BENCHMARKING BEAM DYNAMICS SIMULATIONS FOR THE MYRRHA PHASE 1 ACCELERATOR

L. De Keukeleere†, U. Dorda, J. Hustings, E. Tanke, J. Van De Walle
Belgian Nuclear Research Centre, Mol, Belgium

Abstract

At the Nuclear Research Center SCK CEN in Belgium, the first phase of the MYRRHA project (an accelerator driven system) is under construction. Included in MYRRHA Phase 1 are a 17 MeV normal conducting injector linac (RFQ + CH cavities) and a super-conducting LINAC (60 single-spoke cavities), providing a CW proton beam of 4 mA at 100 MeV. In this contribution, beam dynamics simulations of the MYRRHA Phase 1 accelerator are compared for four different beam dynamics codes: TRACEWIN, PyOrbit3, DYNAC and PyAccel. The beam simulation results for a reference case are compared, as well as the computing times. Furthermore, approximations which can be made to speed up the computation times are discussed. These comparisons are made in view of a future on-line deployment of a simulation tool during beam commissioning.

INTRODUCTION

The last few decades a multitude of beam optics simulation codes were developed within the accelerator community. Despite the many similarities amongst these codes, there are differences in how, for example, space charge and field map tracking are modelled. Furthermore, different software languages or back-ends are deployed, going from Fortran to Python. Hence, the choice does not only depend on the performance, but also on practical considerations like what operating system and software languages will be available in the control room. In this paper four software packages are compared – one of them is tested in two different modes - with respect to beam optics parameters and computing power. All of them are introduced in the next paragraphs.

The first tool considered for developing an online virtual accelerator is *Trace Win* [1], a licensed closed-source software package distributed and maintained by CEA and used as a reference code for projects like IFMIF, ESS and SPIRAL2. It has two tracking 'modes': transfer-matrix tracking and multi-particle tracking. While the former is fast and sufficient for simulating basic beam parameters like envelopes and phase advances, the latter can uncover more detailed behaviour like beam loss and halo formation.

The second tool is the open-source multi-particle tracking code *DYNAC* [2], which contains both a numerical method, capable of simulating single and multi-charge state ion beams in accelerating elements as well as a fast analytical method for single charge state beams. It has been benchmarked against other codes, like TraceWin, as well as against measurements [3, 4]. It is well suited for online modelling and as an aid in view of commissioning [5, 6].

The third tool PyOrbit3, is an open-source Python/C++ implementation of the ORBIT (Objective Ring Beam Injection and Tracking) code [7]. It has multiple space-charge and EM-field tracking models and can be compiled with an MPI for parallel computation. It was already benchmarked and used for large accelerator complexes such as SNS and ESS [8, 9].

The fourth package, *PyAccel,* was developed in-house at SCK CEN and is, like one of the TraceWin modes, a transfer-matrix tracking code, allowing for the fast simulation of collective beam behaviour. It is purely python-based, and the linear space-charge model and EM field dynamics model were adopted from TraceWin. The biggest advantage of having an in-house tool is the level of control and customization.

THE MINERVA ACCELERATOR

The MINERVA accelerator (the layout is shown in Fig. 1) consists of a normal conducting injector followed by a superconducting LINAC. An RFQ installed in the former, bunches a 4mA beam originating from an ECR source at a frequency of 176.1 MHz and accelerates the beam to 1.5 MeV. A sequence of normal-conducting CH cavities then further accelerates the bunched beam to 16.6 MeV. After passing through a matching section, the beam is accelerated to the desired 100 MeV by means of a superconducting LINAC with 60 single spoke cavities pair-wise confined in 30 cryomodules and operating at 352.2 MHz.

In parallel to the requirements tests for MYRRHA [10, 11], two user facilities will be able to receive the beam delivered by High-Energy Beam Transport section (HEBT): The "Proton Target Facility" (PTF) will perform Isotope Separation On-line experiments on non-fissile material targets, while the "Full Power Facility" (FPF) is designed to operate at full CW beam power, i.e. 400 kW, and its main goal is fusion material research [12, 13].

BEAM OPTICS SIMULATIONS

To decide which beam optics codes can be employed to build a virtual accelerator, a comparative study was carried out. The results reported below all correspond to the simulation of one line in the MINERVA accelerator, i.e. going from the exit of the RFQ all the way down to the FPF (see Fig. 1). A qualitative interpretation will be given to the differences amongst the results, as at this point no physical data from beam diagnostics devices are available to compare to.

