LONG-TERM SIMULATIONS OF BEAM-BEAM DYNAMICS ON GPUS*

B. Terzić†, K. Arumugam, R. Majeti, C. Cotnoir, M. Stefani, D. Ranjan, M. Zubair, A. Godunov
Old Dominion University, Norfolk, VA 23529, USA
V. Morozov, H. Zhang, F. Lin, Y. Roblin, E. Nissen, T. Satogata
Jefferson Lab, Newport News, VA 23606, USA

Abstract

Future machines such as the electron-ion colliders (JLEIC), linac-ring machines (eRHIC) or LHeC are particularly sensitive to beam-beam effects. This is the limiting factor for long-term stability and high luminosity reach. The complexity of the non-linear dynamics makes it challenging to perform such simulations which require millions of turns. Until recently, most of the methods used linear approximations and/or tracking for a limited number of turns.

We have developed a framework which exploits a massively parallel Graphical Processing Units (GPU) architecture to allow for tracking millions of turns in a symplectic way up to an arbitrary order and colliding them at each turn. The code is called GHOST for GPU-accelerated High-Order Symplectic Tracking. As of now, there is no other code in existence that can accurately model the single-particle non-linear dynamics and the beam-beam effect at the same time for a large enough number of turns required to verify the long-term stability of a collider. Our approach relies on a matrix-based, arbitrary-order, symplectic particle tracking for beam transport and the Bassetti-Erskine approximation for the beam-beam interaction.

REQUIREMENTS

To put the problem of long-term beam-beam simulations in perspective, for the current JLEIC [1] layout in one hour of collider operation each bunch makes about 400 million turns. The requirements for the long-term beam-beam simulations are: (i) high-order symplectic tracking; (ii) speed; (iii) beam-beam collisions. An additional requirement for the JLEIC design is the ability to accommodate the "gear change", an uneven number of bunches in each colliding beam.

GHOST speeds up computations by employing approximations and using novel computational architectures.

GHOST: CODE OUTLINE

We present GHOST (Gpu-accelerated High-Order Symplectic Tracking) code, which resolves the computational challenges of beam-beam dynamics simulation by: (i) using one-turn maps for particle tracking; (ii) employing Bassetti-Erskine approximation for collision; (iii) implementing the code on a massively-parallel GPU platform.

Computation on GPUs are: (i) ideal for "same instruction multiple data" (particle tracking); (ii) best when no communication is required (tracking; collision); (iii) Moore's law still applies to GPUs (no longer for CPUs).

GHOST: Tracking

The particle transport through the ring is carried out using an arbitrary-order Taylor one-turn map $\mathbf{M}$ generated by COSY Infinity [2]:

$$x = \sum_{\alpha\beta\gamma\eta\lambda\mu} \mathbf{M}(x|\alpha\beta\gamma\eta\lambda\mu) x^\alpha a^\beta y^\gamma b^\eta l^\lambda \delta^\mu, \qquad (1)$$

for each of the six phase-space coordinates: x, $a \equiv p_x/p_0$, y, $b \equiv p_y/p_0$, l, and δ where x and y are the transverse particle positions, a and b are the transverse momentum components p_x and p_y, respectively, normalized to the reference momentum p_0, $l = -(t-t_0)v_0\gamma_0/(1+\gamma_0)$ and $\delta = (K-K_0)/K_0$. Here t, K, v_0, and γ_0 are the time of flight, kinetic energy, velocity, and Lorentz factor, respectively. The subscript 0 indicates the reference value of the variable. The six variables form three canonically conjugate pairs.

Symplectic tracking option is implemented using the generating function F_2 [3]:

$$(\boldsymbol{q}_f, \boldsymbol{p}_i) = \mathbf{J}\nabla F_2(\boldsymbol{q}_i, \boldsymbol{p}_f), \qquad (2)$$

with

$$\mathbf{J} = \begin{bmatrix} 0 & -\mathbf{I} \\ \mathbf{I} & 0 \end{bmatrix}. \qquad (3)$$

Given the generating function F_2 and the corresponding truncated map $\mathbf{M}$, we first calculate $(\boldsymbol{q}'_f, \boldsymbol{p}'_f)$ by applying $\mathbf{M}$ to $(\boldsymbol{q}_i, \boldsymbol{p}_i)$, and then use $(\boldsymbol{q}_i, \boldsymbol{p}_i, \boldsymbol{q}'_f, \boldsymbol{p}'_f)$ as a starting point for solving Eq. (2) numerically. Because $(\boldsymbol{q}'_f, \boldsymbol{p}'_f)$ is very close to $(\boldsymbol{q}_f, \boldsymbol{p}_f)$, Eq. (2) can be solved to machine accuracy in a few iterations.

We compare the results of particle tracking from GHOST, which uses a single-turn map, with that from elegant [4], which uses element-by-element tracking. Dynamic apertures computed with the two conceptually different codes are in excellent agreement (Figure 1).

