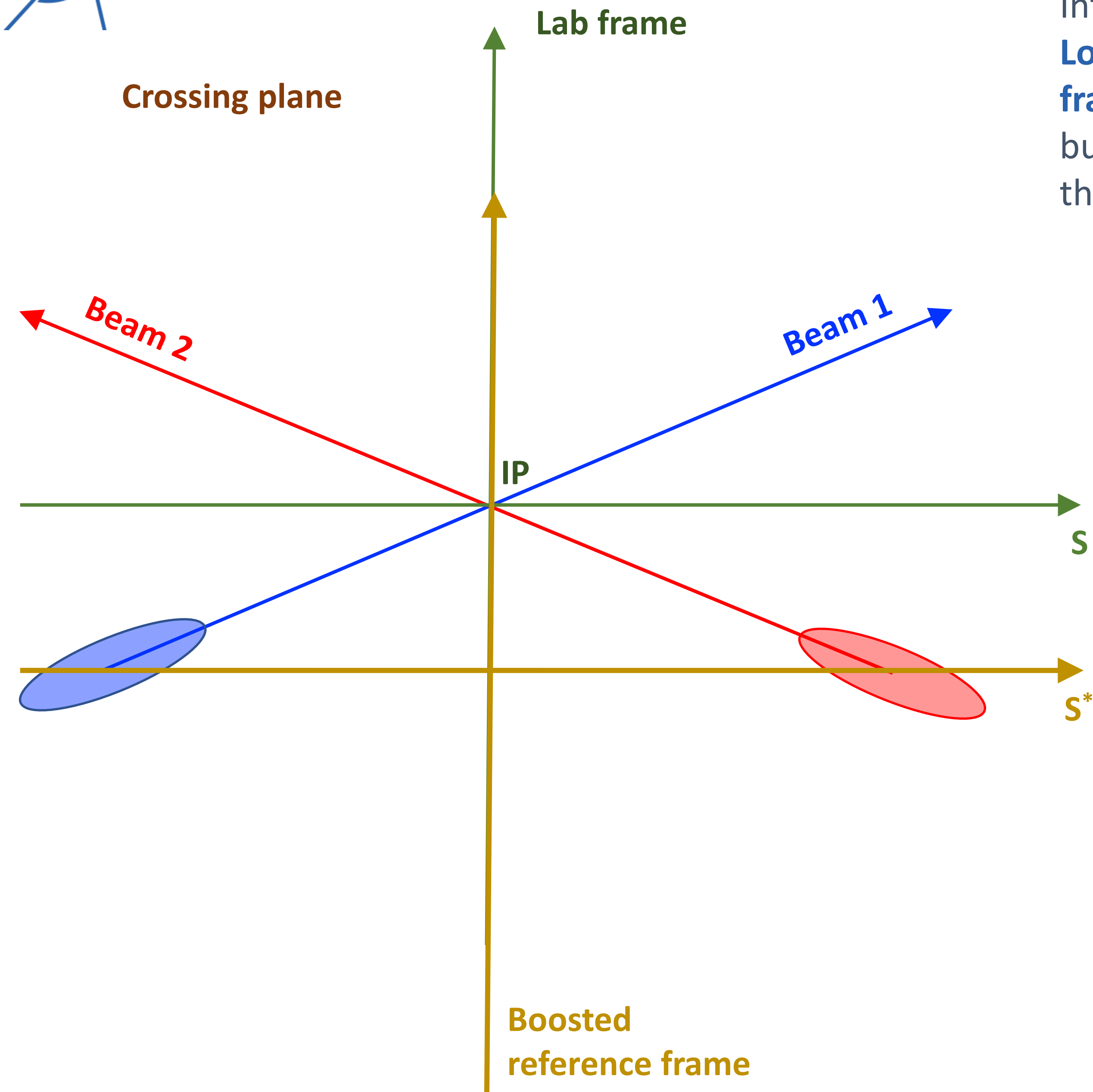
- **Introduction**
- **Architecture and main capabilities**
  - Lattice modeling

Last missing item on Peter's wishlist developed last summer ☺



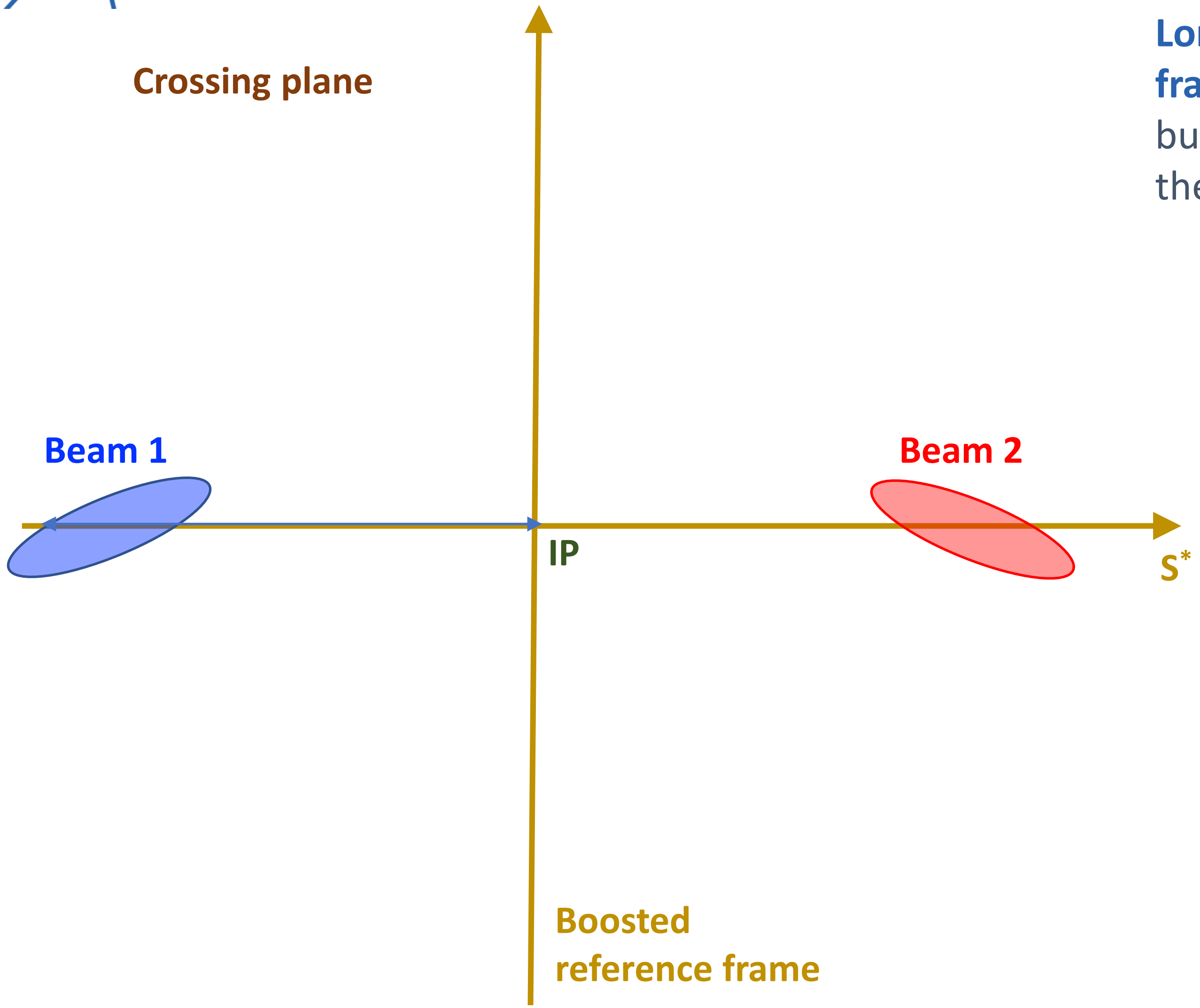
- **Beam-beam capabilities**
  - Weak-strong
  - Strong-strong (soft-Gaussian)
  - Beamstrahlung and Bhabha effect
  - Strong-strong (Particle In Cell)
- **Summary and final remarks**