†lennert.de.keukeleere@sckcen.be

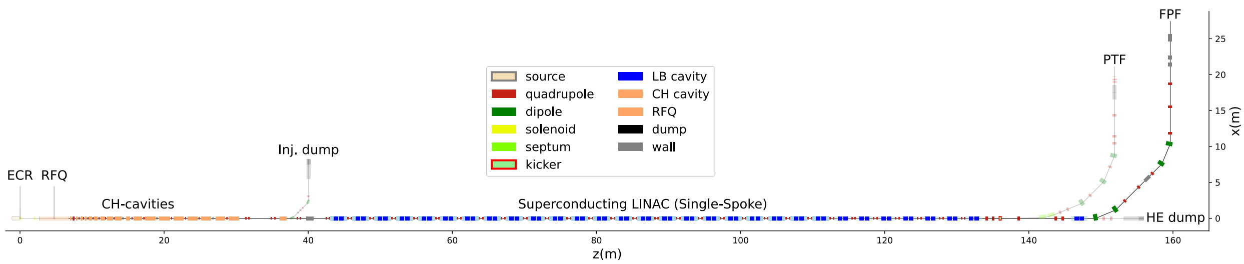

Figure 1: The MINERVA accelerator layout. Sections that were not part of the simulations in this paper are transparent.

Simulation Parameters

Since the codes have multiple models available for space charge and EM field tracking, a choice had to be made. In case of the two transfer-matrix codes, *TraceWin (env)* and *PyAccel*, a linearized space charge model is used where the beam is approximated by a uniform 6D ellipsoid. For the multi-particle mode in TraceWin, *TraceWin (MP),* a 3D Particle-in-cell model was chosen, with a mesh of 18 bins in each direction (x,y,z) and a total size of 9 times the R.M.S. In *DYNAC*, the r-z Particle-In-Cell (PIC) space charge model SCHEFF was used with the default mesh bin sizes (4x R.M.S.) and number of bins in radial (20) and longitudinal (40) direction. Two space charge models were initially tested in *PyOrbit3*, a 3D model using Fast-Fourier Transform (FFT) and a model based on calculating the space charge forces from a uniform ellipse. Both models produce very similar beam optics, not considering the loss profile, where the FFT model predicted considerably higher losses, not in line with the other simulation codes. Furthermore, the uniform ellipse model outperformed the FFT model with regards to computation time. Hence, the choice was made to use the former.

The same one-dimensional (on-axis) EM cavity fields were employed for all codes. In TraceWin, DYNAC and PyOrbit3 an approximate field map is derived from those on-axis fields.

In TraceWin and in DYNAC particles are numerically tracked in the approximate field maps. The numerical method in DYNAC is based on Boole's rule [4]. In PyOrbit3 a three-point Transit-Time Factor (TTF) model takes care of particle tracking [14].

In case of the two transfer-matrix codes, a Taylor expansion down to first order of the field around the theoretical bunch centre was carried out to produce a stepwise linear multi-gap model of the cavity fields [15].

Finally, a step size of 1 cm was used for all codes, and if not otherwise specified, all multi-particle codes were run with 1k particles.

Results

Figure 2 displays a comparison of the transversal and longitudinal beam envelopes, and the losses. Evidently, the latter can only be shown for the multi-particle tracking codes.

Apart from the vertical beam envelope towards the end of the SC LINAC simulated with PyAccel, the codes exhibit good agreement with regards to the transversal envelopes. The longitudinal beam envelopes are reproduced quite well, although they appear to spread over the drift section after the SC LINAC. Nevertheless, this can be expected for an accelerator with 79 accelerating structures (19 normal conducting, 60 super conducting cavities).

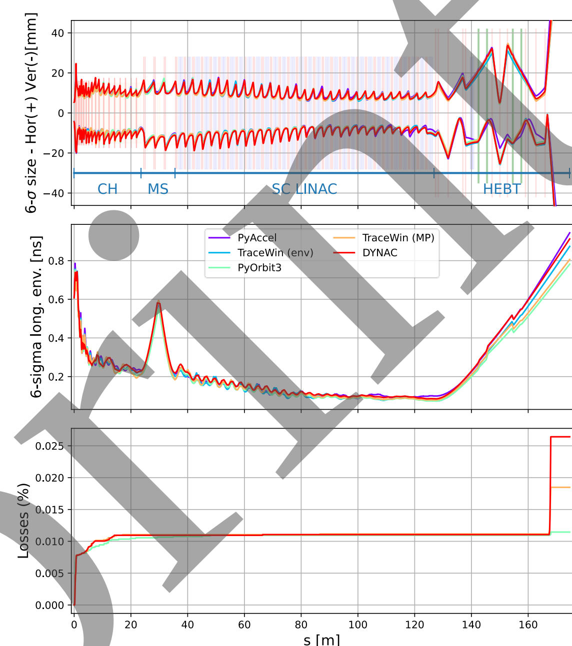

Figure 2: A comparison of the main beam parameters as simulated with the different codes. The top panel shows the transverse beam sizes (horizontal on the positive scale, vertical on the negative scale). The optics elements with their respective apertures are displayed in the background. The middle panel contains the longitudinal beam size in units of time. The bottom panel shows the losses calculated with the three multi-particle tracking codes for a distribution of 1M particles.

