ISBN: 978-3-95450-253-0 ISSN: 2673-5571 doi:10.18429/JACoW-HB2023-TUA2I1

Xsuite: AN INTEGRATED BEAM PHYSICS SIMULATION FRAMEWORK

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,
G. Sterbini, F. F. Van Der Veken, CERN, Geneva, Switzerland
P. Belanger, TRIUMF, Vancouver, Canada
P. Niedermayer, GSI, Darmstadt, Germany
D. Di Croce, T. Pieloni, L. Van Riesen-Haupt, M. Seidel, EPFL, Lausanne, Switzerland

Abstract

Xsuite is a newly developed modular simulation package combining in a single flexible and modern framework the capabilities of different tools developed at CERN in the past decades, notably Sixtrack, Sixtracklib, COMBI and PyHEADTAIL. The suite consists of a set of Python modules (Xobjects, Xpart, Xtrack, Xcoll, Xfields, Xdeps) that can be flexibly combined together and with other accelerator-specific and general-purpose python tools to study complex simulation scenarios. The code allows for symplectic modeling of the particle dynamics, combined with the effect of synchrotron radiation, impedances, feedbacks, space charge, electron cloud, beam-beam, beamstrahlung, and electron lenses. For collimation studies, beam-matter interaction is simulated using the K2 scattering model or interfacing Xsuite with the BDSIM/Geant4 library. Tools are available to compute the accelerator optics functions from the tracking model and to generate particle distributions matched to the optics. Different computing platforms are supported, including conventional CPUs, as well as GPUs from different vendors.

INTRODUCTION

CERN has a long tradition in the development of software tools for beam physics in circular accelerators. In particular, over the years it has provided the following tools to the user community:

- MAD-X, which has become a standard to describe accelerator lattices, to perform optics calculation, design, and tracking simulations [1];
- SixTrack, a fast tracking program used mainly for long single-particle tracking simulations [2];
- SixTracklib, a C/C++ library for single-particle tracking compatible with Graphics Processing Units (GPUs) accelerators [3];
- COMBI, a simulation code for the simulation of beam-beam effects using strong-strong modelling [4];
- PyHEADTAIL, a Python toolkit for the simulation of collective effects (impedance, feedbacks, space charge, and e-cloud) [5].

These tools were developed over several years, mostly by independent teams. Although they provide very advanced features in their respective domains, with the exception of PyHEADTAIL and SixTracklib [6–8], their design does not allow effectively combining them for integrated simulations involving complex heterogeneous effects.

Several of these simulations are very well suited for computation acceleration based on GPUs. However, it would be cumbersome to retrofit such a capability in the existing codes.

Furthermore, some of the tools provide their own user interface, consisting in some cases in input/output text files, in some others in an ad-hoc scripting language like in the case of MAD-X. In contrast to this approach, the present de-facto standard in scientific computing is to provide software tools in the form of Python packages that can be easily used in notebooks or integrated within more complex Python codes. This allows leveraging an ever-growing arsenal of general-purpose Python libraries (e.g. for statistics, linear algebra frequency analysis, optimization, data visualization), which is boosted by substantial investments from general industry, especially for applications related to data science and artificial intelligence.

Based on these considerations, in 2021 the Xsuite project [9] has been launched to bring the know-how built in developing and exploiting the aforementioned codes into a modern Python toolkit for accelerator simulations, which is designed for seamless integration among the different components and for compatibility with different computing platforms, including multicore CPUs and GPUs from different vendors.

Xsuite has by now reached a mature stage of development and has already been adopted as "production tool" for several types of simulations across a quite large user community.

In this contribution, we will first describe the overall code structure and development strategy and then illustrate the main features and applications of Xsuite.

STRUCTURE, RESOURCES, AND DEVELOPMENT STRATEGY

Xsuite is composed of six modules:

- **Xtrack**: provides a single-particle tracking engine, featuring thick and thin maps for a variety of accelerator components, together with tools to load and save beam line models, track particles ensembles, characterize the beamline optics;
- **Xpart**: provides functions for the generation of particle distributions matched to the beamline optics;

* giovanni.iadarola@cern.ch

ISBN: 978-3-95450-253-0 ISSN: 2673-5571 doi:10.18429/JACoW-HB2023-TUA2I1

- **Xfields**: provides modules for the simulation of collective effects (space charge, beam-beam, electron clouds);
- **Xcoll**: provides tools for the simulation of particle-matter interaction in collimators and other beam-intercepting devices (see also [10]);
- **Xdeps**: manages tasks and deferred expressions for modeling and updating of accelerator circuits and other parameter dependencies, provides a multi-objective optimizer (see also [11]) and a general purpose text-based tabular data explorer; I
- **Xobjects**: provides the low-level infrastructure for memory management and multi-platform code compilation and execution (see also [11]).

All packages are available in the standard Python Package Index (PyPI), they can be installed with a single command (i.e. `pip install xsuite`) and can be easily updated to the latest version at any time.

