GPU OPTIMIZED CODE FOR LONG TERM SIMULATIONS OF BEAM-BEAM EFFECTS IN COLLIDERS*

Y. Roblin, V. Morozov, B. Terzić, Jefferson Lab, Newport News, VA, USA
M. Aturban, D. Ranjan, M. Zubair, Old Dominion University, Norfolk, VA, USA

Abstract

We report on the development of a new code for long-term simulation of beam-beam effects in particle colliders. The underlying physical model relies on a matrix-based arbitrary-order symplectic particle tracking for beam transport and the Bassetti-Erskine approximation for the beam-beam interaction. The computations are accelerated through a parallel implementation on a hybrid GPU/CPU platform. With the new code, previously computationally prohibitive long-term simulations become tractable. We are planning to use the new code to model the proposed medium-energy electron-ion collider (MEIC) at Jefferson Lab.

INTRODUCTION

Beam-beam effects are one of the primary effects limiting the luminosity in proton or ion colliders such as MEIC[1]. Until recently, the study of long-term stability of rings with beam-beam interactions was prohibitive due to the heavy computational load. Previous attempts at investigating this for MEIC were restricted to linear transport and short term behavior [2, 3].

The interaction between two colliding beams is described by the Poisson equation which can be solved by a number of methods at a high computational cost. Various approximations have been proposed in the past to lessen the computational load.

BEAMBEAM3D [4] uses a shifted integrated 2D Green's function method to solve the equation on a grid. The 2D approximation is made possible by dividing the beams in thin slices.

A further approximation can be to assume a gaussian beam distribution which leads to a one-dimensional integration [5]. Finally Bassetti-Erskine (BE)[6] introduce one more level of approximation where they assume that the beams have vanishing length and Gaussian transverse distributions. They obtained a closed form suitable for efficient implementation. We generalized their formalism for flat beams to the general geometry which may also include upright ($\sigma_y > \sigma_x$) and round ($\sigma_y = \sigma_x$) beams.

The assumption that the beam are gaussian is not limiting in any way. We are interested in the steady-state solution for which we have a stable long-term behavior. Departure from Gaussian would hint at excessive collective or resonant effects which can be avoided by a better choice of the working point and design parameters.

The new approach presented in this paper enables us to carry out weak-strong and strong-strong beam beam simulations with a nonlinear transport of arbitrary high order. We implemented a GPU-based parallel computation for both tracking and collisions.

ALGORITHM DESCRIPTION

Each beam is represented by an appropriate gaussian distribution of particles and the effect of the collision is computed using the generalized BE approximation. For this purpose, we consider a region given as the length of the longest bunch. Each beam is divided into slices that are small enough for the approximation to hold.

It is important to note that the calculation scales linearly with the number of particles since the forces on each particles are computed by the BE integrals of the colliding slices.

Particle Tracking

In between collisions, the particles are transported symplectically with a one turn map. We modeled the interaction region of the MEIC ring in Cosy Infinity[7] and generated symplectic maps of arbitrarily high orders.

Given the expected equilibrium bunch length, energy spread and longitudinal tune, a linear synchro-betatron map is constructed to provide the longitudinal transport. This map can be extended to higher orders to accurately represent high harmonic cavities utilized for the bunching process.

Collision

In order to handle the collision, we take the one turn map expressed at the interaction point and transport each individual slice from their respective positions in the bunches to the interaction point by drifting a distance $\frac{N-1}{2}\Delta$ where Δ is the size of a individual slice and N the initial slice position. This is accomplished by modifying the one-turn map as such:

$$\mathcal{M}_{21}^{(\pm)} = \mathcal{D}\left(\pm\frac{N-1}{2}\Delta\right)\mathcal{M}^{(\pm)}\mathcal{D}\left(\pm\frac{N-1}{2}\Delta\right). \quad (1)$$

The slices are collided with all the other slices in the other beam by sliding them across. This treatment allows for the accurate depiction of the hourglass effect.

[*] Authored by Jefferson Science Associates, LLC under Contract No DE-AC05-06OR23177. The U.S Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce this manuscript for United States Government Purposes

RESULTS

We simulated the beam-beam collisions between the electrons and protons in the MEIC machine. For these simulations, the rings were set at their design parameters for collision. Table 1 summarizes the design parameters for MEIC.