Strong-strong simulations – Particle In Cell



Interaction is computed in a **Lorentz-boosted reference frame** in which the two bunches are moving along the same direction

Strong-strong simulations – Particle In Cell

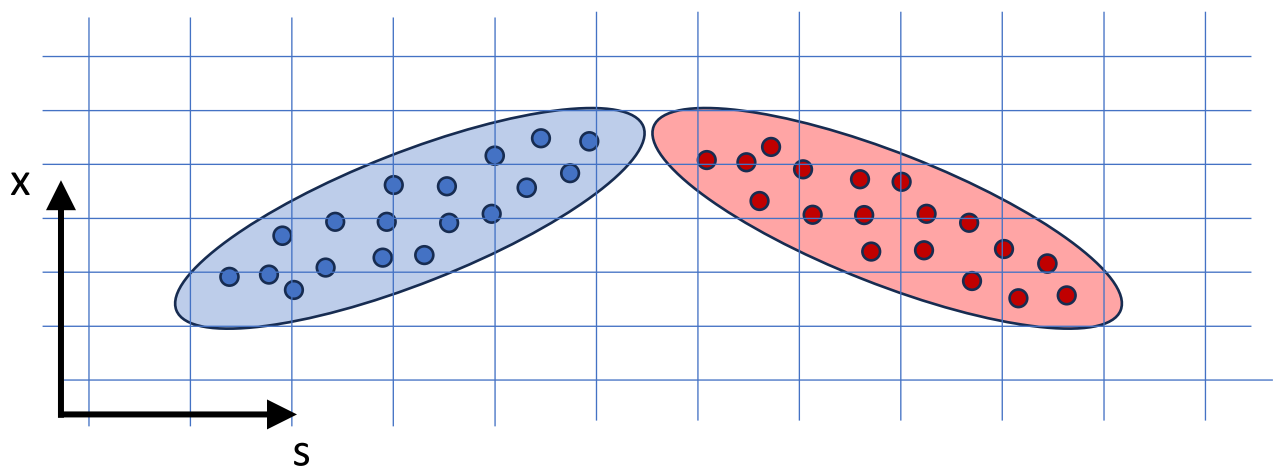


Interaction is computed in a **Lorentz-boosted reference frame** in which the two bunches are moving along the same direction

Strong-strong simulations – Particle In Cell

The beam-beam forces are computed on a **cartesian 3D Grid** using a **discrete time step**. At each step:

- The particles charge **is deposited on the grid cells**
- The scalar potential ϕ is computed by solving for each slice a **2D Poisson equations** (FFT method, Integrated Green Function[1])
- Force on individual particles is computed by **interpolation** (both transverse kicks and energy change are applied)
- **Particles are propagated** for a single time step

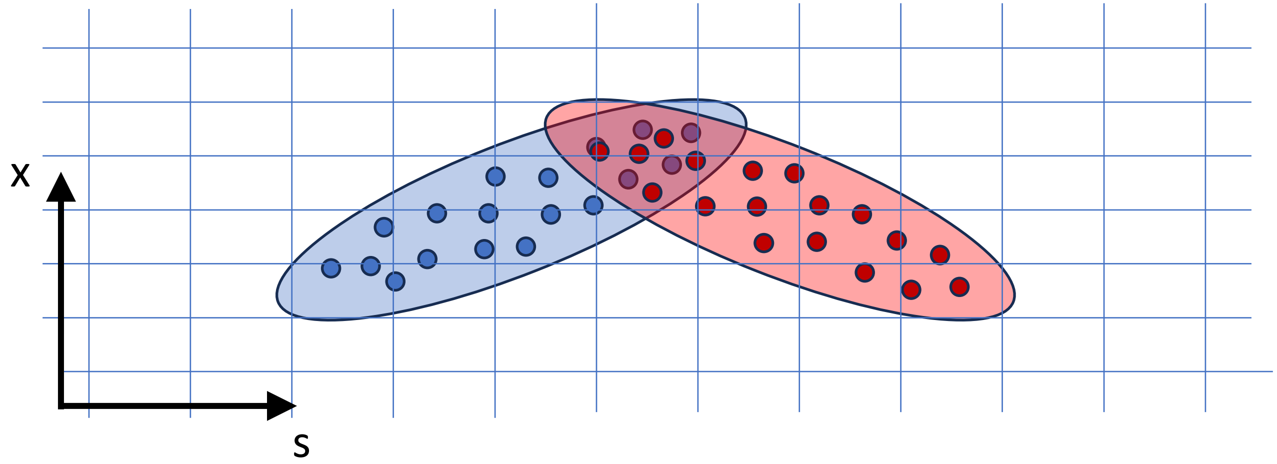


[1] From J. Qiang et al., "A parallel particle-in-cell model for beam–beam interaction in high energy ring colliders", Journal of Computational Physics 198 (2004) 278–294