The code has been designed bearing in mind that, while running on different hardware platforms and covering a large spectrum of phenomena and applications, the software needs to grow in a "sustainable" way, being managed and maintained by a small core team integrating contributions from a wider developer community. For this purpose, the project employs an "orthogonal" design strategy, ensuring that each module and functional block remains well-isolated and interacts with the others through clearly defined interfaces. This approach has two key advantages: firstly, it enables contributors to modify or expand specific components without requiring comprehensive knowledge of the other parts nor of the underlying software infrastructure. Secondly, it aims to minimize the overall codebase complexity, ensuring that it increases linearly rather than exponentially as new features are added.

Since the early stages, the developers engaged with the user community, encouraging and supporting the users to test and exploit the available features in full-scale simulations studies and integrating their feedback. The tool evolved incrementally, promptly applying corrections and improvements when needed. Such an approach is based on a fast release cycle, with new versions released typically multiple times per month, while ensuring that there is no disruption due to version changes on the user's side. This is made possible by an extensive effort in automatic testing, with each version of Xsuite undergoing over a thousand automatic checks (on CPU and GPU) before being rolled out to production.

Throughout the development process, we put a significant effort into building and maintaining a proper code documentation, which is of great importance when developing a tool that is meant to serve a large user community and to cover a broad set of use cases. Xsuite offers different complementary documentation sources [9]:

- A *User's guide*, describing how to install and use the code for different applications;
- A *Physics guide* describing the implemented physical models and numerical methods;
- A *Developer's guide* providing information on the code internals and describing how to add new features;
- *Docstrings* associated to each class and method, which are accessible directly in the Python interpreter or notebook and are collected in the *Xsuite API reference*.

AVAILABLE FEATURES AND THEIR APPLICATIONS

Lattice Modelling and Single-particle Tracking

The beam line is represented as a sequence of Python objects from the Xtrack module, each corresponding to an accelerator element or to other physical processes (e.g. magnets, cavities, aperture restrictions, etc.). The model can be defined manually by the user or imported from MAD-X, accounting for element tilts, misalignments, which are handled as changes of the reference system, and multipolar errors.

The implemented models are largely based on the SixTrack and SixTracklib implementations, where a 'thin' lattice integration method is adopted. Additionally, 'thick' models are available for bending magnets and quadrupoles. In particular, for bending magnets, it is possible to choose between a 'full' map, which is appropriate for small accelerators [12], or an 'expanded' one, which is appropriate for large machines and small bending angles [13]. Dipole edge effects (including fringe fields) can be included either in their linearized form or as full non-linear maps (using the same fringe model as in MAD-NG and PTC [14]).

To speed up the simulation, Xsuite assembles and compiles a C kernel callable from Python, which is able to track the entire beamline on 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 [15]. Developments are ongoing to deploy Xsuite on the LHC@Home volunteer computing platform [16].

Xsuite tracking simulations have been already used for several applications, notably slow extraction studies [17–20], Dynamic Aperture and lifetime simulations [21], simulations of halo cleaning using hollow electron lenses [15, 22], studies on chaos indicators for non-linear beam dynamics [23], dynamic aperture estimates based on machine learning [24].

Dynamic Control of Beam Line Parameters

Accelerators and beam lines have complex control patterns. For example, a single high-level parameter can be used to control groups of accelerator components (e.g. sets of magnets in series, groups of RF cavities, etc.) following complex dependency relations. The Xdeps module provides the capability to include these dependencies in the simulation model so that changes in the high-level parameters are automatically propagated down to the line elements properties (see also [11]).

For example, as in MAD-X, the user can set the crossing angle between the two colliding beams for the LHC by setting "`lhc.vars['on_x1'] = 160 # murad`", which automatically changes the strength of 40 dipole corrector magnets in

ISBN: 978-3-95450-253-0 ISSN: 2673-5571 doi:10.18429/JACoW-HB2023-TUA2I1

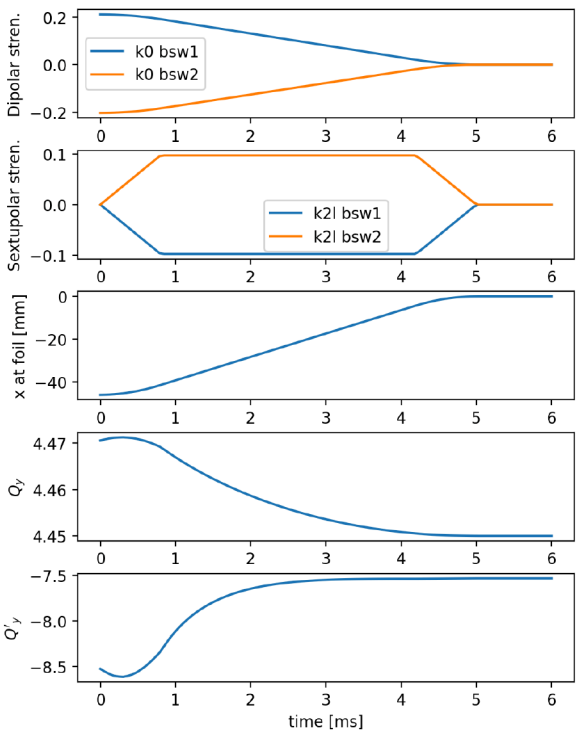


Figure 1: Simulation of a fast orbit bump used for the H^- injection into the CERN PS Booster. From top to bottom: 1) Time dependence of the dipolar strength of two of the dipole correctors; 2) corresponding sextupolar components from induced eddy currents; 3) resulting position at the stripping foil as computed by Xsuite; 4) effect of the dipole change on the vertical tune (mostly due to fringe fields); 5) effect of the dipole change on the vertical chromaticity.

