ISBN: 978-3-95450-231-8 ISSN: 2673-5490 doi: 10.18429/JACoW-IPAC2023-MOPL063

BENCHMARK AND PERFORMANCE OF BEAM-BEAM INTERACTION MODELS FOR XSUITE

P. Kicsiny*, CERN, Geneva, Switzerland and EPFL, Lausanne, Switzerland
X. Buffat, G. Iadarola, D. Schulte, CERN, Geneva, Switzerland
T. Pieloni, M. Seidel, EPFL, Lausanne, Switzerland

Abstract

The understanding of beam-beam effects, which influence the choice of the FCC-ee design parameters for several aspects, require sophisticated and high-performance numerical simulations. The self-consistent study of the interplay of nonlinear dynamical phenomena resulting from collisions in the machine is key to accurately assess its potential performance. Although current simulation frameworks can address specific aspects of the dynamics separately, they are difficult to interface with each other for more complex studies. To address this challenge, Xsuite, a new general purpose software framework for beam dynamics simulations, is currently under development. We discuss the implementation of the beam-beam interaction in this new toolkit and the evaluation of its performance on multiple platforms.

INTRODUCTION

Beam-beam effects [1] at the Future Circular e^+e^- Collider (FCC-ee) [2] are among the principal limitations for the achievable luminosity. The nonlinear nature of the electromagnetic interaction during collisions makes it impossible to study the dynamics with only analytic methods. The common approach is therefore to use numerical simulations to track the dynamical variables of the beam particles as they pass through the various elements of an accelerator. Such simulations are made difficult by the complexity of the accelerator. Efficient numerical tools are needed to explore the vast associated parameter space. In addition, different dynamical effects can often influence each other. A realistic simulation which gives reliable information on beam instabilities and lifetimes must therefore be self-consistent. The main topic of this contribution will be to present the implementation of the beam-beam collision in the Xsuite [3] framework and its performance on CPU and GPU platforms.

Beam-Beam Modeling in Xsuite

Xsuite is a general-purpose multi-particle simulation framework used for FCC-ee related beam dynamics studies. A simulation in Xsuite consists of a bunch of ”macroparticles”, representing the charge of a larger set of real particles, and a series of accelerator elements, such as dipole and quadrupole magnets, radiofrequency cavities, or beam-beam elements when two beams collide.

The Xsuite beam-beam element considers the interaction of two beam bunches. Following the approach described in [4], the bunches are first boosted into a head-on frame, then sliced longitudinally and moved across each other one slice at a time. Macroparticles in the overlapping slices will be affected by each others’ electromagnetic field, which is computed with the Bassetti-Erskine formula [5] using the statistical properties of the opposing slice. Compared to a particle in cell (PIC) solver, using an analytical formula is a computationally cheap way of estimating the collective effect of the bunch slice. On the other hand, the formula is valid only as long as the bunch distribution is close to a Gaussian. Depending on the frequency of recomputing the statistical moments of the bunches, we can differentiate between 3 levels of approximation: weak-strong (WS), quasi-strong-strong (QSS) and strong-strong (SS) models. Figure 1 shows a sketch of the different levels of approximation present in Xsuite. It represents the collision of 2 bunches each with 2 longitudinal slices, which equals to 3 overlap configurations or time steps in total. In the WS case, the statistical moments of all slices are input manually at the beginning of the simulation and are never recomputed. In the QSS model, the moments are recomputed periodically, e.g. at the beginning of every second turn. In the SS model the moments are recomputed before each time step, which requires the largest amount of computations but enables the most realistic study of coherent beam dynamics. This beam-beam model allows a flexible choice of the approximation, using the same codebase for all.

PERFORMANCE OF THE XSUITE BEAM-BEAM MODEL

The physics of the beam-beam implementation in Xsuite has been benchmarked against other tracking codes using FCC-ee configurations [6]. It has also been benchmarked against COMBIp [7] without any parallelization. The study has shown that the measured walltimes with Xsuite and COMBIb are comparable with both codes having an approximately linear scaling with respect to the number of longitudinal slices. In this contribution, we focus on the performance scaling of the model on multiple CPUs or on a GPU.