With regards to the losses, there is a very good agreement between the three multi-particle codes in the first part of the LINAC – especially between DYNAC and TraceWin - but in the final drift section towards the target the difference in losses becomes larger. Again, this should not come as a surprise, since foremost the losses are extremely low, i.e. 10 particles with PyOrbit3 and 150 particles with DYNAC out of 1 million and thus dominated by statistical error. Moreover, the losses are located at the intersection between the HEBT beam pipe (aperture radius of 42 mm) and the target line end section where a progressively larger aperture is attained (going from 65 mm to 200 mm over 5 m). Hence, this is a critical location where the heavily diverging beam grazes the corner of the 45 mm-to-64 mm transition point. Consequently, a small difference in beam size can result in a dramatic increase of the losses at this location.

Since the codes should be able to accurately transport the beam from the exit of the RFQ all the way through the SC LINAC and since eventually the 100 MeV beam will be injected into a new section that accelerates the beam further to 600 MeV, it makes sense to have a look at the beam parameters right after the last cavity. This is shown in Fig. 3, where the two transverse (the upper two graphs) and longitudinal (lower left graph) 1σ emittance ellipses are displayed in their respective phase-space. With regards to the FPF, the beam size at the target is critical as well (lower right panel). All codes are in good agreement apart from PyAccel, which exhibits deviating vertical and longitudinal Twiss parameters. As demonstrated by the projections of the particle distributions and the particle distributions in longitudinal phase space also halo formation, i.e. the shape of the tails of the distributions, is significantly different amongst the multi-particle codes. Especially PyOrbit3 seems to be off compared to the other two.

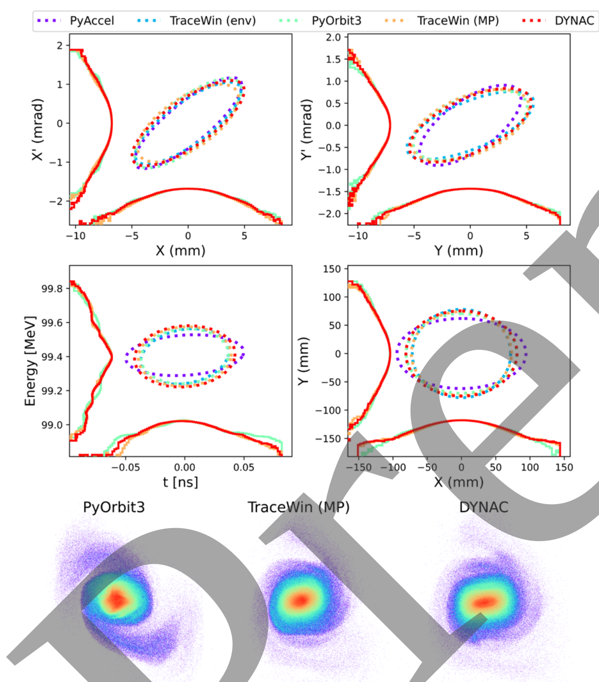

Figure 3: the phase-space ellipses and transverse beam size at the exit of the SC LINAC and FPF target. The top left and right panel display the horizontal and vertical 3σ emittances, respectively. The longitudinal emittance is displayed in the middle-left panel. The 2D transversal 3σ beam size at the FPF target is shown in the middle right panel. The beam projections for the 1M particle distributions are shown in logscale on the marginal axes. The galaxy-like shapes at the bottom corresponds to the particle distributions in longitudinal phase-space.

Not including the aberrant results from PyAccel, the transversal beam parameters agree within 10%. The longitudinal beam parameters exhibit a larger spread with a maximum difference of 20% in the emittance. Further investigations are ongoing to understand the aberrations.

Computing Time

A comparison of the computing times is listed in Table 1. All codes were run with Ubuntu24.04 within a virtual machine on a windows laptop with the following processor: Intel(R) Core(TM) Ultra 5 135U-1.60 GHz (16.0 GB RAM). The processing time is highly dependent on the space-charge model and accelerating field tracking model. However, the best agreement in beam optics was found with the chosen settings. Secondly, for the multi-particle codes the CPU time increases with the number of particles tracked. Considering the required precision that is needed to assess the basic beam parameters it was thus decided that 1k particles is the minimum. Finally, the listed values include both the tracking and loading of the field map and particle distribution files. Since for the two python-based codes this initialization process was not yet optimized and it was straightforward to decouple initialization from tracking, the time needed for the latter was also added in brackets.