Strong-strong simulations – Particle In Cell

The beam-beam forces are computed on a **cartesian 3D Grid** using a **discrete time step**. At each step:

- The particles charge **is deposited on the grid cells**
- The scalar potential ϕ is computed by solving for each slice a **2D Poisson equations** (FFT method, Integrated Green Function[1])
- Force on individual particles is computed by **interpolation** (both transverse kicks and energy change are applied)
- **Particles are propagated** for a single time step

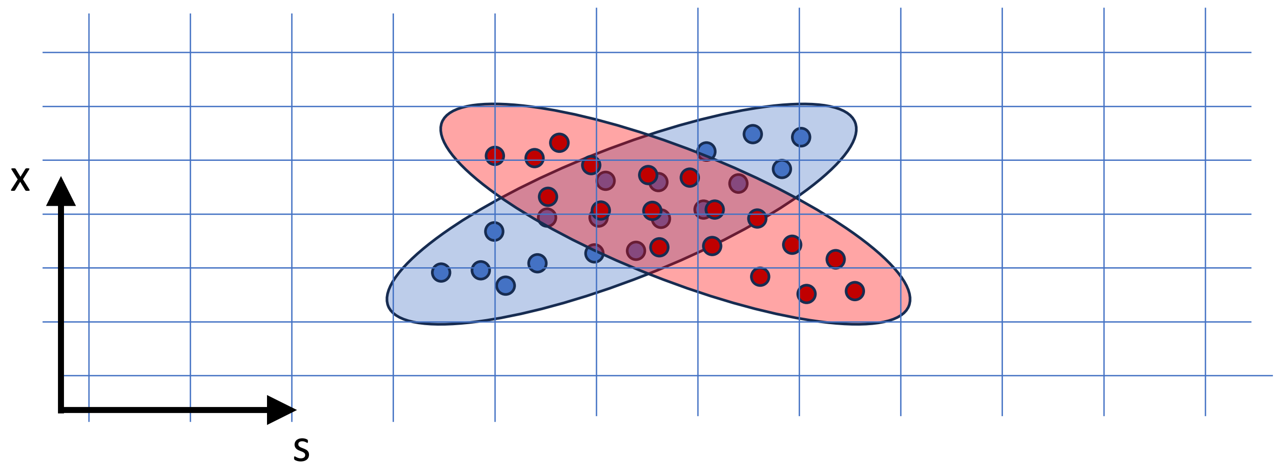


[1] From J. Qiang et al., "A parallel particle-in-cell model for beam–beam interaction in high energy ring colliders", Journal of Computational Physics 198 (2004) 278–294

Strong-strong simulations – Particle In Cell

The beam-beam forces are computed on a **cartesian 3D Grid** using a **discrete time step**. At each step:

- The particles charge **is deposited on the grid cells**
- The scalar potential ϕ is computed by solving for each slice a **2D Poisson equations** (FFT method, Integrated Green Function[1])
- Force on individual particles is computed by **interpolation** (both transverse kicks and energy change are applied)
- **Particles are propagated** for a single time step

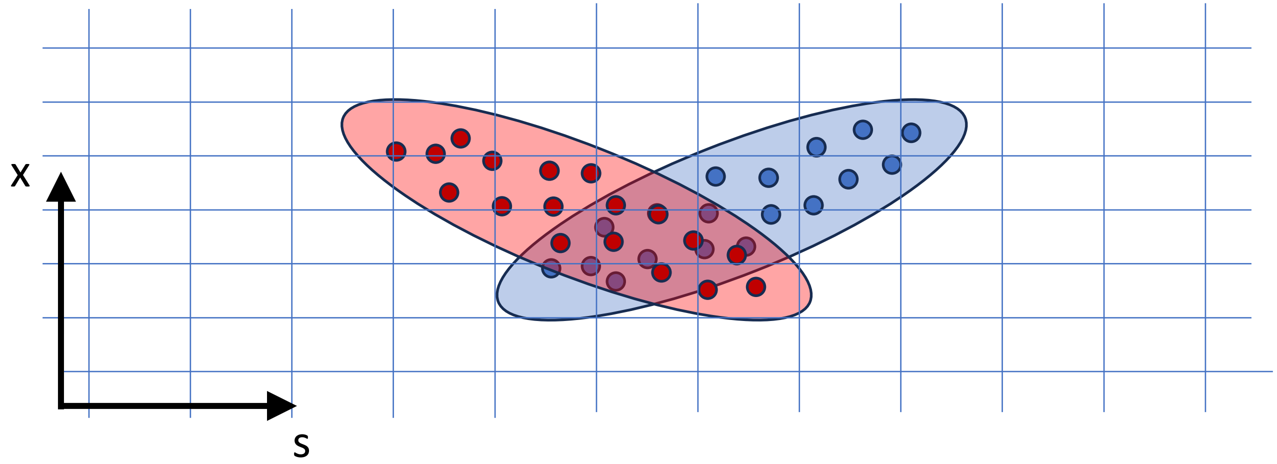


[1] *From J. Qiang et al., "A parallel particle-in-cell model for beam–beam interaction in high energy ring colliders", Journal of Computational Physics 198 (2004) 278–294*

Strong-strong simulations – Particle In Cell

The beam-beam forces are computed on a **cartesian 3D Grid** using a **discrete time step**. At each step:

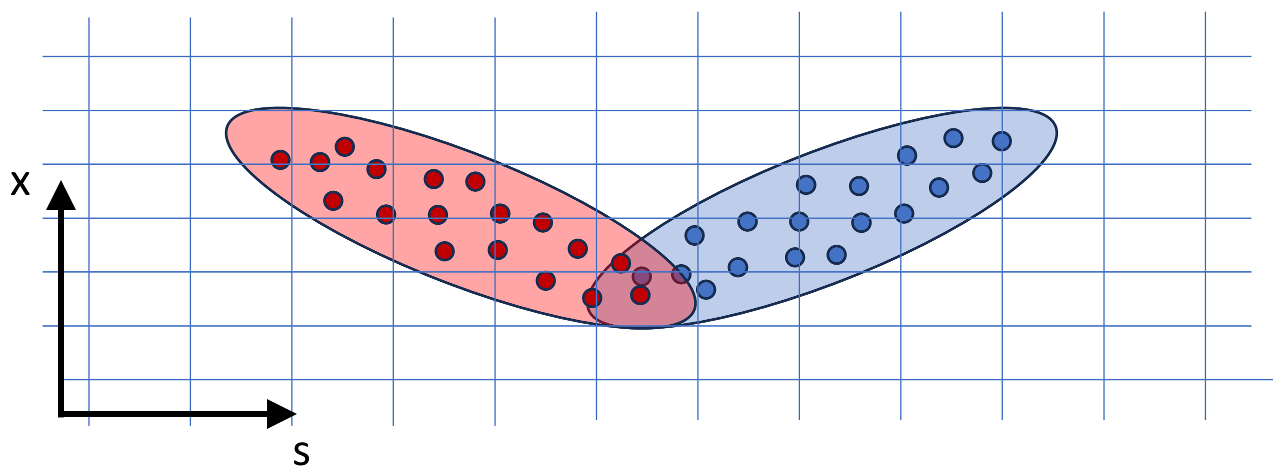
- The particles charge **is deposited on the grid cells**
- The scalar potential ϕ is computed by solving for each slice a **2D Poisson equations** (FFT method, Integrated Green Function[1])
- Force on individual particles is computed by **interpolation** (both transverse kicks and energy change are applied)
- **Particles are propagated** for a single time step



[1] From J. Qiang et al., "A parallel particle-in-cell model for beam–beam interaction in high energy ring colliders", Journal of Computational Physics 198 (2004) 278–294

The beam-beam forces are computed on a **cartesian 3D Grid** using a **discrete time step**. At each step:

- The particles charge **is deposited on the grid cells**
- The scalar potential ϕ is computed by solving for each slice a **2D Poisson equations** (FFT method, Integrated Green Function[1])
- Force on individual particles is computed by **interpolation** (both transverse kicks and energy change are applied)
- **Particles are propagated** for a single time step

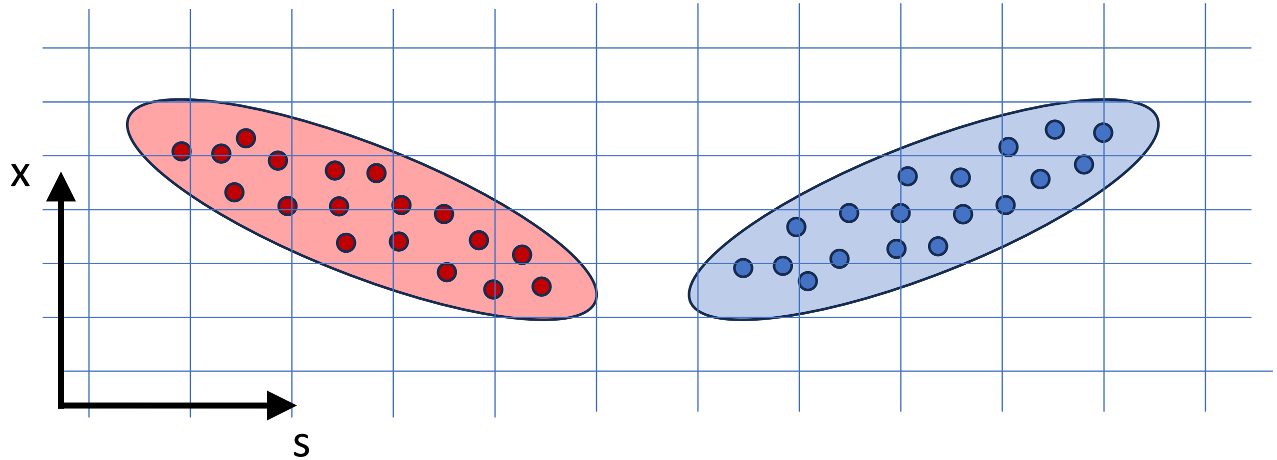


[1] From J. Qiang et al., "A parallel particle-in-cell model for beam–beam interaction in high energy ring colliders", Journal of Computational Physics 198 (2004) 278–294

Strong-strong simulations – Particle In Cell

The beam-beam forces are computed on a **cartesian 3D Grid** using a **discrete time step**. At each step:

- The particles charge **is deposited on the grid cells**
- The scalar potential ϕ is computed by solving for each slice a **2D Poisson equations** (FFT method, Integrated Green Function[1])
- Force on individual particles is computed by **interpolation** (both transverse kicks and energy change are applied)
- **Particles are propagated** for a single time step



[1] From J. Qiang et al., "A parallel particle-in-cell model for beam–beam interaction in high energy ring colliders", Journal of Computational Physics 198 (2004) 278–294

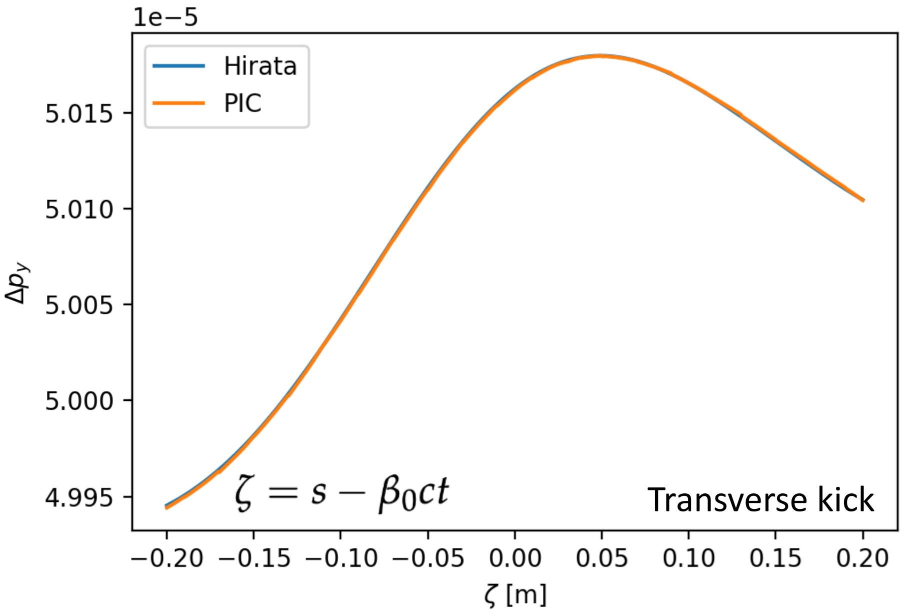
Strong-strong simulations – Particle In Cell

