ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-WEPA066

TOOLS FOR INTEGRATED SIMULATION OF COLLIMATION PROCESSES IN XSUITE

D. Demetriadou*, A. Abramov, G. Iadarola, F. F. Van der Veken, CERN, Geneva, Switzerland

Abstract

The existing code for particle scattering and tracking in collimation systems integrated in SixTrack, called K2, was migrated from the current software in FORTRAN, to a new Python/C interface integrated in the Xsuite tracking code that is being developed at CERN. This is an essential step towards a full integration of collimation studies using Xtrack, and will allow profiting from GPU computing advances and the BOINC volunteer computing network. Furthermore, several improvements to the functionality of the code were introduced, for example aperture interpolation for more precise longitudinal location of particle losses. A thorough testing of the new implementation was performed, using as case studies various collimation layout configurations for the LHC Run 3 and HL-LHC. In this paper, the challenges are outlined and the first results are presented, including simulated loss maps which are compared to the reference results generated by SixTrack.

INTRODUCTION

One of the most well-known tools for 6D single-particle symplectic tracking simulations is SixTrack [1–7], which has been used extensively in the last few decennia to study dynamic aperture simulations. SixTrack has likewise been an important tool to evaluate the performance of beam-intercepting devices like collimators, using its integrated software called K2 to simulate proton-matter interactions [8–10], its coupling to the FLUKA particle physics Monte Carlo simulation package [11–13], and its BDSIM coupling to the Geant4 toolkit for the simulation of the passage of particles through matter [14].

Recently, a new modular simulation package called Xsuite started being developed, combining in a single flexible and modern framework the capabilities of different tools developed at CERN in the past decades, notably Sixtrack, Sixtracklib [6, 15], COMBI [16], and PyHEADTAIL [17]. The suite is made of a set of Python modules 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 modelling of the particle dynamics, combined with the effect of synchrotron radiation, impedances, feedbacks, space charge, electron cloud, beam-beam, beamstrahlung, and electron lenses. 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 Graphic Processing Units (GPUs) from different vendors.

* despina.demetriadou@cern.ch

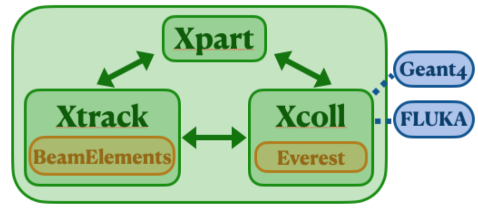

Figure 1: Integration of the Xcoll package into Xsuite. For collimation simulations, we use a tight interplay with Xtrack and Xpart. In green is the API code in Python, in yellow is the low-level tracking and scattering code in C, and in blue is the coupling (in progress) to scattering codes in other programming languages.

XCOLL

The capability to perform collimation studies in Xsuite is given by the Xcoll Python package –still in active development like all of Xsuite– which aims to reproduce and expand the collimation functionalities provided by SixTrack. Simulations involving collimators that intercept a particle beam differ from tracking simulations only through magnetic elements, for several reasons [18–20]. First of all, once a tracked particle is intercepted by a collimator it undergoes a very different type of physics, namely the interaction between a high-energy beam and stationary matter which depends heavily on the material properties of the collimator. Similarly, when a tracked particle is intercepted by a crystal collimator, it undergoes specific processes related to crystal physics such as channelling [21–24]. For both of these we need a dedicated implementation, either internally like K2 in SixTrack, or coupled to an external tool like FLUKA or Geant4. In Xcoll, the internal implementation is translated from the original K2 code into C, called Everest. This is illustrated in Fig. 1.

Furthermore, the jaw opening of a real collimator is typically expressed as a multiple of the transverse beam size at that location. Next, to reduce the amount of CPU-time needed in tracking, the initial conditions of a particle beam are typically instantiated at a given collimator and approximated by a simplified distribution such as an annular halo or a pencil beam [19]. Finally, the simulation code needs to keep track of absorption and scattering locations. In general, several important developments in particle tracking codes, like the pencil beam and the aperture tools, were steered by their necessity in collimation simulations.