In addition, a high statistics run (1M particles) was performed, as this is critical for detailed beam profile and beam loss studies. As expected, the PyOrbit3 code comes out fastest, mainly due to its simplified space-charge model. This is probably at the expense of accuracy in the transverse beam profiles and longitudinal galaxy formation.

Table 1: Comparison of the Computing Times for the Transfer-Matrix Codes and Multi-Particle Codes with a Low Statistics (1k) and a High Statistics (1M) Run

Codes	Transfer-matrix and low stat. (1k)	High stat. (1M)
PyAccel	4.9s (2.4s)	-
TraceWin (env)	**2.2s**	-
PyOrbit3	16.8s (5.3s)	**18m**
TraceWin (MP)	22.6s	50m
DYNAC	6.8s	42m

CONCLUSION

A first step has been made to compare possible candidates for an online simulation tool. Beam optics simulations on one of the MYRRHA-Phase I start-to-end lines (from the RFQ-exit to the FPF target) were performed with 5 different codes. The overall agreement for beam core properties amongst the codes, over a larger part along the LINAC is reasonably good, and close to the expected level of measurement precision. On the other hand, when looking closely at the multi-particle beam profiles, significant disparities were identified. Further investigations and improvements are ongoing.

A preliminary study of the computation times was made as well. Comparing these numbers could help in the decision making. Nevertheless, it is crucial that both accuracy and computation time are properly considered with each use case.

REFERENCES

[1] D. Uriot and N. Pichoff, "Status of TraceWin Code", in *Proc. 6th Int. Particle Accelerator Conf. (IPAC'15)*, Richmond, VA, USA, May 2015, pp. 92-94. doi:10.18429/JACoW-IPAC2015-MOPWA008

[2] DYNAC (current version V8R0), https://github.com/dynac-source

[3] E. Tanke, M. Eshraqi, R. Miyamoto, A. Ponton, R. de Prisco, E. Sargsyan, and S. Valero, "Benchmark of beam dynamics code DYNAC using the ESS proton linac", in *Proc. LINAC2014*, Geneva, Switzerland, Aug.-Sep. 2014, paper THPP043, pp. 945-947.

[4] E. Tanke, M. Eshraqi Y. Levinsen, A. Ponton, and S. Valero, "Status of and plans for the beam dynamics program DYNAC", in *Proc. LINAC'16*, East Lansing, USA, 2016, paper MOPRC007, pp. 80-82. doi:10.18429/JACoW-LINAC2016-MOPRC007

[5] W. Wittmer *et al.*, "Online modelling of the rare isotope reaccelerator REA", in *Proc. IPAC'14*, Dresden, Germany, Jun. 2014, paper THPRO0129, pp. 3195-3197.

[6] Hyunchang Jin and Ji-Ho Jang, "Studies for tuning algorithm of superconductive cavity amplitude and phase in the RAON accelerator", in *Proc. IPAC'16*, Busan, Korea, 2016, pp. 932-934. doi:10.18429/JACoW-IPAC2016-MOPOY038

[7] J. Galambos *et al.*, "ORBIT - A Ring Injection Code with Space Charge", in *Proc. PAC'99*, New York, NY, USA, Mar. 1999, paper THP82, pp. 3143-3145.

[8] A. Shishlo, S. Cousineau, J. Holmes, and T. Gorlov, "The particle accelerator simulation code pyorbit", *Procedia Comput. Sci.*, vol. 51, pp. 1272-1281, 2015. doi:10.1016/j.procs.2015.05.312

[9] C. Zlatanov *et al.*, "Evaluating PyORBIT as unified simulation tool for beam-dynamics modelling of the ESS LINAC", in *Proc. 68th Adv. Beam Dyn. Workshop High-Intensity High-Brightness Hadron Beams*, Geneva, Switzerland, 2023, doi:10.18429/JACoW-HB2023-TUC2C2

[10] MYRRHA, https://www.sckcen.be/en/infrastructure/myrrha

[11] MYRRHA Accelerator Experiment, research and development programme, https://cordis.europa.eu/project/id/662186

[12] MYRRHA Phase I, MINERVA, https://www.sckcen.be/en/infrastructure/myrrha/myrrha-phase-1-minerva

[13] U. Dorda and A. Fabich, "Implementation Status of Myrrha Phase 1 (Minerva)", *Phys.: Conf. Ser.* vol. 2687, p. 05200. doi:10.1088/1742-6596/2687/5/052007

[14] A. Shishlo and J. Holmes, "Physical Models for Particle Tracking Simulations in the RF Gap," ORNL TechNote ORNL/TM-2015/247, Jun. 2016.

[15] D. Uriot, N. Pichoff, R. Duperrier, "TraceWin manual", https://dacm-codes.fr/Softwares/TraceWin_DOC/tracewin.htm