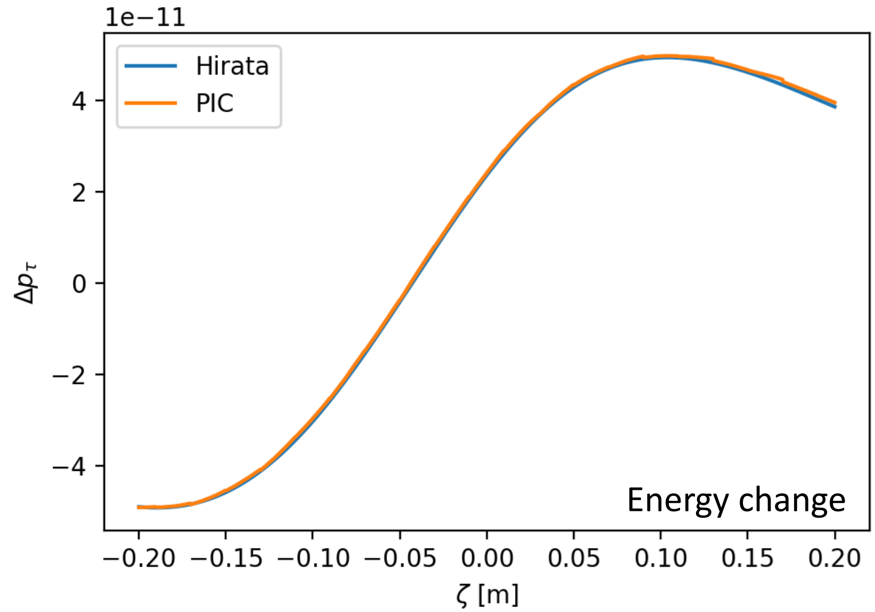
Result from PIC computations **checked against Hirata method for Gaussian distribution**

GPU acceleration available also for PIC calculation:

- Observed **speed-up of about x30** compared to single CPU process

Check on kick received by particles with a large angle as a function of the arrival time





Outline

- **Introduction**
- **Architecture and main capabilities**
  - Lattice modeling
  - Single-particle tracking
  - Twiss module
  - Optimized
  - Particle-matter interaction
  - Synchrotron radiation
  - Collective effects (wakefields, spacecharge, IBS)
- **Beam-beam capabilities**
  - Weak-strong
  - Strong-strong (soft-Gaussian)
  - Beamstrahlung and Bhabha effect
  - Strong-strong (Particle In Cell)
- **Summary and final remarks**

Summary and final remarks

Xsuite offers a **rather complete set of beam beam capabilities** ranging from **4D and 6D weak strong** modelling to self-consistent strong-strong interaction with both the **"Soft-Gaussian" and the Particle In Cell method**

These capabilities **can be combined with several other features** which often need to be studied in combination with beam beam, notably:

- Symplectic element-by-element tracking
- Optics calculations and matching
- Dynamic effects (functions, ripples, noise)
- Particle-matter interaction (collimation)
- Synchrotron radiation, beamstrahlung and Bhabha effects
- Collective effects (IBS, Wakefields, Space charge)

The code is **publicly available** on GitHub and can be installed through pip

- **You are very welcome to give it a try**
- **Installation instructions and many examples** available in the doc pages
- **Feedback is very welcome** (please contact us for issues, questions, suggestions)

The code is **open-source** and is open to **developments from the community:**

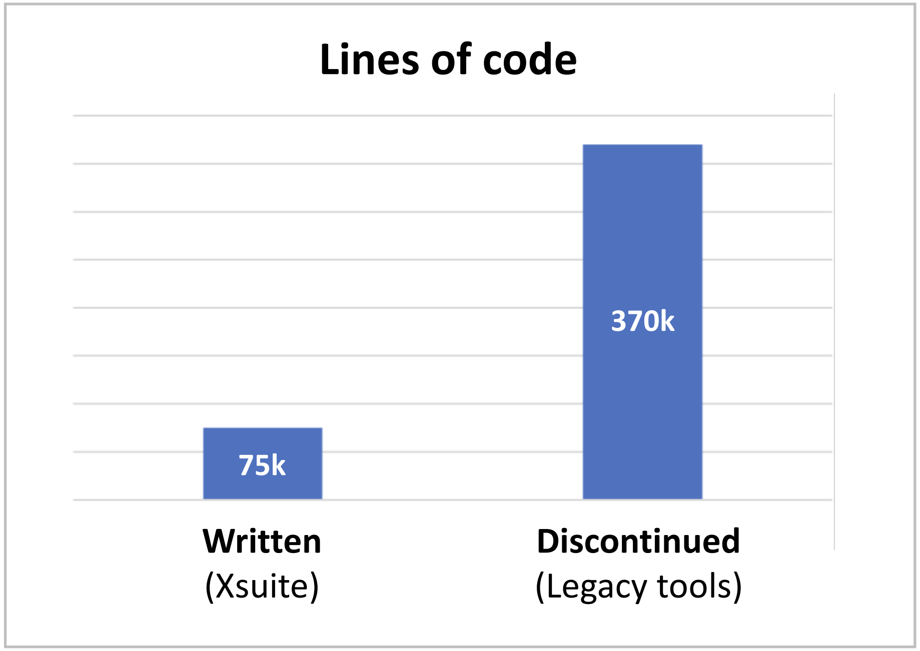
- Get in touch if you are interested in contributing to the development

Thanks for your attention!

In 2022-23 we have **essentially discontinued** the development and, to a very large extent, the usage of the following tools:

- Sixtrack
- Sixdesk
- Sixtracklib
- COMBI
- PySSD
- DistLib

This led to a **massive simplification** of our code base.

