GPU-ACCELERATED LONG-TERM SIMULATIONS OF BEAM-BEAM EFFECTS IN COLLIDERS*

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

Abstract

We present an update on the development of the new code for long-term simulation of beam-beam effects in particle colliders. The underlying physical model relies on a matrix-based arbitrary-order particle tracking (including a symplectic option) for beam transport and the generalized Bassetti-Erskine approximation for 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. The new code will be used to model the proposed Medium-energy Electron-Ion Collider (MEIC) at Jefferson Lab.

INTRODUCTION

Beam-beam interaction is one of the most important dynamical factors limiting the collider's luminosity and therefore limiting its scientific efficiency. 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 the beam-beam interaction for the proposed Medium-energy Electron-Ion Collider (MEIC) [1] at Jefferson Lab were restricted to linear transport and short term behavior [2, 3].

Effects due to collision between the two beams in a collider is described by the Poisson equation which can be solved by a number of methods at a high computational cost. This computational load can be alleviated by invoking various approximations.

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 into thin slices. Another approximation is to assume a gaussian beam distribution which leads to a one-dimensional integration [5]. Going a step further, Bassetti-Erskine (BE) [6] solution introduces one more level of approximation in which the beams are treated as if (1) they have vanishing length and (2) gaussian transverse distributions. When these approximations hold, the solution to the Poisson equation is exact and amenable to efficient numerical implementation. Because of this efficiency, the BE model at the heart of a beam-beam code gives us the best chance of accurately studying the long-term dynamics in colliders. Their solution for flat beams is generalized here to the general geometry which may also include upright ($\sigma_y > \sigma_x$) and round ($\sigma_y = \sigma_x$) beams.

We relax the first approximation of an infinitesimally short bunch by dividing both finite-length colliding bunches into several slices. Each of the slices is then treated as an infinitesimally short bunch. To that end, the new code employs the synchro-beam mapping, the only known symplectic beam-beam map usable for the long bunch [7].

The second approximation—that the beams are transversally gaussian—is not limiting in any way. We are interested in the steady-state, 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. We check the adequacy of the BE approximation by monitoring the higher order moments of the beam distributions in the slices to detect deviations from gaussian.

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 arbitraryly high order. The code is significantly accelerated by carrying out massively parallel computations involving particle tracking and collisions on a GPU platform.

ALGORITHM DESCRIPTION

Each colliding beam is simulated by a set of particles sampling their initial gaussian distributions. Beam-beam effects of one beam on the other is modeled using the generalized BE approximation. Each beam is divided into slices that are small enough for the BE approximation of infinitesimally short length to hold. In between the consecutive collisions, the beams are transported through the rings using the Taylor maps, with symplectic tracking as a computationally more expensive option.

Particle Tracking

The particle transport through the ring is carried out using an arbitrary-order Taylor map generated by COSY Infinity [8]. We choose a map-based tracking because it is much faster than the alternative approach of integrating the particle motion through the ring lattice and thus is the only way to provide the necessary large number of turns within a reasonable computation time. A map-based tracking is efficiently parallelizable. Moreover, the expansion coefficients may offer more insight into the optical system.

Symplectic tracking option is implemented using the generating function F_2 [9] for which $(\boldsymbol{q}_f, \boldsymbol{p}_i) =$

* 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

$\mathbf{J}\nabla F_2(\boldsymbol{q}_i, \boldsymbol{p}_f)$, and $\mathbf{J} = \begin{bmatrix} 0 & -\mathbf{I} \\ \mathbf{I} & 0 \end{bmatrix}$, where $(\boldsymbol{q}_i, \boldsymbol{p}_i)$ and $(\boldsymbol{q}_f, \boldsymbol{p}_f)$ are the initial and final phase-space coordinates, respectively. Applying a truncated Taylor map $\mathbf{M}$ associated with the generating function F_2 calculates $(\boldsymbol{q}'_f, \boldsymbol{p}'_f) = \mathbf{M}(\boldsymbol{q}_i, \boldsymbol{p}_i)$, which, along with $(\boldsymbol{q}_i, \boldsymbol{p}_i)$, is used as the starting point in computing symplectic final solution $(\boldsymbol{q}_f, \boldsymbol{p}_f)$. Because $(\boldsymbol{q}'_f, \boldsymbol{p}'_f)$ is very close to $(\boldsymbol{q}_f, \boldsymbol{p}_f)$, the equation above can be solved to machine accuracy within a few iterations.

Collision

The generalized BE formalism applies only to an infinitely short bunch. In order to simulate realistic beams of finite length, we use the prescription similar to what has been done in BEAMBEAM3D: divide each bunch in several slices, each of which can be treated as an infinitesimally short bunch. At every collision between the two beams, each slice in one beam collides with each slice in the other beam according to the generalized BE formalism.