Table 1: Design Parameters for the MEIC

Quantity	**Unit**	e^- **beam**	p **beam**
Energy	GeV	5	60
Collision frequency	MHz	750	
Particles per bunch	10^{10}	2.5	0.416
Beam current	A	3.0	0.5
Energy spread	10^{-3}	0.71	0.3
rms bunch length	mm	7.5	10
Horiz. bunch size at IP	μm	23.4	
Vertical bunch size at IP	μm	4.7	
Horiz.l emit. (norm.)	μm	53.5	0.35
Vertical emit. (norm.)	μm	10.7	0.07
Horizontal β^*	cm	10	
Vertical β^*	cm	2	
Vertical beam-beam tune shift		0.029	0.0145
Damping time	turns	1516 (6.8 ms)	$\approx 2.4 \times 10^6$ ($\approx$ 11000 s)
Synchrotron tune		0.045	0.045
Ring length	m	1340.92	1340.41
Peak luminosity	cm^{-2}s^{-1}	0.562×10^{34}	
Reduction (hourglass)		0.957	
Peak luminosity with hourglass effect	cm^{-2}s^{-1}	0.538×10^{34}	

The number of longitudinal slices was varied and the luminosity computed from the particle distributions for each slice and then summed up:

$$\mathcal{L} \equiv \frac{N_+N_-f_0}{2\pi\Sigma_x\Sigma_y}\exp\left[-\frac{(\bar{x}_+-\bar{x}_-)^2}{2\Sigma_x^2}-\frac{(\bar{y}_+-\bar{y}_-)^2}{2\Sigma_y^2}\right], \qquad (2)$$

where $\Sigma_q = \sqrt{(\sigma_q^+)^2+(\sigma_q^-)^2}$, and $\bar{q}^\pm = \frac{1}{N_\pm}\sum_{i=1}^{N_\pm} q_i^\pm$, and f_0 is the bunch collision frequency.

From Fig. 1, one can see that in this particular calculation two slices are sufficient to reach convergence after a few collisions. The drop in luminosity between the calculation with one slice and the calculation with two slices is due to the hourglass effect. It is more obvious if one injects a longer bunch. Figure 2 shows that for a 10 centimeter bunch, one would need three slices. The expected luminosity drop is within a few percent of what the hourglass effect reduction is predicted to be.

We check the adequacy of the BE approximation by monitoring the higher order moment of the beam distributions in the slices to detect deviations from gaussian.

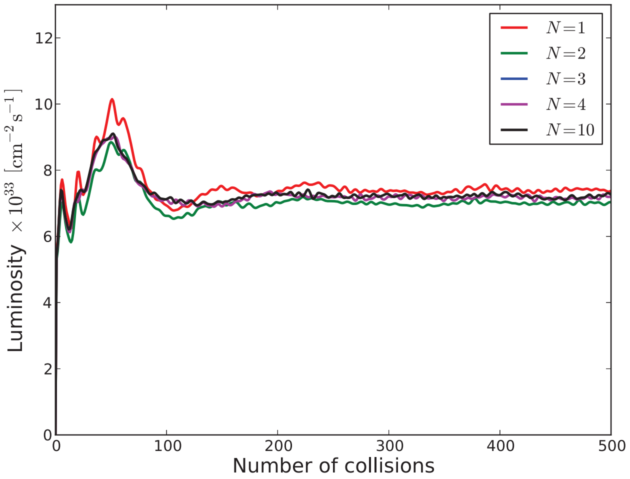

Figure 1: Comparison of computed luminosities as a function of the number of slices with the new algorithm for a 1 cm bunch.

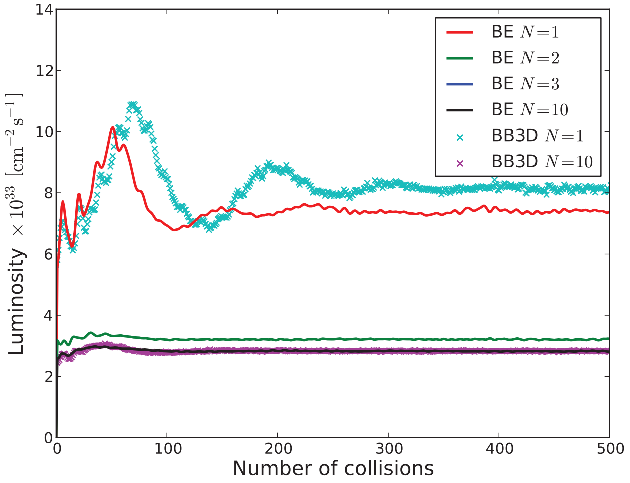

Figure 2: Comparison of computed luminosities as a function of the number of slices with the new algorithm for a 10 cm bunch. Results from BeamBeam3D are also shown.

