15th International Particle Accelerator Conference,Nashville, TN JACoW Publishing
ISBN: 978-3-95450-247-9 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2024-WEPR56

# Xsuite: AN INTEGRATED BEAM PHYSICS SIMULATION FRAMEWORK

G. Iadarola*, R. De Maria, S. Łopaciuk, A. Abramov, X. Buffat, D. Demetriadou,
L. Deniau, C. Droin, J. Dilly, P. Hermes, P. Kicsiny, S. Kostoglou, P. Kruyt, A. Latina,
L. Mether, C. E. Montanari, K. Paraschou, T. Prebibaj, F. Soubelet, G. Sterbini,
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, EPFL, Lausanne, Switzerland

## *Abstract*

Xsuite is a modular simulation package bringing to a single flexible and modern framework capabilities of different tools developed at CERN in the past decades, notably MAD-X, 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. Different computing platforms are supported, including conventional CPUs, as well as GPUs from different vendors. 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. Methods are made available to compute and optimize the accelerator lattice functions, chromatic properties, equilibrium beam sizes. By now the tool has reached a mature stage of development and is used for simulations studies by a large and diverse user community.

## INTRODUCTION

CERN has a long tradition in the development of software tools for beam physics in circular accelerators, having provided to accelerator community MAD-X [1], Sixtrack [2], Sixtracklib [3], COMBI [4], PyHEADTAIL [5]. These tools were developed over several years, mostly by independent teams and, although they provide very advanced features in their respective domains, their design does not allow effectively combining them for integrated simulations involving complex heterogeneous effects. 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, as is 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. Furthermore, several of these simulations are very well suited for computation acceleration based on Graphics Processing Units (GPUs). However, it would be cumbersome to retrofit such a capability in the existing codes.

Based on these considerations, in 2021 the Xsuite project [6] 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 quite a large user community. In this contribution, we describe the overall code structure and development strategy and then illustrate the main features and applications of Xsuite. Further information can be found in Ref. [7] and references therein.

## 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;
- **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 Ref. [8]);
- **Xdeps**: manages tasks and deferred expressions for modeling and updating of accelerator circuits, provides a multi-objective optimizer (see also Ref. [9]) and a general purpose text-based tabular data explorer;
- **Xobjects**: provides the low-level infrastructure for memory management and multi-platform code compilation and execution (see also Ref. [9]).

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 main-

---

* giovanni.iadarola@cern.ch

tained by a small core team integrating contributions from a wider developer community.

Since the early stages, the developers have 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. 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.

## 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 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, which are more accurate when simulating small accelerators with large magnet curvatures [10].

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 [11].

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.). 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 Ref. [9]). 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.

### *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 [12]. 4. Particles tracking is used to propagate the eigenvectors along the beam line; 6. Twiss parameters ($\alpha$, $\beta$, $\gamma$), dispersion functions, phase advances, as well as the effect of linear coupling are obtained using the Mais-Ripken approach [13]. The accuracy of such a method is found to be excellent for all accelerators tested so far. 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.

Truncated Power Series Algebra (TPSA) computations and non-linear normal form analysis are made available by interfacing to the MAD-NG code [14].

### *Optimizer*

Accelerator design and simulations often require solving optimization problems, for example to control the lattice functions (optics matching) or to correct the model. 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 MAD-X [15], which has proven over the years to be very well suited for accelerator design and tuning. It is possible to perform optimizations involving targets and knobs from multiple beam lines and, 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 (an example of this capability can be found in Ref. [16]).

### *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, i.e., the "mean" model, for which the energy loss from the radiation is applied particle by particle without accounting for quantum fluctuations, and the "quantum" model for which the actual photon emission is simulated including quantum fluctuations, using the algorithm described in Ref. [17].

When synchrotron radiation is present, additional calculations can be enabled in the Twiss calculation. 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 of the one-turn matrix, and the equilibrium emittances are calculated from the lattice linear normal form following the approaches described in Refs. [18] and [19]. 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 [19].