the two beam lines to control the angle of the two beams at the interaction point and compensate the resulting spurious dispersion. Dependencies among parameters can be defined directly by the user or imported from the MAD-X model and can be easily inspected and modified at any time. Furthermore, it is possible to define "time functions", i.e. time dependent knobs that are updated automatically during the simulation. For example, Fig. 1 shows the evolution of the tune and chromaticity as a result of fringe fields and eddy currents in the fast dipole magnets used for the H^- injection in the CERN PSB.

Dedicated features have been developed to efficiently model noise or ripples in a large number of machine elements as well as fast beam excitations. These have been used, for example, to study different methods for slow extraction and to study the effect of amplitude and phase noise on the HL-LHC crab cavities [17, 18, 25].

Twiss Module

The user can easily obtain the lattice functions of a ring or a beamline using the Twiss method associated to the Xsuite beam line object. The calculation, which probes the lattice simply by tracking suitable particles, is performed through the following steps:

1. The closed orbit is found by applying a standard Python root finder to identify the fixed point of the one-turn map (by tracking a particle at each iteration);
2. The Jacobian matrix of the one-turn map is computed by tracking particles to evaluate the derivatives, using a central-difference formula;
3. Lattice functions are obtained by computing the "Linear Normal Form" of the map from the eigenvalues and eigenvectors of the Jacobian matrix [26];
4. Particles tracking is used to propagate the eigenvectors along the beam line;
5. Twiss parameters (α, β, γ), dispersion functions, phase advances, as well as the effect of linear coupling are obtained using the Mais-Ripken approach [27].

The accuracy of such a method is found to be excellent for all accelerators tested so far. For example, for the LHC, the computed beta functions are consistent with the values computed by MAD-X up to the fifth significant digit. The Twiss computation time is similar to other tools used for the same purpose. For example, for the LHC, the Twiss computation takes a similar time compared to MAD-X.

The Twiss computation can be optionally performed on a portion of the beam line with given initial conditions, and for an assigned beam momentum, which allows studying off-momentum beta-beating, non-linear chromaticity etc.

The fact that the computation of the Twiss parameters is based on the tracking engine itself provides two main advantages. Firstly, the effect of any physical model included in the tracking is automatically accounted for in the Twiss calculation without additional code development effort and including cases where no analytic expression is available for the map, e.g. for interpolators. Secondly, this makes the Twiss a powerful diagnostics tool on the built tracking model, allowing to measure key quantities like tunes, chromaticities and closed orbit on the tracking model itself directly, effortlessly and without exporting or manipulating the model. This proved to be an invaluable tool for identifying and investigating issues with user's models and in the code itself.

The results of the Twiss method are also used to generate particles distributions matched to the optics, and to configure optics-dependent parameters in the simulations, e.g. collimator gaps, beam sizes in space-charge and beam-beam elements or grid sizes in electron cloud simulations.

Optimizer

Accelerator design and simulations often require solving optimization problems. Typical examples consist in setting given quadrupole and sextupole circuits to obtain desired values of the tunes or the chromaticity, or in using multiple orbit correctors to obtain a closed orbit bump at a given location, or in optimizing quadrupole strengths to fulfill given constraints on the lattice functions (optics matching).

For this purpose, Xsuite provides an optimizer module to "match" model parameters to assigned constraints, which is built based on the extensive experience of MAD-X. The chosen optimization algorithm is the same implemented in

ISBN: 978-3-95450-253-0 ISSN: 2673-5571 doi:10.18429/JACoW-HB2023-TUA2I1

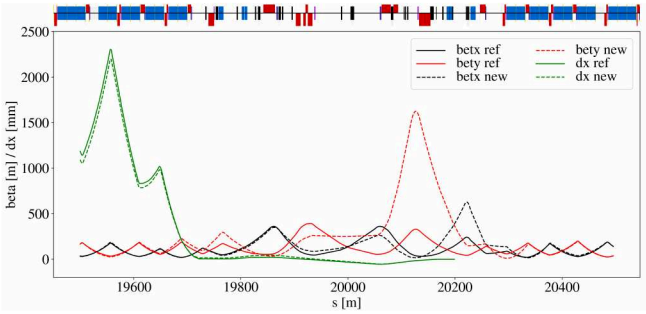


Figure 2: Novel optics for the LHC betatron collimation insertion (IR7) designed using the Xsuite optimizer to improve the beam cleaning efficiency and decrease the contribution of the collimators to the transverse impedance of the ring [29].

MAD-X [28], which has proven over the years to be very well suited for accelerator design and tuning.

The optimizer interface is designed to maximize usage flexibility. In particular, the user can interact with the program during the optimization process, by enabling/disabling targets or knobs, changing target values and tolerances, rolling back optimization steps, and changing knob limits. It is possible to perform optimizations involving targets and knobs from multiple beam lines. The optimizer can also be used to synthesize knobs controlling multiple parameters, as in the case of the crossing angle knob illustrated above.