GHOST: Collisions

GHOST uses a Bassetti-Erskine (BE) [5] approximation which greatly reduces the computational load associated with beam-beam interaction when the interacting beams are: (i) well-approximated by a Gaussian transverse distribution,

* We are thankful for the generous support of the Old Dominion University Research Foundation through the Research Seed Funding Program 15-492. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research. We acknowledge the support of Jefferson Lab grant to Old Dominion University 16-347. Authored by Jefferson Science Associates, LLC under U.S. DOE Contract No. DE-AC05-06OR23177 and DE-AC02-06CH11357.

† bterzic@odu.edu

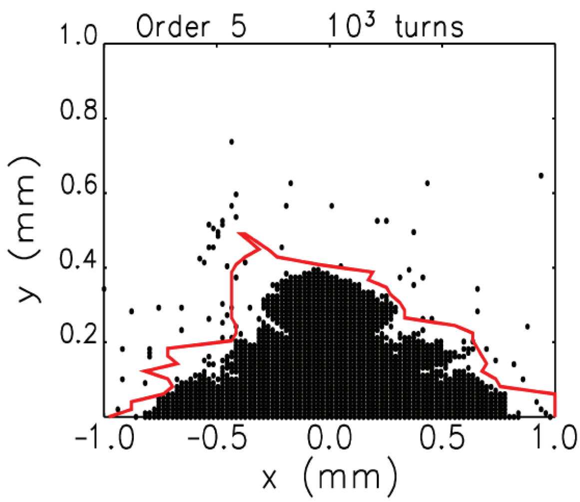

Figure 1: Dynamic aperture computed using 5th order one-turn tracking map with GHOST (black points) and with an element-by-element tracking code elegant [4] (red line). The agreement between the two codes is excellent.

(ii) infinitesimally short and (iii) transversally flat. In that case, the computationally intensive Poisson equation reduces to a closed-form solution amenable to efficient numerical implementation.

The generalized BE formalism applies only to an infinitely short bunch. Finite length is modeled by dividing every bunch in several slices, each of which is well-approximated by an infinitesimally short bunch. At every collision between the two bunches, each slice in one bunch collides with each slice in the other bunch according to the generalized BE formalism (Figure 2).

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 bunch. This means that the computational load associated with the collision of the two bunches scales linearly with the number of slices, and linearly with the number of particles.

When the beam's length is on the order of the beta function at the IP (β^*), the luminosity experiences a geometric reduction known as the *hourglass effect* [6]. We compute the hourglass effect in the JLEIC design and compare it to the analytic solution [6]. The agreement is excellent (Figure 3).

Parallelization on GPUs

We implemented the new beam-beam algorithm on a hybrid CPU/GPU platform, resulting in substantial overall speedup (Fig. 4). We used an NVIDIA Tesla K40 cards. The details of the implementation are reported in [7].

An important advantage of implementation of GHOST on GPUs as opposed to on CPUs is that the approximate Moore's law still applies to GPUs–each new generation of GPU cards which come out every 1-2 years usually double the computational power of the previous generation (this is no longer true for the CPUs). This means that the GPU

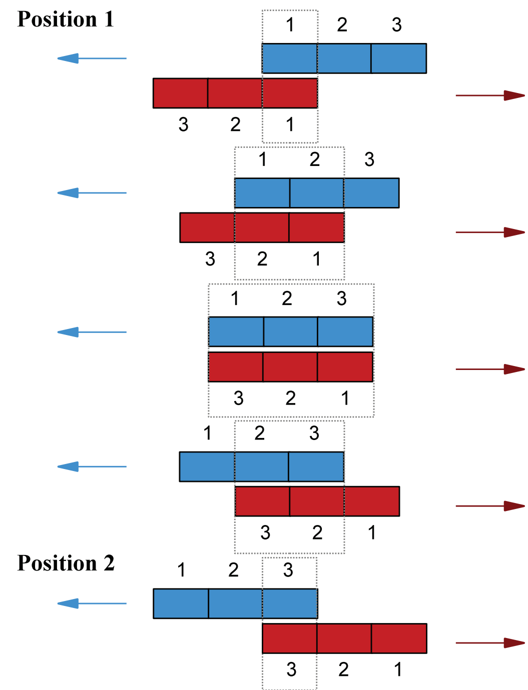

Figure 2: Collision between two multi-slice beams, starting at Position 1 and ending with Position 2. After each line, all slices in both beams drift in the direction of the arrow by a half of a slice width. Grey rectangles denote slices that are colliding at each time.

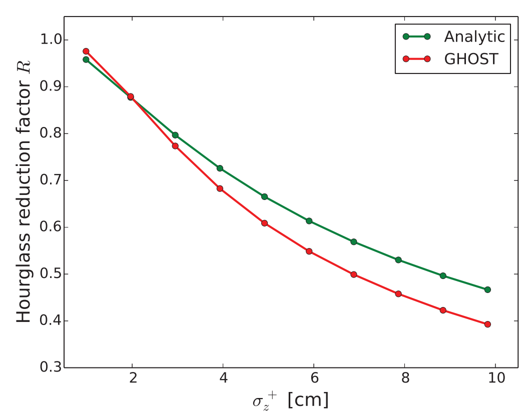

Figure 3: Hourglass effect computed using GHOST (128,000 particles and 10 slices) versus the analytical result [6].

codes such as GHOST will continue to benefit from this speedup in the foreseeable future.