In high-energy lepton rings, the beam loses 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"). Xsuite provides an automatic iterative method to perform these two corrections, which are interdependent and cannot be done sequentially. For more details on radiation modeling and compensation see Ref. [7].

### *Particle-matter Interaction and Collimation Studies*

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 [8]. The interaction can be simulated using different engines, namely the "Everest" engine embedded in Xcoll, which is an evolution of the K2 model developed for Sixtrack [20]; the "Geant 4" engine, which exploits an interface between Xsuite and the Geant4 library [21], built through the BDSIM library [22, 23]; and the "FLUKA" engine, interfacing with the FLUKA Monte Carlo code [24].

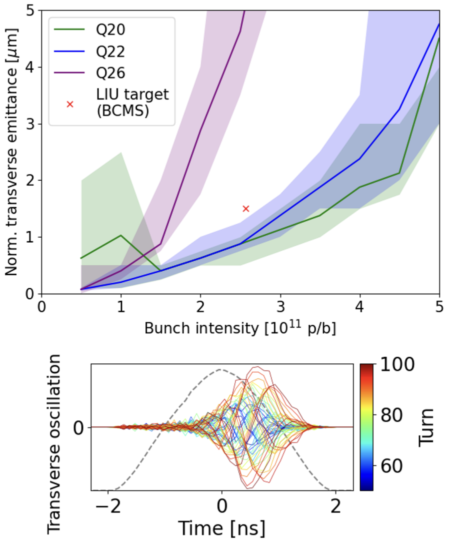
An important goal of simulation studies for collimation is the precise localization of the beam losses along the accelerator, to estimate the power deposition on accelerator components in order, for example, to study equipment activation or quench limits. The aperture model of the accelerator can be imported by Xsuite as part of the MAD-X sequence and the particle loss localization is refined in a post-processing stage to reach the accuracy set by the user, typically 1 to 10 cm.

### *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. 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.

The effect of beam space charge can be included in the simulation choosing among different space-charge models including tracking through "frozen" distributions, as well as self-consistent "Particle In Cell (PIC)" computations. Space charge simulations strongly profit from the speed up provided by GPUs. Figure 1 shows the outcome of a study on beam stability for the CERN SPS based including the effect of space charge (PIC model) and wakefields. In this case, the GPU-accelerated simulation was found to be more than 100 times faster compared to the serial CPU implementation.

Similarly to the Sixtrack and COMBI simulation codes, Xsuite provides different models for the simulation of beam-beam effects in colliders, including a "6D" model, which applies longitudinal and transverse forces accounting for the longitudinal coordinates of the particles, using the approach described in Ref. [25]. All models can be used either in "weak-strong" mode, simulating the interaction of the beam particles with a fixed charge distribution, 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). The simulation of beam-beam compensation with wires as well as beamstrahlung and Bhabha scattering are also implemented [26–28].



| | |
|---|---|
| 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 1: Simulation study for the CERN SPS including the full non-linear lattice, space charge (PIC) and wakefields. Top: obtained instability thresholds for different beam optics; middle: one of the simulated instabilities; bottom: some relevant parameters.

Xsuite has been exploited to study the effect of electron cloud on slow beam degradation (emittance growth, losses), as described in Ref. [29]. The simulation of wakefields and transverse feedback systems can be performed by inserting the corresponding PyHEADTAIL elements into the Xsuite line. An interface to the RFTrack tracking code is presently being developed [30]. Methods to compute growth rate from Intrabeam Scattering have also been recently added [31].

## ACKNOWLEDGMENTS

The authors would like to thank H. Bartosik, S. Fartoukh, A. Fornara, M. Hofer, M. Giovannozzi, S. Kostoglou, P. Kruyt, B. Lindstrom, Y. Papaphilippou, T. Pugnat, S. Redaelli, G. Rumolo, R. Tomas for their important support. Work supported by the Swiss accelerator research and technology (CHART).