The Xcoll API provides a `CollimatorManager` class that handles all settings and functionalities in a collimation simulation. After loading an accelerator lattice with Xtrack, the manager provides functions to manually insert collimator elements in the lattice, or to load a collimator database from a file –where both the custom SixTrack for-

ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-WEPA066

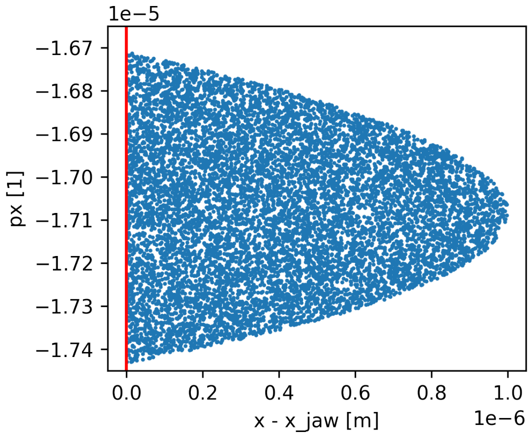

Figure 2: Initial conditions: direct halo on one jaw.

mat or a dictionary/yaml/json format are supported– which contains the information to install a full set of collimators at once. In the current release, three collimator elements are available: a `BlackAbsorber` that is an idealised collimator without scattering, and an `EverestCollimator` and an `EverestCrystal` which uses the successor of the K2 scattering code. Next, the manager converts the collimator jaw settings from the values given in units of the local optics to a value in physical units, similarly to SixTrack.

The next step is to define a set of initial conditions for the simulation. It is noted that those functionalities in SixTrack were part of the collimation implementation, while an effort has been made in Xsuite to generalise these features that can have a more generic use. The Xcoll manager relies on, and expands, the functionality in Xpart in order to provide several options for the initial distribution. First, there is the possibility to generate an annular halo, which is a uniformly-populated thin strip of the matched phase space ellipse around a given normalised amplitude. This amplitude is typically set just above the jaw opening of the primary collimator that is to be intercepting the beam. The starting location of tracking can be chosen freely, but particles might need several turns before they are intercepted. Second, there is a direct halo implementation, which is like the annular halo but at the longitudinal position of the primary collimator, and the halo is cut such that only particles are generated that will hit the collimator in the first turn. This is demonstrated in Fig. 2, which shows a zoom of such a distribution on one jaw. Third, the contribution of the particle's off-momentum to the transversal position (from the local dispersion function) can be controlled as well, to be able to create particles with arbitrary off-momentum at a given transversal position. This functionality can be combined with the previous two, in order to add an off-momentum contribution to the halo, but keep the matching to the normalised collimator amplitude. Finally, custom distributions can be easily constructed using Xpart and even imported from files.

TRACKING AND INTERPOLATION

Tracking is performed using regular function calls in Xtrack, and the integration with the Xcoll routines will ensure that whenever a particle hits a collimator, the relevant scattering routines are called. If the result of scattering is anything except absorption, tracking continues. Furthermore, Xcoll keeps a record of every particle-matter interaction that took place, to be able to construct a list of absorbed particles and aperture losses, as well as to inspect the behaviour of specific collimators. This procedure allows to perform loss maps simulations, where the distribution of losses along the accelerator is investigated. Both betatron and off-momentum loss maps can be simulated this way, where the latter have historically been complex to set up [25, 26]. As a first approximation, one can use the equivalent of a pencil beam in the longitudinal plane. This can be achieved by controlling the off-momentum as mentioned above, which has an option to give a purely off-momentum contribution to the initial particles hitting the collimator jaw, hence without any betatron contribution.

To recreate a more realistic scenario of an off-momentum loss map, we cannot just add a frequency shift to the cavities as tracking codes typically assume that particles are synchronous to the longitudinal reference trajectory; one has to implement a change in length of the reference trajectory as well. In SixTrack this can be achieved with the DYNK block [4], while in Xtrack this can applied to the elements directly. The Xcoll manager provides an easy function to achieve this, which essentially moves the separatrix in the longitudinal phase space adiabatically up or down along the off-momentum axis, until the off-momentum of the bunch is so large that particles are naturally intercepted by the off-momentum collimators. Similarly, the manager provides the possibility to perform a realistic betatron loss map, by gradually blowing up a bunch until most of it is lost on the betatron collimators.