"GEAR CHANGE"

Beam synchronization in electron-ion colliders has to take into account the different speeds at which the two beams propagate. The most efficient way to synchronize beams is to have a different number of bunches in each. This leads to

ISBN 978-3-95450-182-3

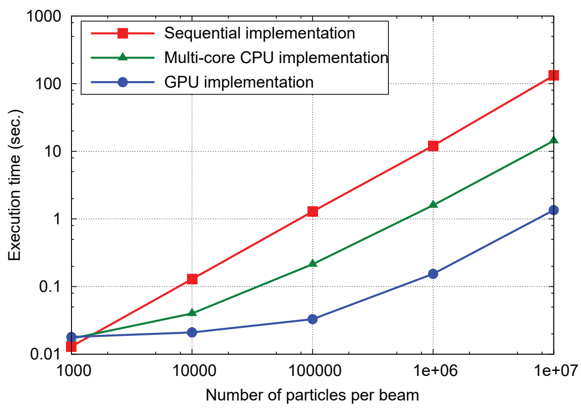

Figure 4: Execution time of the collision procedure in sequential (on a single CPU core), multi-core CPU (on 20 CPU cores), and GPU (on Tesla K40) implementation for varying number of particles with the number of slices fixed to $M = 6$ slices per bunch.

non-pair-wise collisions of beams with different number of bunches; if the number of bunches in two beams are mutually prime, all pairs of bunches collide. This type of synchronization, the so-called "gear change", is highly desirable because it reduces the magnet movement and RF adjustments. It also simplifies particle detection and polarimetry: (i) cancellation of systematic effects associated with bunch charge and polarization variation–great reduction of systematic errors, sometimes more important than statistics (ii) simplified electron polarimetry–only need average polarization, much easier than bunch-by-bunch measurement.

In the traditional case where the two colliding beams have the same number of bunches, a beam-beam simulation can reap considerable benefits from symmetry–only one-on-one bunch simulation captures the dynamics of the collider. This is because the same pair of bunches always collides at the interaction point (the bunches of the same beam sample the same distribution). In the case of "gear change", this symmetry is broken: for mutually prime numbers of bunches in beams, each bunch in one beam will see every bunch in the other beam consecutively. This means that *every* bunch in the beam should be simulated, an increase in the computational load proportional to the number of bunches in the beams.

The issue of dynamical stability of a "gear change" has been studied recently [8], with the conclusion that only a high-fidelity simulation can provide a definitive answer. With GHOST, we have the capability to carry out such a high-fidelity simulation. The tremendous computational load (for instance, in a JLEIC, each turn would consist of over 3000 pairs of bunches colliding) is alleviated by parallelizing collisions on GPUs. These simulations are currently being carried out; we will report the results in the near future.

FUTURE WORK

Systematic studies of small-scale (on the order of 3/4 or 10/11 bunches as in [8]) and large-scale (up to 3400 bunches as in JLEIC case [1]) "gear change" are currently underway. We will first make contact with the existing literature on the subject, most notably [8], and then study the stability of the "gear change" for JLEIC. The results from these simulations will be reported in a future publication.

A number of additional features are being developed and will be included in the next iteration of the code, including: (i) using fast multipole algorithm for collisions whenever Bassetti-Erskine approximation is not warranted; (ii) synchrotron damping; (iii) electron cooling of the ion beam (iv) intrabeam scattering; (v) space charge.

REFERENCES

[1] S. Abeyratne *et al.*, "MEIC design summary," `arXiv:1504.07961`, 2015.

[2] K. Makino and M. Berz, "COSY INFINITY version 8," *Nucl. Instr. Meth. A*, vol. 427, p. 338, 1991.

[3] M. Berz, "Nonlinear problems in future particle accelerators," M. Berz, Ed. Hackensack, NJ, USA: World Scientific Publishing, 1991.

[4] M. Borland, "elegant: a flexible SDDS-compliant code for accelerator simulation," Argonne National Laboratory, Argonne, USA, Rep. APS LS-287, 2000.

[5] M. Bassetti and G. Erskine, "Closed expression for the electrical field of a two-dimensional Gaussian change," CERN, Geneva, Switzerland, Rep. CERN-IRS-TH/80-06, 1980.

[6] M. Furman, "Hourglass effects for asymmetric colliders," in *Proc. Particle Accelerator Conf. (PAC'91)*, San Francisco, USA, May 1991, pp. 422.

[7] K. Arumugam, R. Majeti, D. Ranjan, M. Zubair, A. Godunov, and B. Terzić, "GPU-accelerated high-fidelity simulation of beam-beam effects in particle colliders," in *Summer Computer Simulation Conference*, 2017, submitted.

[8] Y. Hao, V. Litvinenko, and V. Ptitsyn, "Beam-beam effects of gear changing in ring-ring colliders," *Phys. Rev. ST Accel. Beams*, vol. 16, p. 101001, 2013.