## REFERENCES

[1] R. D. 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," pp. 3200–3203, 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, Brazil, May 2021, 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. 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] Xsuite. http://xsuite.web.cern.ch

[7] G. Iadarola *et al.*, "Xsuite: An Integrated Beam Physics Simulation Framework," in *Proc. HB'23*, Geneva, Switzerland, 2024, pp. 73–80. doi:10.18429/JACoW-HB2023-TUA2I1

[8] F. F. V. der Veken *et al.*, "Recent Developments with the New Tools for Collimation Simulations in Xsuite," in *Proc. HB'23*, Geneva, Switzerland, 2024, pp. 474–478. doi:10.18429/JACoW-HB2023-THBP13

[9] S. Łopaciuk, R. D. Maria, and G. Iadarola, "Xobjects and Xdeps: Low-Level Libraries Empowering Beam Dynamics Simulations," in *Proc. HB'23*, Geneva, Switzerland, 2024, pp. 543–546. doi:10.18429/JACoW-HB2023-THBP32

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

[11] P. D. Hermes *et al.*, "A Novel Tool for Beam Dynamics Studies with Hollow Electron Lenses," no. 13, pp. 176–179, 2022. doi:10.18429/JACoW-IPAC2022-MOPOST045

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

[13] F. Willeke and G. Ripken, "Methods of Beam Optics," *AIP Conf. Proc.*, vol. 184, pp. 758–819, 2008. doi:10.1063/1.38050

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

[15] 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.

[16] B. Lindström *et al.*, "Mitigating Collimation Impedance and Improving Halo Cleaning with New Optics and Settings Strategy of the HL-LHC Betatron Collimation System," in *Proc. HB'23*, Geneva, Switzerland, 2024, pp. 183–187. doi:10.18429/JACoW-HB2023-TUC4C2

[17] H. Burkhardt, "Monte Carlo Generator for Synchrotron Radiation," CERN, Tech. Rep., 1990. https://cds.cern.ch/record/443490

[18] A. W. Chao, "Evaluation of Beam Distribution Parameters in an Electron Storage Ring," *Journal of Applied Physics*, vol. 50, no. 2, pp. 595–598, 2008. doi:10.1063/1.326070

[19] E. Forest, *From Tracking Code to Analysis: Generalised Courant-Snyder Theory for Any Accelerator Model*. Springer Tokyo, 2016. doi:10.1007/978-4-431-55803-3

[20] D. Demetriadou, A. Abramov, G. Iadarola, and F. V. der Veken, "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

[21] Geant4. https://geant4.web.cern.ch

[22] 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

[23] L. Nevay *et al.*, "BDSIM: An Accelerator Tracking Code with Particle–Matter Interactions," *Computer Physics Communications*, vol. 252, p. 107 200, 2020. doi:10.1016/j.cpc.2020.107200

[24] FLUKA. https://fluka.cern

[25] 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

[26] G. Sterbini, A. Rossi, A. Bertarelli, A. Poyet, P. Belanger, and Y. Papaphilippou, "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

[27] P. Kicsiny, X. Buffat, G. Iadarola, T. Pieloni, D. Schulte, and M. Seidel, "Towards Beam-Beam Simulations for FCC-ee," in *Proc. eeFACT'22*, Frascati, Italy, 2022. doi:10.18429/JACoW-eeFACT2022-WEZAT0102

[28] P. Kicsiny, X. Buffat, G. Iadarola, T. Pieloni, D. Schulte, and M. Seidel, "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

[29] K. Paraschou, G. Iadarola, and L. Mether, "Simulations of incoherent effects driven by electron clouds forming in the inner triplets of the Large Hadron Collider," presented at IPAC'24, Nashville, TN, USA, May 2024, paper WEPR57, this conference.

[30] A. Latina, "The Tracking Code RF-Track and Its Application," no. 68, pp. 245–248, 2024. doi:10.18429/JACoW-HB2023-WEA3C1

[31] F. Soubelet, H. Bartosik, S. Kostoglou, K. Paraschou, and M. Zampetakis, "Development of Numerical Tools for Intra-Beam Scattering Modelling," presented at IPAC'24, Nashville, TN, USA, May 2024, paper MOPS37, this conference.