When each bunch is divided into M slices, there is a total of M^2 collisions between the slices. Each particle experiences M kicks, one from each slice in the other beam. This means that the computational load associated with the collision of the two beams scales linearly with the number of slices. It also scales linearly with the number of particles.

We bin the bunch particles into slices based on their longitudinal positions. Each slice then has a sequence of locations along the beam orbit around the IP, at which it collides with the similarly-defined different slices of the opposing beam. Since the bunches' transverse phase space parameters are changing rapidly around the IP, each slice of each colliding pair has to be propagated properly to each collision point. For each of the slices in a collision, we apply an appropriate drift transformation taking the slice from the IP to the collision point. We then calculate and apply the beam-beam kick to each of the particles in the two colliding slices based on the slices' transverse parameters at the collision point. We then propagate the slices back to the IP by an inverse drift but with the kick information already contained in the particle coordinates. This process is repeated sequentially for each slice in every collision in the order, in which the slices collide. This treatment allows for an accurate depiction of the hourglass effect.

Hourglass Effect

When the bunch's length is on the order of the beta function (β^*) at the IP, the luminosity experiences a geometrical reduction known as the *hourglass effect*. The reduction factor in luminosity due to the hourglass effect is analytically estimated in [10].

We use our new code to compute the hourglass effect by comparing the luminosity computed with many slices—where hourglass effect can be successfully modeled—to the luminosity computed in a simulation with a single slice, where the entire distribution is collapsed to a single infinitesimal slice, and as such cannot account for the effect.

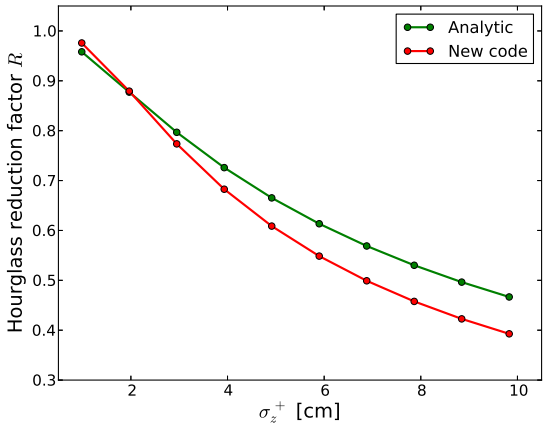

Figure 1: The hourglass effect as a function of the initial bunch length, computed in a simulation with $N = 128000$ particles and $M = 10$ slices in each bunch using our new code (red curve) and using the analytic expression from [10] (green curve). The lattice used is that of the MEIC, with the constant ratio σ_z^+/σ_z^-; the current MEIC baseline design has $\sigma_z^+ = 1$ cm and $\sigma_z^- = 0.75$ cm [1].

Each analytic point (shown in green in Fig. 1) is computed after extracting average values of transverse beam sizes and effective beta functions at the IP after the equilibrium is reached and plugging them into the integral equation for the reduction factor found in [10]. The ratio between the two luminosities yields the computed value for geometric reduction factor. The results are shown in Fig. 1. The agreements between the computed hourglass reduction factor and the analytically predicted value is excellent.

Convergence

When the finite longitudinal size of a particle bunch is simulated with M slices, it is expected that the relevant dynamical quantities converge as the number of slices grows. It is also important to see at which M this convergence occurs, so as not to unnecessarily add to the computational overhead (recall that the computational load scales linearly with the number of slices M).

Figure 2 shows the luminosity of the collision between the electron and ion beams in the proposed MEIC as a function of the number of slices M, executed with our new code. The panel (a) shows a simulation of the nominal MEIC parameters (with $\beta_x^* = 10$ cm and $\beta_y^* = 2$ cm at the IP) for which the geometric reduction in luminosity due to the hourglass effect is minimal, only about 2.5%, with the longitudinal rms size $\sigma_z^- = 0.75$ cm and $\sigma_z^+ = 1$ cm. The panel (b) shows a simulation of the same set of parameters, only with the longitudinal rms size of each beam six times the nominal value, for which the reduction due to the hourglass effect is significant, about 55%. The hourglass effect becomes important when the rms size of the bunches becomes of the order of the focusing at the IP (β^*).