Last, the Xcoll manager provides functions to generate a summary of the losses on the collimators as well as a dictionary of all losses and their locations, which then can be analysed and plotted for presentation with external tools. One additional advantage of having the Xcoll API in Python, is that one of these analysis tools exists in Python as well,

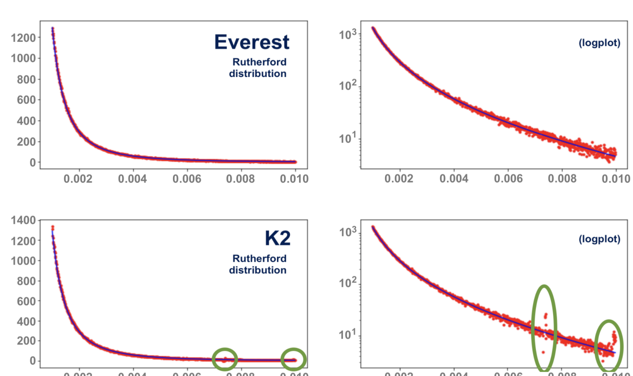

Figure 3: PDF of a Rutherford distribution.

ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-WEPA066

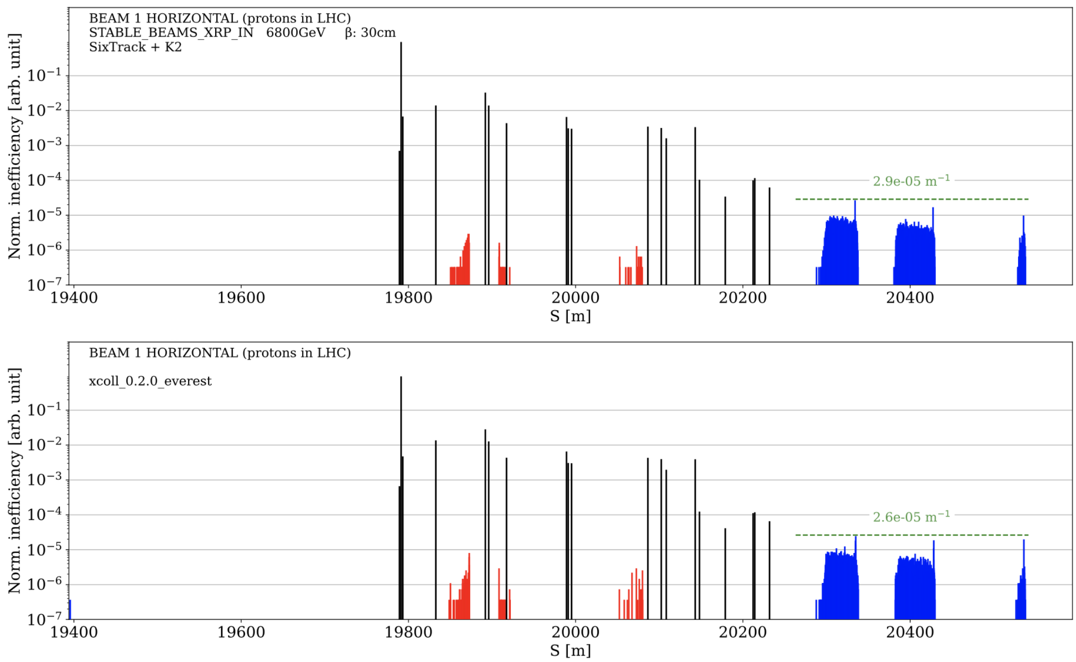

Figure 4: Comparison between a simulated B1H loss map for the LHC in SixTrack with K2 (top) and Xsuite with Xcoll (bottom). Shown is a zoom on the IR7 insertion containing the collimators for betatron cleaning. Red and blue lines represent aperture losses in warm resp. cold regions, and black lines represent losses on the collimators. The values are scaled by length/binning and normalised by the losses on the primary collimator.

and the full workflow of a loss map simulation can hence be combined in a single script.

An essential ingredient for loss map simulations, is the possibility to interpolate the aperture at a fixed interval. This was a part of the workflow for SixTrack (K2 or FLUKA), and has been implemented in Xtrack for a more generic usage, and has been tested thoroughly.

EVEREST

While the integration of the coupling routines to FLUKA and Geant4 is still in process, the translation of K2 from its native FORTRAN code in SixTrack towards a C implementation compatible with Xsuite is now in a working state. The new routine is named Everest, and covers all functionalities from the original code, except the jaw flatness which will be ported at a later stage, while new features have been added such as the possibility to specify tilts to collimator jaws and to define a custom collimator material.