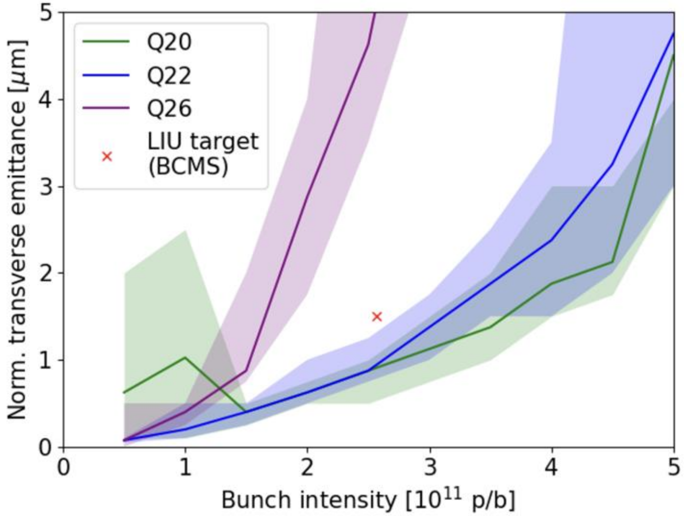


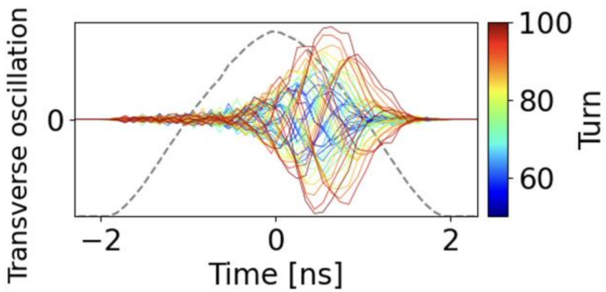
The implementation is largely based on **PyHEADTAIL-PyPIC**

Different **space-charge models** are implemented:

- The **“frozen” model**, in which particles interact with fixed charge distributions
- The **“quasi frozen”** model, in which the **beam intensity and beam sizes are recomputed at each interaction**
- The **“Particle In Cell (PIC)”** model:
    - Charge of tracked particles distributed on a **rectangular grid**
    - **Fast Poisson solver** based on **FFT** method with Integrated Green Functions
- Space charge simulations **strongly profiting from GPU acceleration**

Simulation campaign for the CERN SPS including full non-linear lattice, space charge and wakefields





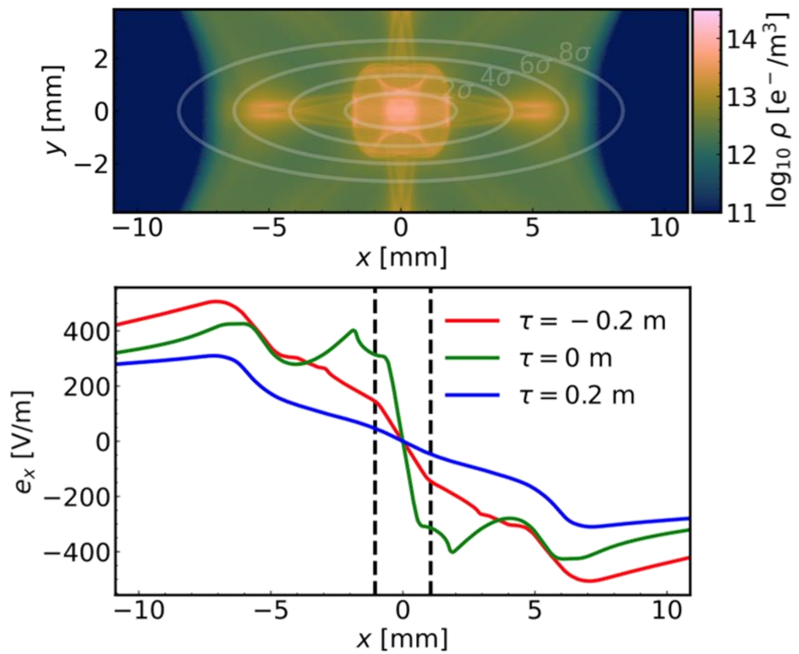
N. simulations	400
Number of PIC calculations per turn	540
Number of turns per simulation	40’000
Computing time per sim. (GPU)	~3 days
Computing time per sim. (CPU serial)	> 12 months

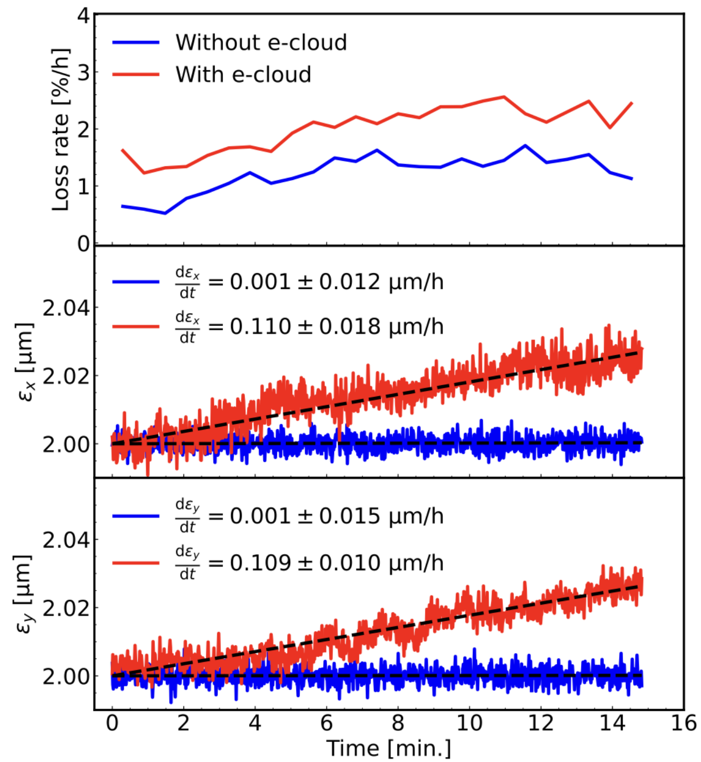
Courtesy X. Buffat

Electron cloud modeling

Xsuite has been exploited to study the **effect of electron cloud** on **slow beam degradation** (emittance growth, lifetime degradation).

- Done by applying a **high-order interpolation scheme** to the **e-cloud potential imported** from a dedicated multipacting simulator.
  - Scheme designed to **preserve the symplecticity of the resulting map** by ensuring the global continuity of the potential and required derivatives.
- **Use of GPUs is mandatory** to simulate the required long time scales (>10^6 turns).