While by default the targets consist of selected outputs of the Twiss calculations, it is possible to define custom targets and actions involving arbitrarily complex operations. For example, a special "Luminosity" target is defined to control beam separations in a collider to obtain a desired luminosity. In another application shown in Fig. 2, a custom action tracking particles within the LHC collimation area, has been used to optimize the optics functions in order to maximize the efficiency of the collimation system [29]. The Xsuite optimizer has also been used for optics matching of the LHC and of FCC-ee collider, showing its capability of handling large problems with several constraints and degrees of freedom.

Synchrotron Radiation Models and Compensation

The effect of synchrotron radiation can be included in Xsuite tracking simulations. For this purpose, the user can choose between two models:

- 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 including quantum fluctuations, using the algorithm described in [30].

Presently, these features are implemented only for the "thin" lattice modelling.

When synchrotron radiation is present, additional calculations can be enabled in the Twiss calculation. In particular, the energy loss from synchrotron radiation is measured along the beam line, the damping times for the longitudinal and transverse planes are computed from the eigenvalues

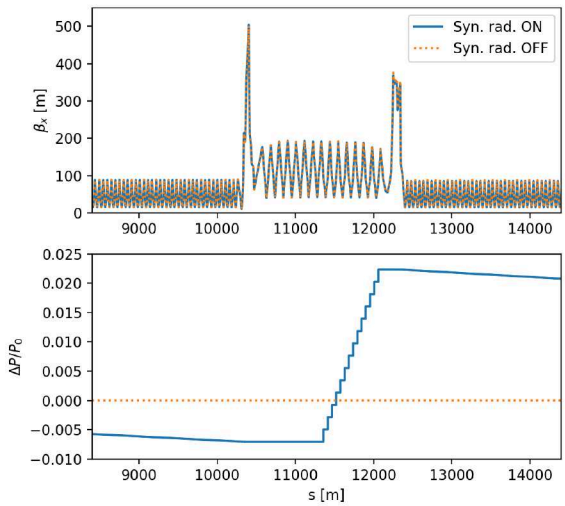


Figure 3: Optics and beam energy deviation in the area close to one of the RF insertions of the FCC-ee collider plotted in the absence of radiation (orange) and after the energy loss compensation (blue).

of the one-turn matrix, and the equilibrium emittances are calculated from the lattice linear normal form following the approach described in [31].

In the presence of radiation, the one-turn matrix is not symplectic, hence the lattice functions cannot be calculated using the conventional Mais-Ripken approach. Instead, two alternative methods are provided by Xsuite to compute the lattice functions in the presence of radiation:

- the "full" method, which computes the De Moivre representation of the one-turn matrix from which the lattice functions can be extracted [32];
- the "kick as closed orbit" method, which uses modified tracking maps that measure the momentum changes from synchrotron radiation on the closed orbit and apply the same kicks to all particles. This provides a symplectic one turn matrix accounting to first order for the energy loss from the radiation, on which the Mais-Ripken approach is then applied.

The second method is used by default since, for most cases of interest, it provides an excellent compromise between accuracy and computation speed.

In high-energy lepton rings, the beam looses significant energy due to synchrotron radiation. RF cavities around the ring need to be correctly phased to compensate for the loss and the strength of magnetic elements needs to be adjusted to match the actual energy of the beam at the magnet location (this operation is often called "tapering"). These two corrections are interdependent and cannot be done sequentially. Xsuite provides an automatic iterative method that obtains the compensation by performing the following steps:

1. Find the 4D closed orbit with no radiation and frozen particle energy along the ring;
2. With radiation enabled, track a particle on the closed orbit while adjusting the strength of each magnet to the actual energy of the incoming particle, which allows

ISBN: 978-3-95450-253-0 ISSN: 2673-5571 doi:10.18429/JACoW-HB2023-TUA2I1

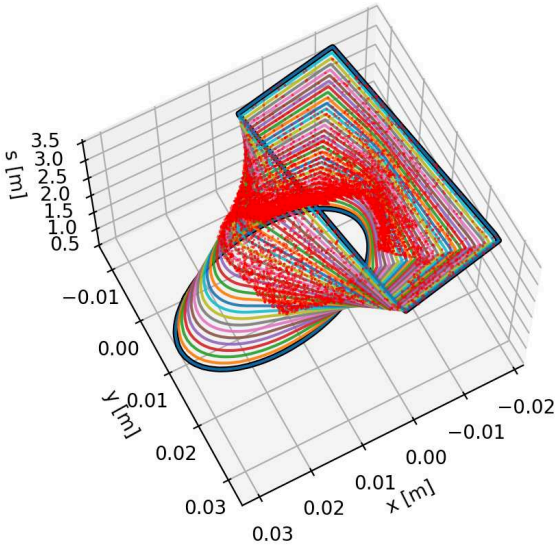


Figure 4: Precise localization of particles lost in the transition between a rectangular aperture and an elliptic aperture, which are shifted and tilted with respect to each other.

measuring the energy loss while preserving the optics and the orbit

3. Adjusts phases of RF cavities to compensate the energy loss, distributing the energy loss according to the cavity voltage;
4. Repeat 2-3 until convergence is achieved.

