HIGH-FIDELITY SIMULATIONS OF LONG-TERM BEAM-BEAM DYNAMICS ON GPUs

B. Terzić*, K. Arumugam, M. Aturban, C. Cotnoir, A. Godunov, D. Ranjan, M. Stefani, M. Zubair
Old Dominion University, Norfolk, Virginia, USA
F. Lin, V. Morozov, Y. Roblin, H. Zhang, Jefferson Lab, Newport News, Virginia, USA

Abstract

Future machines such as the Electron Ion Collider (MEIC), 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 typically requiring millions of turns. Until recently, most of the methods have involved using 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 sympletic way up to an arbitrary order. The code is called GHOST for GPU-accelerated High-Order Symplectic Tracking. 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.

INTRODUCTION AND BACKGROUND

The proper magnetic optics design and performance of a storage ring or a collider—such as the LHC, RHIC, LHeC, and electron-ion colliders—crucially depends on its its long-term dynamics. Approaches which approximate the long-term dynamical stability based on relatively short-term simulations do not provide the necessary level of confidence. Ultimately, to simulate accurately the beam dynamics in a storage ring or a collider, it is necessary to track the beam particles for millions to billions of turns—comparable to the beam lifetime. However, until the recent advent of the GPU technology, such long-term simulations have been prohibitively expensive computationally.

Long-term simulations require the tracking to be symplectic—invariants of motion must be explicitly preserved. A constant linear transfer map can be made trivially symplectic by ensuring that it satisfies the symplecticity criterion. Indeed, many “kick-drift” codes take advantage of this fact to perform a symplectic step-by-step integration of the particle’s equations of motion through a ring represented by a piecewise constant Hamiltonian. However, this approach is not suitable for long-term tracking due to the inherently large number of steps required for each particle turn around the ring. In order to attain the required efficiency, our new Gpu-accelerated Higher-Order Symplectic Tracking (GHOST) code uses a truncated single-turn non-linear Taylor map to track a particle while explicitly enforcing the symplecticity by solving a set of associated implicit, non-linear set of equations.

The beam collisions are described by the Poisson equation which can be solved by a number of methods at a high computational cost. To reduce the computational load, a number of approximations have been proposed. BEAM-BEAM3D [1] 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. Another approximation can be to assume a gaussian beam distribution which leads to a one-dimensional integration [2]. Finally Bassetti-Erskine (BE) [3] approach introduces one more level of approximation by assuming that the beams have vanishing length and a Gaussian transverse distribution. This reduces the Poisson equation to a single evaluation of a complex error function, which is computationally efficient. In GHOST, we use the BE formalism for beam interaction, generalized to an arbitrary geometry, which may also include upright and round beams (as opposed to flat beams originally derived in BE).

GHOST: ALGORITM DESCRIPTION

In GHOST, the beam bunches are represented by an appropriate Gaussian distribution of point particles, while the effect of the collision is computed using the generalized BE approximation. The new code is SDDS-compliant [4], which can be readily post-processed with the powerful SDDS tools.

Particle Tracking

Various established particle tracking codes feature lattice map generation techniques. Our code relies on maps generated by the well-established algorithms of COSY Infinity [5]. COSY Infinity generates Taylor maps of an arbitrary order for a given optical system by numerically integrating the equations of motion in the system using differential algebraic techniques. As a result, it generates coefficients $M(x|\alpha\beta\gamma\eta\lambda\mu)$ of the Taylor expansion of the form

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

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

[*] bterzic@odu.edu

ISBN 978-3-95450-136-6

indicates the reference value. The six variables form three canonically conjugate pairs.

For symplectic tracking, the initial and final coordinates must satisfy the symplectic condition, described as a partial differential equation constructed from the generating function of the dynamic system. Taking the second kind of generating function as an example, assuming the initial coordinates of a symplectic system are $(\boldsymbol{q}_i, \boldsymbol{p}_i)$ and the final coordinates of it are $(\boldsymbol{q}_f, \boldsymbol{p}_f)$, the coordinates satisfy

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

where

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

One can perform symplectic tracking by solving the above equations. The generating function F_2 can be derived from the truncated map $\mathbf{M}$ [6]. With $\mathbf{M}$ and F_2 known, first one calculates $(\boldsymbol{q}'_f, \boldsymbol{p}'_f)$ applying $\mathbf{M}$ on $(\boldsymbol{q}_i, \boldsymbol{p}_i)$, and then use $(\boldsymbol{q}_i, \boldsymbol{p}_i, \boldsymbol{q}'_f, \boldsymbol{p}'_f)$ as a start point to solve 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 [6].