Performance Scaling with Multiple Threads

In Xsuite it is possible to parallelize the workload using OpenMP [8], a multi-threading API, most commonly used on multi-core CPUs. With OpenMP each requested CPU core will act as a separate path of sequential execution, called a thread, and be assigned to process part of the data, i.e. a subset of the bunch particles.

The Xsuite beam-beam model consists of 3 main operations, which can be parallelized: the assigning of the slice

* peter.kicsiny[at]cern.ch

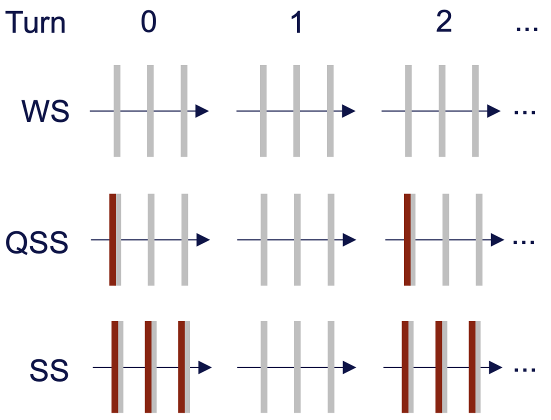

Figure 1: Sketch of an example multi-turn simulation with beam-beam collisions of two bunches with both having 2 slices and an update frequency of 2 turns in the different approximations: weak-strong (WS), quasi strong-strong (QSS) and strong-strong (SS). The horizontal axes show time. The gray lines represent the 3 time steps where overlapping slices interact via the Bassetti-Erskine formula. The red lines indicate the (re)computation of the slice statistical moments.

index to the macroparticles (1), the computation of the statistical moments of the slices (2), and the application of the 6D beam-beam force on the macroparticles, referred to as the synchrobeam kick (3). If the number of slices in the bunch is N_s, in case of the SS model all 3 operations will be executed $2N_s - 1$ times for each bunch. In the QSS case, operations (1) and (2) will be called only once for each bunch before the interaction of the first slice pair, while operation (3) will still be called $2N_s - 1$ times. In the WS model, where only one bunch is tracked, there is no calculation of slice indices and statistical moments as in this case the equilibrium beam parameters are input by the user. Operation (3) in the WS case will be executed only once. Table 1 summarizes the number of executions for these 3 operations in the different approximations.

Table 1: Number of Calls per Bunch to the Main Operations in the Simulation of a Single Collision with the Different Beam-Beam Models, Using N_s Longitudinal Slices

Beam-beam with N_s slices	WS	QSS	SS
Assign slices	0	1	$2N_s - 1$
Compute stat. moments	0	1	$2N_s - 1$
Synchrobeam kick	1	$2N_s - 1$	$2N_s - 1$

Strong scaling A common way to estimate the parallel performance of an algorithm is to perform a strong scaling study, in which the number of parallel processes is increased while the workload is kept constant. Using n parallel processes, one can measure the speedup $\frac{T_1}{T_n}$ defined as the ratio of the walltime of the sequential and parallel execution. On the other hand, Amdahl's law [9] can be used to make a theoretical estimate $S(p,n)$ on the speedup as a function of the number of parallel processes and the fraction p of the sequential runtime that is parallelized:

$$S(p,n) = \frac{1}{(1-p) + \frac{p}{n}}. \tag{1}$$

To test the strong scaling of Xsuite's beam-beam model, we have chosen the FCC-ee Z configuration [2] with 10^6 macroparticles and 100 slices in both colliding bunches. In the WS case, the synchrobeam kick is performed in only one call to the kernel and the corresponding workload will be the full bunch. This part of the code can be parallelized and it makes up approximately 99.6% of the total walltime when simulating a single collision with the chosen setup. Conversely, in the other models each call to the kernel will have a different workload corresponding to the number of overlapping slices in the given step. In the sequential case these operations make up approximately 99.2% of the total time, with a negligible $\sim O(10^{-4})$ difference between the 2 models. Using Eq. (1), we have estimated the speedup by taking $p = 0.992$ for the QSS and SS models, and $p = 1$ for the WS case. In our study, we have scanned the number of OpenMP threads and in each case simulated a single beam-beam collision with each beam-beam model. The measured speedup values, compared to the execution time with only one thread, are shown in Fig. 2.