The differences between the $M = 1$ and $M > 1$ results

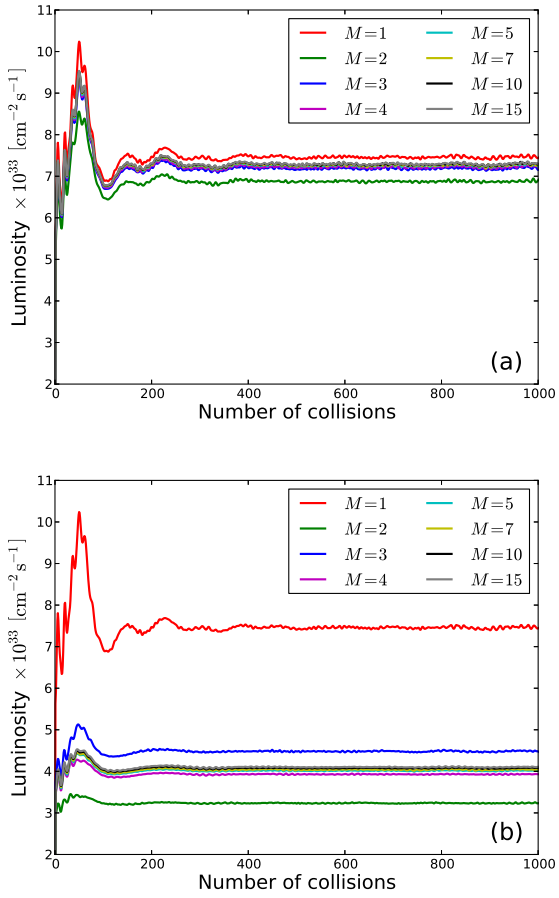

Figure 2: Computed luminosities for different number of slices M with $N = 40000$ particles. Panel (a): The nominal MEIC parameters where the reduction factor due to hourglass is only 2.5% (the first point from the left on the red curve in Fig. 1). Panel (b): The same MEIC nominal parameters, except the beam bunches are six times longer, with the hourglass reduction factor of 55% (the sixth point from the left on the red curve in Fig. 1).

are due to the hourglass effect. Simulations with $M = 2$ consistently underestimate the values to which the luminosity converges as the number of slices is increased. For cases where the hourglass effect is either minimal, as in Fig. 2(a), or moderate, simulations with $M \geq 3$ yield virtually the same results, indicating rapid algorithm convergence. This is an indication that the relevant physics of the geometric reduction in luminosity due to the hourglass effect is accurately captured with as few as three slices. However, as the rms longitudinal size of the bunches is further increased, and the hourglass effect becomes severe, it is expected that more slices are needed to accurately model the hourglass reduction. This is observed in Fig. 2(b), where $M = 3$ slices is no longer in perfect agreement with $M > 3$ cases, as was the case with the bunches which are one sixth the size (Figs. 2(a)).

PARALLELIZATION

The sheer amount of computation involved in tracking and colliding beams over $10^7 - 10^9$ collisions is daunting. In serial, the problem would simply be computationally intractable, which is why the use of sophisticated, finely-tuned algorithms running on massively-parallel platforms is required. The new code is ideally suited for the Single Instruction Multiple Data (SIMD) concept that makes GPU computation so powerful—both particle tracking and beam collision are processes which execute the same set of computations without the need for communication.

We implemented the new beam-beam algorithm on a hybrid CPU/GPU platform, taking the full advantage of the highly repetitive nature of the calculations. More precisely, one portion of the code—the setup, initialization and I/O—runs on the traditional CPU platform, while computationally intensive parts—particle tracking and beam collisions—execute on a single or several GPU devices. This speeds up the most time-consuming calculations by a few orders of magnitude, leading to substantial overall speedup. We used NVIDIA Tesla M2090 consisting of 512 cores. Each CPU hosts 4 GPU devices.

The tracking algorithm results in a maximum speedup (CPU time/GPU time) on a single GPU device of over 280 obtained after a few thousand turns where the overhead of the initial I/O becomes negligible. The speedup scales nearly linearly with multiple GPU devices.

We are currently benchmarking the GPU code in the full collision mode. Our preliminary results are encouraging and will be soon reported in a publication.

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 and intrabeam scattering. Finally, we are going to use the new code for long-term simulations of the MEIC.

REFERENCES

[1] S. Abeyratne *et al.*, arXiv:1209.0757 (2012)

[2] Y. Zhang and J. Qiang, PAC 2009, p. 2653

[3] B. Terzić and Y. Zhang, IPAC 2010, TUPEC083

[4] J. Qiang, R. Ryne and M. Furman, Phys. Rev. ST Accel. Beams 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] A. Chao and M. Tigner, eds., *Handbook of Accelerator Physics and Engineering* (World Scientific, 2009)

[8] K. Makino and M. Berz, *Nucl. Instr. Meth. A* 427, 338 (1999)

[9] M. Berz, in *Nonlinear Problems in Future Particle Accelerators* (World Scientific, 1991), p. 288

[10] M. Furman, PAC 1991, p. 422