Tracking and Twiss with radiation as well as the energy loss compensation method are presently in use for studies within the FCC project, handling energy loss per turn as high as 5% of the total beam energy. Figure 3 shows the optics and the beam momentum in a region close to one of the RF insertion of the FCC-ee collider. One can observe how the optics is perfectly preserved by the energy compensation algorithm.

Particle-matter Interaction and Collimation

Xsuite provides dedicated features for simulating particle-matter interactions and to study beam collimation.

The tools to model the interaction of beam particles with intercepting devices like collimators, targets, dumps, crystals are provided through the Xcoll package [10]. The interaction is simulated using one of the following engines:

- The "Everest" engine embedded in Xcoll, which is an evolution of the K2 model developed for SixTrack [33];
- The "Geant 4" engine, which exploits an interface between Xsuite and the Geant4 library [34], built through the BDSIM library [35, 36].
- The "FLUKA" engine (presently being finalized and tested), which exploits an interface between Xsuite and the FLUKA Monte Carlo code [37].

Xcoll also provides facilities to automatically install sets of collimators in the Xtrack beam line and to set the positions of their jaws accounting for the local orbit and beam size as calculated through the Xtrack Twiss module.

An important goal of simulation studies for collimation is the precise localization of the beam losses along the accelerator, to estimate the power deposition on accelerator components in order, for example, to study equipment activation or quench limits. The aperture model of the accelerator can be imported by Xsuite as part of the MAD-X sequence. Dedicated elements are installed along the beam line that check the particle positions with respect to the defined aperture and stop the tracking for the particles found to be outside. For collimation studies, such a check is performed at each magnetic element slice, which provides a localization of the losses with a few meters accuracy. The localization is then further improved in a post-processing stage to reach the accuracy set by the user (typically 1 to 10 cm), using particle backtracking and a locally refined aperture model obtained by polygonal interpolation (an example is shown in Fig. 4).

Integration of Collective Effects

Collective elements, i.e. elements for which the action on a particle depends on the coordinates of other particles, can also be part of an Xsuite beam line. For such elements, the tracking of different particles cannot happen asynchronously.

In Xsuite, the handling of collective beam elements is fully automatic. The Xtrack line module identifies the collective elements and splits the sequence at the locations of the collective elements, so that the simulation of the non-collective parts can be done asynchronously to gain speed, while the simulation of the collective effects is performed synchronously.

If requested by the collective element, the particles' data is automatically transferred from GPU to CPU for the collective calculation and then moved back.

The simulation of wakefields and transverse feedback systems can be performed by inserting the corresponding PyHEADTAIL elements into the Xsuite line, after enabling a dedicated compatibility layer, which handles the different naming conventions adopted in the two codes.

The effect of space charge and beam-beam, as well as incoherent effects from electron clouds are simulated through the corresponding features of the Xfields module, which will be discussed briefly in the following sections.

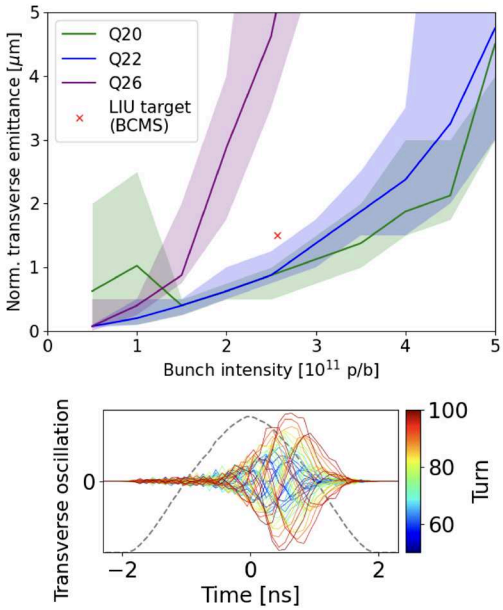
Space Charge

The effect of beam space charge can be included in the simulation by inserting dedicated "lumped" space-charge elements into the beam line, which apply the space-charge forces to the circulating beam particles. To correctly model the impact of space charge on the beam dynamics, it is necessary to install enough elements to achieve a good sampling of the beam envelope, which typically results in hundreds of space-charge elements installed in the ring.

Xsuite provides an automatic tool to install and configure the space-charge elements. The user can choose among different space-charge models:

- The "frozen" model, in which particles interact with frozen charge distributions that are configured based

ISBN: 978-3-95450-253-0 ISSN: 2673-5571 doi:10.18429/JACoW-HB2023-TUA2I1



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

Figure 5: Simulation study for the CERN SPS based on simulations including the full non-linear lattice, space charge (PIC model) and wakefields. Top: obtained instability thresholds for different beam optics; middle: one of the simulated instabilities; bottom: some relevant simulation parameters.

on the parameters provided by the user (intensity, emittances) and are kept constant over the simulation. In this case, the Xtrack Twiss module is used to compute the optics and obtain transverse beam sizes. The transverse distribution is assumed to be Gaussian.