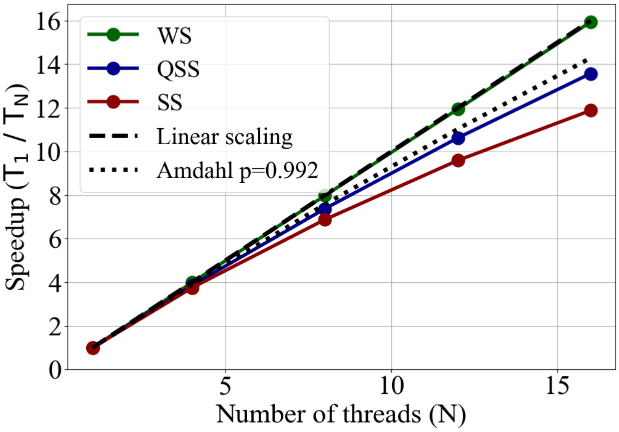

Figure 2: Strong scaling speedup of the 3 beam-beam models up to 16 OpenMP threads, using the FCC-ee Z parameters, 100 slices and 10^6 macroparticles per bunch.

The figure demonstrates that there is an overall good scaling with all models up to 16 threads. The WS model yields a perfect scaling with the number of threads, while with 16 threads a factor 14 speedup can be achieved in the QSS model and a factor 12 using the full SS model. The reduction in performance with the QSS model can be explained by the increased number of function calls to the synchrobeam kick operation. In the SS model all operations indicated in Tab. 1 are performed $2N_s - 1$ times for both bunches, where $N_s = 100$ is the number of slices in this particular study. This means an increased number of function calls and memory accesses which results in a somewhat bigger overhead.

Weak scaling In a weak scaling study one is interested in the change of execution time with respect to the workload per parallel processing unit which is kept constant. This means one increases the problem size and the number of processing units at the same time. In the ideal case, the total measured walltime should remain constant. We have performed a simple scaling test with the same setup as in the strong scaling study, except that we have reduced the number of macroparticles to 10^5 per bunch. The measured efficiencies $\frac{T_1}{T_n}$ of one beam-beam collision are shown in Fig. 3.

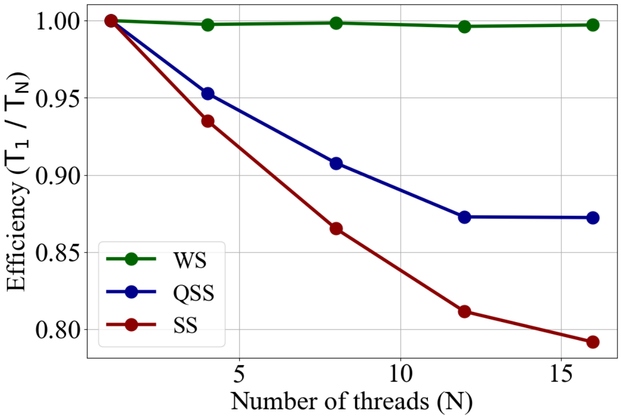

Figure 3: Weak scaling efficiency of the 3 beam-beam models up to 16 OpenMP threads, using the FCC-ee Z parameters, 100 slices and 10^5 macroparticles per bunch.

It can be seen from the figure that the efficiency decreases with the increase of the model complexity. While the WS model, being a single loop over the macroparticles, has a perfect efficiency up to 16 threads, the efficiency of the QSS and SS models decreases to 85% and 80% respectively. This is caused mainly by the large number of function calls to the parallel operations and memory accesses.

Performance on a GPU

We have benchmarked the beam-beam collision model on different GPU platforms: NVIDIA A100, TITAN V and Tesla T4. We have measured the walltime of our benchmark job on these platforms and compared them to that measured on a single CPU core, which shares the compute node with the corresponding GPU. For this study, a setup with 100 slices, 10^6 macroparticles and the nominal FCC-ee Z parameters [2] was chosen. The simulation consisted of a single interaction with the beamstrahlung and Bhabha scattering features disabled. Figure 4 shows the speedup values (GPU walltime over single CPU core walltime) obtained by executing the job on 3 different types of GPUs.