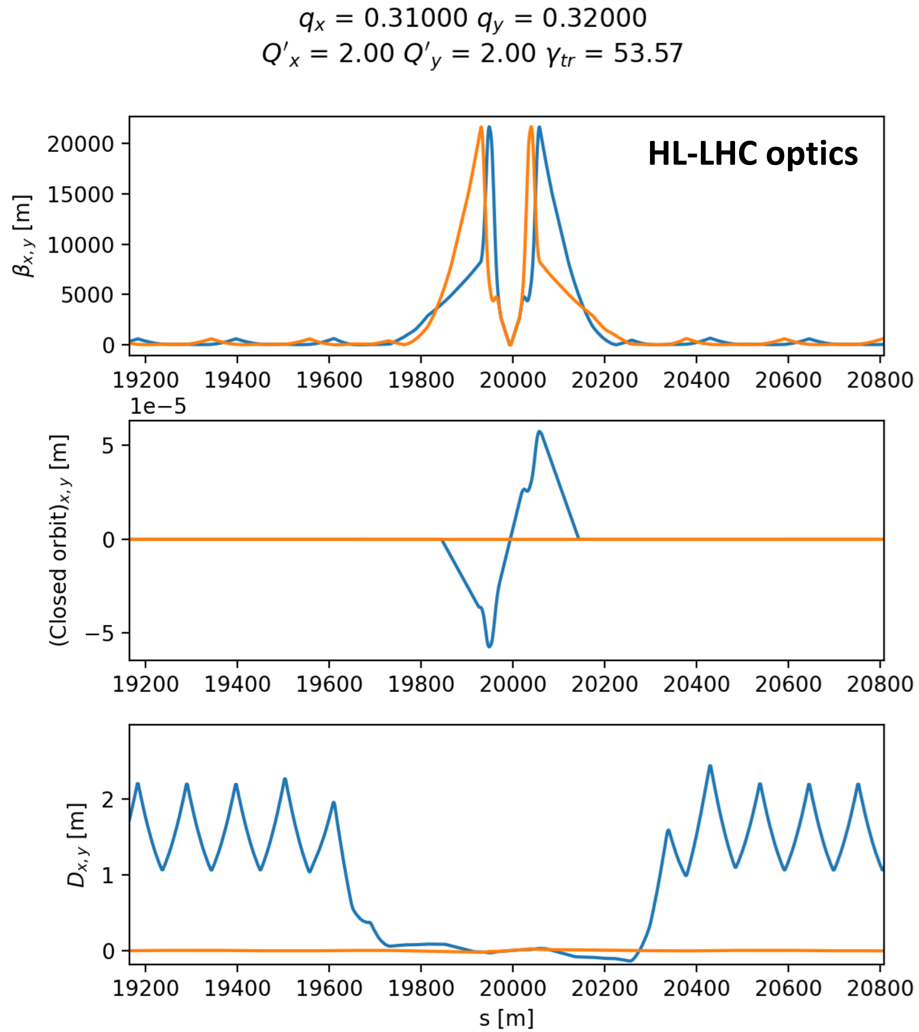


See also: K. Paraschou, THBP16

Twiss module

Computation of **Twiss parameters based on the tracking** has **two main advantages**:

- Any **physical model included in the tracking** is **automatically usable in Twiss**
  - Without additional development effort
- Twiss becomes a **powerful diagnostics tool** on the built **tracking model**
  - Allows **measuring directly on the tracking model** tunes, chromaticities, closed orbit, beta functions, etc.
  - Can be done **effortlessly** and **without exporting or manipulating the model**.
  - **Used daily** to for validating simulation models, catching mistakes, investigating issues

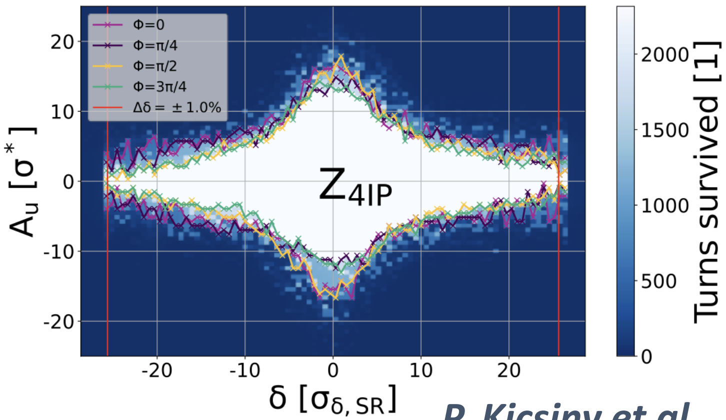


Accuracy compared to MAD-X: $\Delta\beta / \beta << 10^{-4}$
Computation time is very similar

```
In [37]: tw.bety[0] # xsuite
Out[37]: 149.4305507849305

In [38]: mad.table.twiss.bety[0] # madx
Out[38]: 149.43055000962505

In [39]: t_mad_ms
Out[39]: 202.0

In [40]: t_xsuite_ms
Out[40]: 185.0
```

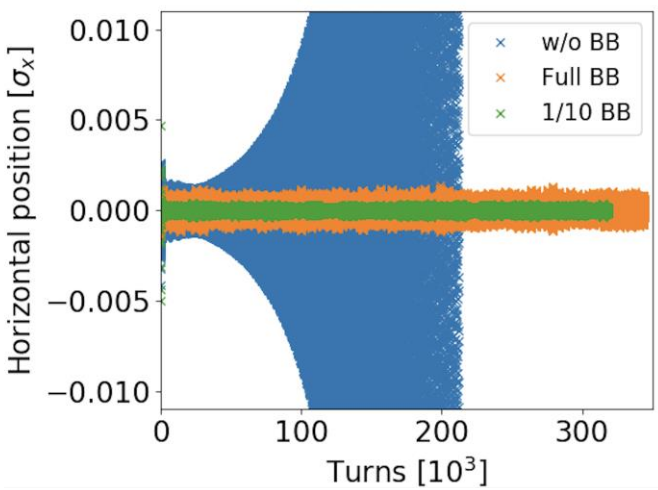
Example

FCC-ee DA studies (with bb)



P. Kicsiny et al.

Wakefield + beam-beam simulations for HL-LHC (strong-strong modelling)



- Xsuite implementation based on experience from **Sixtrack** and **COMBI**
- Two models are provided:
  - The **"4D" model**, which applies **only transverse** forces **independent on the longitudinal motion**
  - The **"6D" model**, which applies **longitudinal and transverse** forces accounting for the synchrotron motion (method by Hirata et al.)
- Both models can be used either in **"weak-strong" mode** (fixed assigned distribution for the other beam) or in **"strong-strong" mode** (self-consistent two-beam simulation, "soft Gaussian")
- For the simulation of lepton colliders, the code can also simulate for **beamstrahlung** and **Bhabha scattering** (developed in collaboration with EPFL)
- **Strong-strong simulations** are accelerated by **parallel computing on HPC clusters** (based on MPI)
  - **"Pipeline" algorithm**[(1)] used to optimize workload distribution across the nodes