- 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 from the circulating beam particles.
- The "Particle In Cell (PIC)" model, which consists in the self-consistent computation of the beam field at each space-charge interaction, obtained by distributing the charge from the tracked particle on a uniform rectangular grid and by solving the Poisson equation using a fast solver based on the FFT method with Integrated Green Functions [38]. The implementation is largely based on the one from PyHEADTAIL-PyPIC [39].

Space-charge simulations, especially those employing the PIC method, can be significantly time-consuming. The use of GPUs provides a substantial speed-up. Figure 5 shows the outcome of a study on beam stability for the CERN SPS based on simulations including the full non-linear lattice, space charge (PIC model) and wakefields. In this case, the GPU-accelerated simulations turned out to be more than 100 times faster compared to the serial CPU implementation.

Beam-beam Interactions

Similarly to the SixTrack and COMBI simulation codes, Xsuite provides two models for the simulation of beam-beam effects in colliders:

- The "4D" model, which applies only transverse forces independent on the particles longitudinal coordinates, assuming a transverse Gaussian beam distributions;
- The "6D" model, which applies longitudinal and transverse forces accounting for the particles longitudinal coordinates, using the approach described in [40], assuming a Gaussian distribution in the transverse plane, and allowing for arbitrary longitudinal beam profile.

Both models can be used either in "weak-strong" mode, simulating the interaction of the beam particles with a fixed charge distribution modelling the other beam, or in "strong-strong" mode where all bunches in the two beams are actually simulated by tracking particles and their interaction is computed accounting for the evolving moments of their charge distribution (updated with a frequency defined by the user).

A tool is provided to automatically identify the locations of the beam-beam encounters, install and configure the beam-beam lenses, based on the geometry of the two rings and on the closed orbit and optics of the two beams as obtained from the Xtrack Twiss. The simulation of beam-beam compensation with wires is also implemented [41].

In the strong-strong mode, simulations can be very heavy, especially when a large number of bunches needs to be simulated (in the LHC case, thousands of bunches per beam are required). For these cases, the use of parallel computing on HPC clusters is exploited by simulating different bunches on different computing nodes, which exchange information on the distribution of the respective bunches using MPI communication based on the pipeline algorithm [42].

For the simulation of lepton colliders, the 6D beam element can also account for beamstrahlung and Bhabha scattering. The models implemented for this purpose have been benchmarked against the GUNEA-PIG code [43, 44].

Presently, Xsuite has replaced SixTrack and COMBI as workhorse for weak-strong and strong-strong beam-beam studies for the LHC and HL-LHC [45] and is being used for beam-beam studies for the design of the FCC-ee lepton collider [46].

Incoherent Effects from Electron Cloud

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

This is done by applying a high-order interpolation scheme, to the e-cloud potential imported on a discrete grid from a dedicated multipacting simulator. Also in this case, the use of GPUs is mandatory in order to simulate the required long time scales (millions of revolutions).

The interpolation scheme is designed to preserve the symplecticity of the resulting map by ensuring the global continuity of the potential, its first derivatives and selected second-order derivatives. More details on these studies can be found in Refs. [47–49].

ISBN: 978-3-95450-253-0 ISSN: 2673-5571 doi:10.18429/JACoW-HB2023-TUA2I1

ACKNOWLEDGMENTS

The authors would like to thank F. Asvesta, H. Bartosik, D. Demetriadou, J. Dilly, C. Droin, S. Fartoukh, A. Fornara, M. Hofer, M. Giovannozzi, S. Kostoglou, P. Kruyt, B. Lindstrom, C. E. Montanari, Y. Papaphilippou, T. Prebibaj, T. Pugnat, S. Redaelli, G. Rumolo, R. Tomas for their important input and support to the development of Xsuite.

Work supported by the Swiss accelerator research and technology (CHART).

REFERENCES

[1] R. De Maria *et al.*, "Status of MAD-X V5.09," in *Proc. IPAC'23*, Venice, Italy, 2023, pp. 3340–3343. doi:10.18429/JACoW-IPAC2023-WEPL101

[2] R. D. Maria *et al.*, "SixTrack Version 5: status and new developments," *J. Phys.: Conf. Ser.*, vol. 1350, no. 1, p. 012 129, 2019. doi:10.1088/1742-6596/1350/1/012129

[3] M. Schwinzerl, H. Bartosik, R. D. Maria, G. Iadarola, A. Oeftiger, and K. Paraschou, "Optimising and extending a single-particle tracking library for high parallel performance," in *Proc. IPAC'21*, Campinas, SP, Brazil, 2021, paper THPAB190, pp. 4146–4149. doi:10.18429/JACoW-IPAC2021-THPAB190

[4] T. Pieloni, "A study of beam-beam effects in hadron colliders with a large number of bunches," Presented on 4 Dec. 2008, 2008. https://cds.cern.ch/record/1259906

[5] K. S. B. Li *et al.*, "Code development for collective effects," in *Proc. HB'16*, Malmö, Sweden, Jul. 2016, pp. 362–367. doi:10.18429/JACoW-HB2016-WEAM3X01