To simplify the procedure, the code was first translated from FORTRAN to Python, and then from there to C. In between, the partially-translated code was regularly tested against K2 to be sure no hidden bugs would be introduced. As a Monte Carlo scattering code, both K2 and Everest depend of course on a random generator. Once this random generator was ported to C as well, Everest was no longer reproducible against K2 for individual-particle tracking. Special care was hence taken to deeply test the statistical properties of the new random generator, with the requirement to match or outperform the existing one in K2. Indeed, not only proved the new random generator to be faster, it also showed a higher statistical accuracy. Where the K2 random generator suffered from certain artefacts due to internal rounding errors, the new random generator does not. This is illustrated in Fig. 3, where the PDF of a Rutherford random distribution is shown, and some artefacts are clearly visible towards high values. Note that this did not impact the quality of previous simulations performed with SixTrack and K2, as the statistical probability of these artefacts is extremely low. After the translation of the random generator was completed, new reference results were generated to benchmark the new code against during the rest of the porting process.

After the implementation of Everest was completed, betatron loss map simulations on the LHC Run 3 lattice were performed and compared to K2. These were performed using an initial pencil beam of 20 million particles, both horizontal and vertical loss maps, and for both beams. The case for B1H is illustrated in Fig. 4, and shows a very close agreement between both.

CONCLUSION

A new software package for collimation simulations Xcoll, has been developed and integrated into the new Xsuite environment. In a short time, we recovered nearly the full functionalities included in SixTrack over many years. Furthermore, the built-in engine for high-energy proton-matter interactions, K2, has been translated from FORTRAN into C and thoroughly tested. The results are very satisfying, and new development is on-going.

ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-WEPA066

REFERENCES

[1] F. Schmidt, "SixTrack. User's Reference Manual," CERN, Geneva, Switzerland, Tech. Rep. CERN/SL/94-56-AP, 1994.

[2] R. De Maria *et al.*, "Recent Developments and Future Plans for SixTrack," in *Proc. IPAC'13*, Shanghai, China, May 2013, pp. 948–950. https://jacow.org/IPAC2013/papers/MOPWO028.pdf

[3] R. De Maria, A. Mereghetti, J. Molson, and K. Sjobak, "SixTrack status," in *Proc. of the ICFA Mini-Workshop on Tracking for Collimation*, CERN, Geneva, Switzerland, 2018, pp. 11–15. doi:10.23732/CYRCP-2018-002.11

[4] K. Sjøbak *et al.*, "Dynamic simulations in SixTrack," in *Proc. of the ICFA Mini-Workshop on Tracking for Collimation*, CERN, Geneva, Switzerland, 2018, pp. 123–133. doi:10.23732/CYRCP-2018-002.123

[5] R. De Maria *et al.*, "SixTrack Version 5: Status and New Developments," in *Proc. IPAC'19*, Melbourne, Australia, May 2019, pp. 3200–3203. doi:10.18429/JACoW-IPAC2019-WEPTS043

[6] R. De Maria *et al.*, "SixTrack V and runtime environment," *Int. J. Mod. Phys. A*, vol. 34, no. 36, p. 1 942 035, 2020. doi:10.1142/S0217751X19420351

[7] *Sixtrack web site*, http://sixtrack.web.cern.ch/SixTrack/.

[8] G. Robert-Demolaize, R. W. Assmann, S. Redaelli, and F. Schmidt, "A New Version of SixTrack with Collimation and Aperture Interface," in *Proc. PAC'05*, Knoxville, TN, USA, May 2005, pp. 4084–4086. https://jacow.org/p05/papers/FPAT081.pdf

[9] E. Quaranta, A. Valloni, R. Bruce, A. Mereghetti, and S. Redaelli, "Updated implementation of collimator materials in SixTrack and MERLIN codes," in *Proc. of the ICFA Mini-Workshop on Tracking for Collimation*, CERN, Geneva, Switzerland, 2018, pp. 109–122. doi:10.23732/CYRCP-2018-002.109

[10] R. Bruce *et al.*, "Status of SixTrack with collimation," in *Proc. of the ICFA Mini-Workshop on Tracking for Collimation*, CERN, Geneva, Switzerland, 2018, pp. 1–10. doi:10.23732/CYRCP-2018-002.1