(1) *S. Furuseth and X. Buffat, Comput. Phys. Commun. 244 (2019)*
(2) For more details, see presentation by P. Kicsiny

Dynamic control of beam line parameters

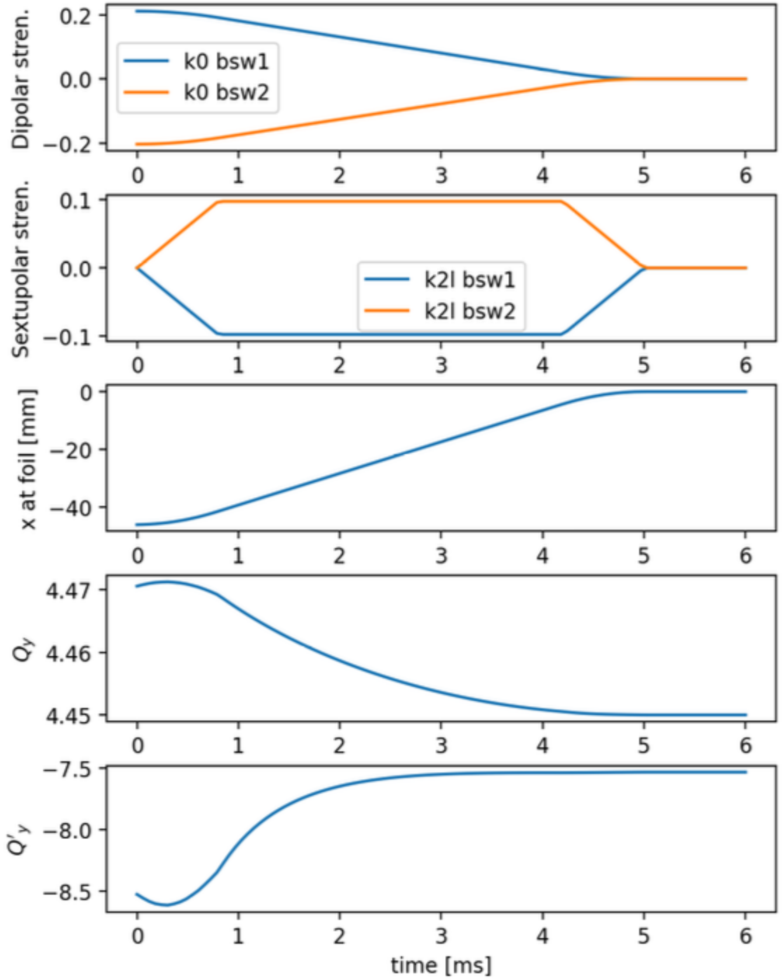
- In accelerators often a single **high-level parameter** can be used to control **groups of components** with complex dependency relations. (e.g. circuits with multiple magnets, groups of RF cavities, etc.)
- The **Xdeps module** provides the capability to **include such dependencies in the simulation model** (as done by MAD-X deferred expressions)
- **Example**, LHC crossing angle knob:

  At any time, the user can set:

```
lhc.vars['on_x1'] = 160 # murad
```

  which **automatically changes the strength of 40 dipole correctors** to get the required crossing angle
- User can also use **"Time functions"**, i.e. **time dependent knobs** that are updated automatically during the simulation

Simulation of a fast orbit bump used for the H^- injection into the CERN PS Booster



Synchrotron radiation

Supported by:

The effect of **synchrotron radiation** can be included in Xsuite tracking simulations. Two models available:

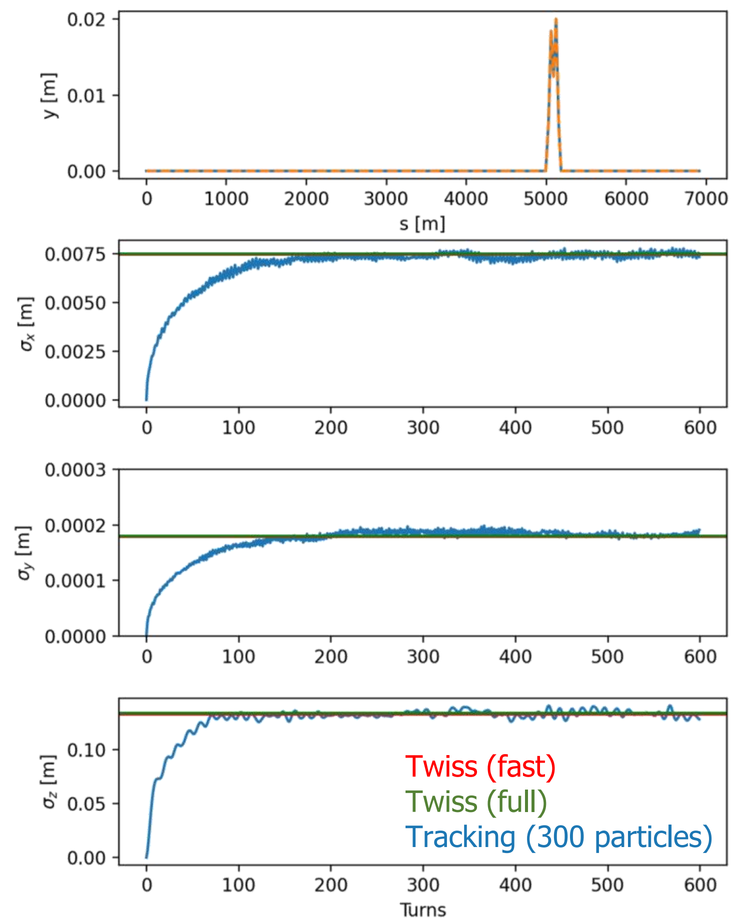
- The **“mean” model**, for which the energy loss from the radiation is applied particle by particle without accounting for quantum fluctuations;
- The **“quantum” model** for which the actual photon emission is simulated[1].

The **Xsuite Twiss** also includes:

- Dedicated algorithm for **non-symplectic one-turn map**[2]
- Computation of **radiation energy loss, damping times and equilibrium emittances**

An **automatic tool** is provided for **phasing the RF cavities** and **adjusting magnet strengths** to **compensate the radiation energy loss (“tapering”)**

Equilibrium emittance (twiss vs track)



[1] Based on H. Burkhardt, “Monte Carlo generator for synchrotron radiation”, 1990. Implementation ported from PLACET (A. Latina)
[2] E. Forest, From tracking code to analysis: generalised Courant-Snyder theory for any accelerator model. Springer, 2016