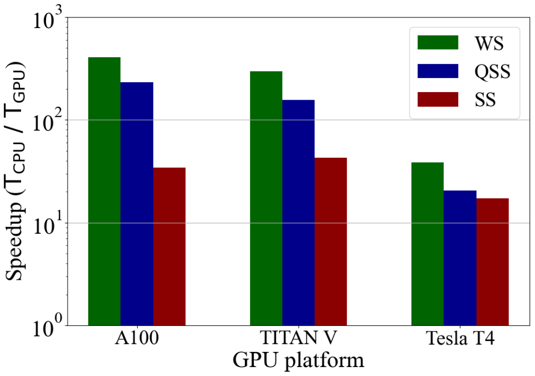

Figure 4: GPU / CPU speedup of simulating a single beam-beam collision on different GPU platforms, compared to a sequential execution on a CPU, for all `Xsuite` beam-beam models, using the FCC-ee Z parameters, 100 slices and 10^6 macroparticles per bunch.

The results indicate that the most significant speedup with the GPU is achieved when using the WS model. This is because this model, consisting only of the synchrobeam kick operation, i.e. a single loop, can execute fully synchronously and requires no communication, which enables the algorithm to benefit the most from GPU parallelization. The QSS model yields a 1 to 2 orders of magnitude speedup, depending on the type of the GPU, while with the SS model we have observed a speedup of up to a factor 40. In the SS case, the performance is affected more by the increased number of calls to the computation of the moments and their communication between the bunches.

SUMMARY

In this contribution the performance of the beam-beam collision model of `Xsuite` has been investigated. We have conducted a strong and weak scaling study with OpenMP, up to 16 parallel threads, and found a good speedup and efficiency for all levels of approximation of the beam-beam element. In addition, we have performed a performance benchmark on 3 different types of GPU platforms. Based on our observations we conclude that all models scale well on GPU, therefore it is a preferred choice over multithreading, given the availability of the resource. The increased number of function calls and communication of data somewhat reduces the scaling of the SS model on GPU. Nevertheless, a good fraction of the relevant dynamical effects in circular colliders are modeled with a sufficient accuracy using the QSS model given the low disruption parameter.

As tracking through the magnetic lattice can be well parallelized on the GPU [10], the good scaling of the WS and QSS models will make it a suitable choice of platform for efficient studies of dynamical effects involving the interplay of the beam-beam forces and the lattice using `Xsuite`.

REFERENCES

[1] W. Herr and T. Pieloni, “Beam-beam effects”, in *Contribution to the CAS - CERN accelerator school: advanced accelerator physics course*, Trondheim, Norway, Aug. 2013. doi:10.5170/CERN-2014-009.431

[2] A. Abada *et al.*,“FCC-ee: The Lepton Collider”, *Eur. Phys. J. Spec. Top.*, vol. 228, pp. 261–623, 2019. doi:10.1140/epjst/e2019-900045-4

[3] xsuite, https://github.com/xsuite

[4] K. Hirata, “Don’t be afraid of beam-beam interactions with a large crossing angle”, *Phys. Rev. Lett.*, vol. 74, p. 2228 1995. doi:10.1103/PhysRevLett.74.2228

[5] M. Bassetti and G. Erskine, “Closed expression for the electrical field of a two-dimensional Gaussian charge”, CERN, Geneva, Switzerland, Rep. CERN-ISR-TH-80-06, ISR-TH-80-06, 1980.

[6] 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, Sep. 2022, pp. 165–170. doi:10.18429/JACoW-eeFACT2022-WEZAT0102

[7] COMBIp, https://combi.web.cern.ch/parallelization

[8] OpenMP, https://www.openmp.org

[9] G. Amdahl, “Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities”, in *Proc. The April 18-20, 1967, Spring Joint Computer Conference*, Atlantic City, NJ, USA, Apr. 1967, pp. 483–485. doi:10.1145/1465482.1465560

[10] M. Schwinzerl, H. Bartosik, R. De 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