[6] M. Madhyastha, *GPGU accelerated beam dynamics interfacing PyHEADTAIL with SixTrackLib*, 2019. doi:10.5281/zenodo.2537967

[7] V. Kornilov, A. Oeftiger, O. Boine-Frankenheim, V. Chetvertkova, S. Sorge, and P. Spiller, "Beam quality and beam loss predictions with space charge for SIS100," *J. Instrum.*, vol. 15, P07020, 2020. doi:10.1088/1748-0221/15/07/P07020

[8] I. Hofmann, A. Oeftiger, and O. Boine-Frankenheim, "Self-consistent long-term dynamics of space charge driven resonances in 2D and 3D," *Phys. Rev. Accel. Beams*, vol. 24, p. 024 201, 2021. doi:10.1103/PhysRevAccelBeams.24.024201

[9] *Xsuite documentation*, 2023. http://xsuite.web.cern.ch

[10] F. F. Van der Veken *et al.*, "Recent developments with the new tools for collimation simulations in Xsuite," presented at HB'23, Geneva, Switzerland, Oct. 2023, paper THBP13.

[11] S. Łopaciuk, R. De Maria, and G. Iadarola, "Xobjects and Xdeps: low-level libraries empowering beam dynamics simulations," presented at HB'23, Geneva, Switzerland, Oct. 2023, paper THBP32.

[12] É. Forest, *Beam Dynamics: A New Attitude and Framework*. Hardwood Academic / CRC Press, 1998, vol. 8.

[13] G. Ripken and F. Schmidt, "A symplectic six-dimensional thin-lens formalism for tracking," CERN, Tech. Rep. CERN-SL-95-12, CERN-SL-95-12-AP, DESY-95-063, 1995. https://cds.cern.ch/record/281283

[14] L. Deniau, *MAD-NG Reference Manual*, 2023. https://cern.ch/mad/releases/madng/html

[15] P. D. Hermes *et al.*, "A novel tool for beam dynamics studies with hollow electron lenses," in *Proc. IPAC'22*, Bangkok, Thailand, 2022, pp. 176–179. doi:10.18429/JACoW-IPAC2022-MOPOST045

[16] *LHC@home website*, 2023. https://lhcathome.cern.ch/lhcathome/

[17] P. Niedermayer *et al.*, "Investigation of micro spill in RF KO extraction using tailored excitation signals," in *Proc. IPAC'23*, Venice, Italy, 2023, pp. 2427–2430. doi:10.18429/JACoW-IPAC2023-TUPM094

[18] F. Kuehteubl *et al.*, "Investigating alternative extraction methods at MedAustron," in *Proc. IPAC'23*, Venice, Italy, 2023, pp. 2419–2422. doi:10.18429/JACoW-IPAC2023-TUPM091

[19] J. Yang *et al.*, "Study on spill quality and transit times for slow extraction from SIS18," in *Proc. IPAC'23*, Venice, Italy, 2023, pp. 2435–2438. doi:10.18429/JACoW-IPAC2023-TUPM097

[20] P. Arrutia Sota *et al.*, "RF techniques for spill quality improvement in the SPS," in *Proc. IPAC'23*, Venice, Italy, 2023, pp. 319–322. doi:10.18429/JACoW-IPAC2023-MOPA116

[21] T. Pugnat, M. Giovannozzi, F. F. Van der Veken, and D. Di Croce, "Analysis tools for numerical simulations of dynamic aperture with Xsuite," presented at HB'23, Geneva, Switzerland, Oct. 2023, paper THBP35.

[22] M. Rakic *et al.*, "Commissioning strategies of hollow electron lens residual kick compensation," in *Proc. IPAC'23*, Venice, Italy, 2023, pp. 590–593. doi:10.18429/JACoW-IPAC2023-MOPL031

[23] C. E. Montanari, A. Bazzani, M. Giovannozzi, and G. Turchetti, "Using Dynamic Indicators for Probing Single-Particle Stability in Circular Accelerators," in *Proc. IPAC'22*, Bangkok, Thailand, 2022, pp. 168–171. doi:10.18429/JACoW-IPAC2022-MOPOST042

[24] F. F. V. der Veken *et al.*, "Determination of the Phase-Space Stability Border with Machine Learning Techniques," in *Proc. IPAC'22*, Bangkok, Thailand, 2022, pp. 183–186. doi:10.18429/JACoW-IPAC2022-MOPOST047

[25] A. Fornara, G. Sterbini, and R. Appleby, "Impact of crab cavity RF noise on the transverse beam profiles in the HL-LHC," in *Proc. IPAC'23*, Venice, Italy, 2023, pp. 3391–3394. doi:10.18429/JACoW-IPAC2023-WEPL119

[26] A. Wolski, *Beam Dynamics in High Energy Particle Accelerators*. World Scientific, 2023. doi:10.1142/13333

[27] F. Willeke and G. Ripken, "Methods of beam optics," Tech. Rep. DESY 88-114, 1988. https://cds.cern.ch/record/194174

[28] R. De Maria, F. Schmidt, and P. K. Skowronski, "Advances in matching with MAD-X.," in *Proc. ICAP'06*, Chamonix, Switzerland, Oct. 2006, pp. 213–215. https://jacow.org/icap06/papers/WEPPP14.pdf

[29] B. Lindstrom *et al.*, "Mitigating collimation impedance and improving halo cleaning with new optics and settings strategy of the HL-LHC betatron collimation system," presented at HB'23, Geneva, Switzerland, Oct. 2023, paper TUC4C2.