[11] A. Mereghetti *et al.*, "SixTrack-Fluka Active Coupling for the Upgrade of the SPS Scrapers," in *Proc. IPAC'13*, Shanghai, China, May 2013, pp. 2657–2659. https://jacow.org/IPAC2013/papers/WEPEA064.pdf

[12] E. Skordis *et al.*, "FLUKA coupling to Sixtrack," in *Proc. of the ICFA Mini-Workshop on Tracking for Collimation*, CERN, Geneva, Switzerland, 2018, pp. 17–25. doi:10.23732/CYRCP-2018-002.17

[13] P. D. Hermes *et al.*, "Simulation of Heavy-Ion Beam Losses with the SixTrack-FLUKA Active Coupling," in *Proc. IPAC'16*, Busan, Korea, May 2016, pp. 2490–2493. doi:10.18429/JACoW-IPAC2016-WEPMW029

[14] L. Nevay *et al.*, "LHC Collimation and Energy Deposition Studies Using Beam Delivery Simulation (BDSIM)," in *Proc. IPAC'16*, Busan, Korea, May 2016, pp. 3101–3103. doi:10.18429/JACoW-IPAC2016-WEPOY047

[15] 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

[16] S. V. Furuseth and X. Buffat, "High-Performance Scheduling of Multi-Beam Multi-Bunch Simulations," in *Proc. IPAC'19*, Melbourne, Australia, May 2019, pp. 3208–3211. doi:10.18429/JACoW-IPAC2019-WEPTS045

[17] A. Oeftiger, "An Overview of PyHEADTAIL," CERN, Geneva, Switzerland, Tech. Rep. CERN-ACC-NOTE-2019-0013, 2019. https://cds.cern.ch/record/2672381

[18] R. Bruce *et al.*, "Simulations and Measurements of Cleaning with 100 MJ Beams in the LHC," in *Proc. IPAC'13*, Shanghai, China, May 2013, pp. 52–54. https://jacow.org/IPAC2013/papers/MOODB202.pdf

[19] R. Bruce *et al.*, "Simulations and measurements of beam loss patterns at the CERN Large Hadron Collider," *Phys. Rev. ST Accel. Beams*, vol. 17, p. 081 004, 8 2014. doi:10.1103/PhysRevSTAB.17.081004

[20] A. Mereghetti *et al.*, "SixTrack for Cleaning Studies: 2017 Updates," in *Proc. IPAC'17*, Copenhagen, Denmark, May 2017, pp. 3811–3814. doi:10.18429/JACoW-IPAC2017-THPAB046

[21] D. Mirarchi, S. Redaelli, W. Scandale, A. M. Taratin, and I. A. Yazynin, "Improvements of the Crystal Routine for Collimation Studies," in *Proc. IPAC'14*, Dresden, Germany, Jun. 2014, pp. 886–889. doi:10.18429/JACoW-IPAC2014-MOPRI111

[22] D. Mirarchi, "Crystal Collimation for LHC," Ph.D. dissertation, Imperial College, London, 2015. http://cds.cern.ch/record/2036210

[23] D. Mirarchi, G. Hall, S. Redaelli, and W. Scandale, "A crystal routine for collimation studies in circular proton accelerators," *Nucl. Instrum. Methods Phys. Res., B*, vol. 355, pp. 378–382, 2015. doi:10.1016/j.nimb.2015.03.026

[24] D. Mirarchi, S. Redaelli, and W. Scandale, "Crystal implementation in SixTrack for proton beams," in *Proc. of the ICFA Mini-Workshop on Tracking for Collimation*, CERN, Geneva, Switzerland, 2018, pp. 91–108. doi:10.23732/CYRCP-2018-002.91

[25] E. Quaranta, R. Bruce, and S. Redaelli, "SixTrack Simulation of Off-momentum Cleaning in LHC," in *Proc. IPAC'13*, Shanghai, China, May 2013, pp. 972–974. https://jacow.org/IPAC2013/papers/MOPWO037.pdf

[26] H. Garcia-Morales, R. Bruce, D. Mirarchi, S. Redaelli, B. Salvachua, and K. Sjobak, "Simulating off-momentum loss maps using SixTrack," in *Proc. of the ICFA Mini-Workshop on Tracking for Collimation*, CERN, Geneva, Switzerland, 2018, pp. 163–170. doi:10.23732/CYRCP-2018-002.163