PARALLELIZATION

We developed a parallel version of these algorithms suitable for GPU implementation. The approach we use is to assign a thread to a single particle. Each thread applies the map and updates the six attributes of the particle in each iteration.

The tracking computation for each particle can be done concurrently while the computation of the beam-beam interaction requires communication between threads.

In our algorithm, the interaction of a slice of the first beam with a slice of the second beam is realized by activating threads associated with the particles in the two slices. Also, interaction of several slices is done in parallel as the beams go through each other. We tested our algorithm using a million particles and with varying number of iterations. We implemented our algorithms on NVIDIA Tesla M2090 consisting of 512 cores.

ISBN 978-3-95450-122-9

The tracking algorithm results in a maximum speedup (CPU time/GPU time) of about 170 obtained after a few thousand turns where the overhead of the initial I/O becomes negligeable. Performing the beam-beam collisions in addition of the tracking, the speed gain is about 70. We are expecting significant improvement over these preliminary results. Among the issues we will look into further optimization are better use of the shared memory, a newer GPU architecture and the higher level of parallelization involving GPU units themselves.

COMPARISON WITH OTHER CODES

BEAMBEAM3D was used to perform the same calculation. For this comparison, we turned synchrotron damping off and ran calculations for a varying number of slices.

Figure 3 shows that the serial version of our code scales linearly with the number of slices and is comparable or faster than the 64 CPU version of BEAMBEAM3D.

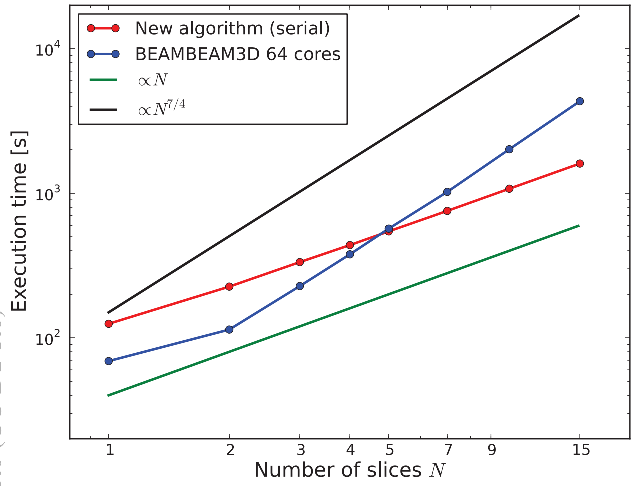

Figure 3: Comparison of execution times for the beam-beam simulation with 10000 particles and 5000 turns with serial version of the new algorithm and with the parallel version of BEAMBEAM3D code on 64 cores.

The estimated luminosities are both qualitatively and quantitively similar, in fact nearly identical to each other as can be seen in Fig. 2 for ten slices.

FUTURE WORK

A number of additional features are being developed and will be included in the next iteration. Amongst these are synchrotron damping, cooling of the proton beam by a low energy electron beam, intra beam scattering and simulation of the crabbing as well as the acceleration scheme from injection (30 GeV) to collision energy (60 GeV) and higher order harmonic cavities utilized for the bunching of the proton beam. We are also exploring a newer GPU architecture and shared memory management scheme which we believe will lead to further significant speedups.

REFERENCES

[1] *Science requirements and conceptual design for a polarized medium energy electron-ion collider at Jefferson Lab* (2012), arXiv:1209.0757

[2] Y. Zhang and J. Qiang, Proc. of PAC2009, Vancouver, Canada (2009)

[3] B. Terzić and Y. Zhang, Proc. of IPAC2010, Kyoto, Japan (2010)

[4] J. Qiang, R. Ryne and M. Furman, PRSTAB 5, 104402 (2002)

[5] R. Wanzenberg, Tech. Rep. DESY M 10-01, DESY (2010)

[6] M. Bassetti and G. Erskine, Tech. Rep. CERN-IRS-TH/80-06, CERN (1980)

[7] K. Makino and M. Berz, *COSY INFINITY version 8*, Nuclear Instruments and Methods, A427:338343, 1999