ISBN: 978-3-95450-253-0 ISSN: 2673-5571 doi:10.18429/JACoW-HB2023-TUA2I1

[30] H. Burkhardt, "Monte Carlo generator for synchrotron radiation," CERN, Tech. Rep., 1990. https://cds.cern.ch/record/443490

[31] A. W. Chao, "Evaluation of beam distribution parameters in an electron storage ring," *J. Appl. Phys.*, vol. 50, pp. 595–598, 2008. doi:10.1063/1.326070

[32] E. Forest, *From tracking code to analysis: generalised Courant-Snyder theory for any accelerator model*. Springer Publishing Company, Incorporated, 2016.

[33] D. Demetriadou *et al.*, "Tools for integrated simulation of collimation processes in Xsuite," in *Proc. IPAC'23*, Venice, Italy, 2023, pp. 2801–2804. doi:10.18429/JACoW-IPAC2023-WEPA066

[34] *Geant4 documentation*, 2023. https://geant4.web.cern.ch

[35] A. Abramov *et al.*, "Development of collimation simulations for the FCC-ee," in *Proc. IPAC'22*, Bangkok, Thailand, 2022, pp. 1718–1721. doi:10.18429/JACoW-IPAC2022-WEPOST016

[36] L. Nevay *et al.*, "BDSIM: An accelerator tracking code with particle–matter interactions," *Comput. Phys. Commun.*, vol. 252, p. 107 200, 2020. doi:https://doi.org/10.1016/j.cpc.2020.107200

[37] *FLUKA documentation*, 2023. https://fluka.cern

[38] J. Qiang, M. A. Furman, and R. D. Ryne, "A parallel particle-in-cell model for beam–beam interaction in high energy ring colliders," *J. Comput. Phys.*, vol. 198, no. 1, pp. 278–294, 2004. doi:https://doi.org/10.1016/j.jcp.2004.01.008

[39] A. Oeftiger and S. Hegglin, "Space charge modules for PyHEADTAIL," in *Proc. HB'16*, Malmö, Sweden, Jul. 2016, pp. 124–129. doi:10.18429/JACoW-HB2016-MOPR025

[40] K. Hirata, "Don't be afraid of beam-beam interactions with a large crossing angle," Tech. Rep. SLAC-PUB-6375, 1994. https://cds.cern.ch/record/271786

[41] G. Sterbini *et al.*, "Potential and constraints of a beam-beam wire compensator in the HL-LHC era," in *Proc. IPAC'23*, Venice, Italy, 2023, pp. 3348–3351. doi:10.18429/JACoW-IPAC2023-WEPL103

[42] S. V. Furuseth and X. Buffat, "Parallel high-performance multi-beam multi-bunch simulations," *Comput. Phys. Commun.*, vol. 244, pp. 180–186, 2019. doi:https://doi.org/10.1016/j.cpc.2019.06.006

[43] P. Kicsiny, X. Buffat, G. Iadarola, T. Pieloni, D. Schulte, and M. Seidel, "Towards beam-beam simulations for FCC-ee," in *Proc. 65th ICFA Adv. Beam Dyn. Workshop High Luminosity Circular e+e- Colliders (eeFACT'22)*, Frascati, Italy, 2022, paper WEZAT0102, pp. 165–170. doi:10.18429/JACoW-eeFACT2022-WEZAT0102

[44] P. Kicsiny *et al.*, "Bhabha scattering model for multi-turn tracking simulations at the FCC-ee," in *Proc. IPAC'23*, Venice, Italy, 2023, pp. 682–685. doi:10.18429/JACoW-IPAC2023-MOPL062

[45] S. Kostoglou *et al.*, "Dynamic aperture studies for the first run of High Luminosity LHC," in *Proc. IPAC'23*, Venice, Italy, 2023, pp. 3344–3347. doi:10.18429/JACoW-IPAC2023-WEPL102

[46] P. Kicsiny *et al.*, "Benchmark and performance of beam-beam interaction models for Xsuite," in *Proc. IPAC'23*, Venice, Italy, 2023, pp. 686–689. doi:10.18429/JACoW-IPAC2023-MOPL063

[47] K. Paraschou and G. Iadarola, "Incoherent electron cloud effects in the Large Hadron Collider," in *Proc. ICFA mini-Workshop on Mitigation of Coherent Beam Instabilities in Particle Accelerators*, Zermatt, Switzerland, 2020, pp. 249–249. doi:10.23732/CYRCP-2020-009.249

[48] K. Paraschou, "Studies of incoherent effects for the upgrade of the large hardon collider and detector applications," Ph.D. dissertation, Aristotle U., Thessaloniki, 2023. doi:10.26262/heal.auth.ir.348933

[49] K. Paraschou and G. Iadarola, "A method for accurate and efficient simulations of slow beam degradation due to electron clouds," 2023. doi:10.48550/arXiv.2302.10581