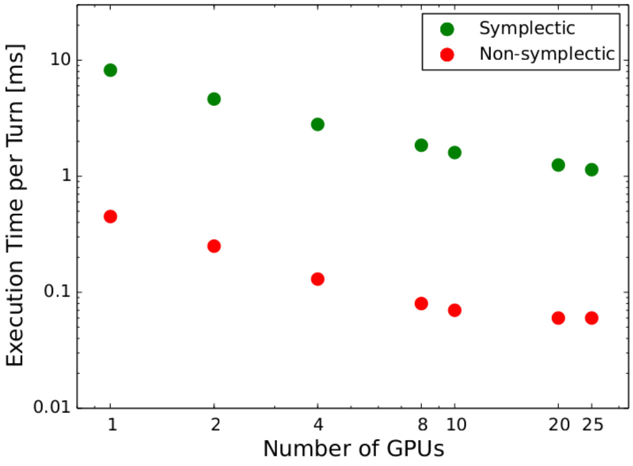

Figure 1: Execution time per turn as a function of the number of GPU devices for symplectic (green dots) and non-symplectic (red) tracking of 3rd order for 100000 particles.

Beam Collision

Because of this efficiency, the BE model [3] at the heart of a beam-beam code gives us the best chance of accurately studying the long-term dynamics in colliders. BE approximation greatly reduces the computational cost of the beam-beam interaction when the interacting bunches are: (i) well-approximated by a Gaussian transverse distribution, (ii) infinitesimally short and (iii) transversally flat. In that case, the computationally expensive Poisson equation exactly reduces to an inexpensive complex error function.

The first approximation—that the bunches are Gaussian—is reasonable because we are interested in the steady-state solution for which we have a stable long-term behavior. Some machines, such as LHC, exhibit strong beam-beam distortions on both beams produced by coherent and incoherent effects as well as long-range interactions. We plan to extend GHOST by implementing a hybrid fast multipole method to address these cases.

The second approximation—that the bunches are infinitesimally short—is resolved by dividing bunches into several transverse slices and treating each as infinitesimally short. GHOST then employs the synchro-beam mapping, the symplectic beam-beam map usable for the long bunch [7,8]. The collision between the two beams at the interaction point is simulated by collisions of individual slices.

The third approximation—that the bunches are flat—is relaxed by deriving the generalized solutions for upright and (nearly) round bunches.

GPU Implementation

The amount of computations required to track and collide bunches over $10^7 - 10^9$ turns is prohibitive. In serial mode, the problem would simply be computationally intractable, with simulation taking years to run. This clearly motivates the use of sophisticated, finely-tuned algorithms running on massively-parallel platforms.

The parallel particle tracking in GHOST is implemented on a hybrid CPU-GPU platform using CUDA (Compute Unified Device Architecture), taking full advantage of the highly repetitive nature of calculations performed. In particular, one portion of the code—the setup, initialization and I/O—run on the traditional CPU platform, while computationally intensive components of the beam tracking and collision execute on a cluster of NVIDIA's GPUs. Cluster parallelization is implemented through MPI.

Our numerical experiments were carried out on a cluster of Kepler K20 GPUs with GK110 processor equipped with 5GB of GDDR5 memory. In tracking mode, GHOST achieves maximum speedup (CPU time/GPU time) on a single GPU device of over 280. The speedup scales nearly linearly with multiple GPU devices.

Figure 1 shows the scalability of tracking code on a cluster of GPUs when tracking 100000 particles with a 3rd order map. Symplectic tracking is, as expected, consistently more expensive because of the symplectic correction at each step (which requires solving a non-linear set of equations). The particles are equally distributed between the available GPUs in the cluster and the absence of communication between the threads and also between the GPUs provides a linear scalability with the number of GPUs. The scalability continues to be linear until the number of particles assigned to each GPU is sufficient enough to keep the device busy by maximizing the utilization.

At this level of performance, tracking 100000 particles for 400 million turns—corresponding, for instance, to one hour of beam lifetime of the proposed MEIC—on 25 GPUs takes about 4.5 days in symplectic and about 7 hours in non-symplectic 3rd order tracking mode. It is expected that the performance improves with each new GPU models (at this time there are two more recent and more powerful GPUs on the market — Tesla K40 and Tesla K80).

BENCHMARKING

At this stage of the development, only the tracking (symplectic and non-symplectic) mode of GHOST is GPU-optimized. The collision mode has been fully developed as a prototype and is currently being implemented and optimized in CUDA. Therefore, the results for tracking (Figs. 1 and 2) have been done in GHOST, and the results for beam-beam collision (Figs. 3 and 4) on the serial prototype. While the simulations reported here are for a generic collider ring design (tracking figures) and the electron-ion design similar to the proposed MEIC [9] (beam-beam figures), the results are general.

We successfully carried out these benchmarks for GHOST:

- Particle tracking—both symplectic and non-symplectic—in GHOST is equivalent to that in COSY Infinity, as illustrated in Fig. 2.
- In beam-beam simulations, the results—for example, luminosity shown in Fig. 3—converge as the number of transverse slices in the simulation are increased.
- In beam-beam simulations, the reduction factor in luminosity due to the hourglass effect shows excellent agreement with the analytic estimate [8, 10] (Fig. 4).

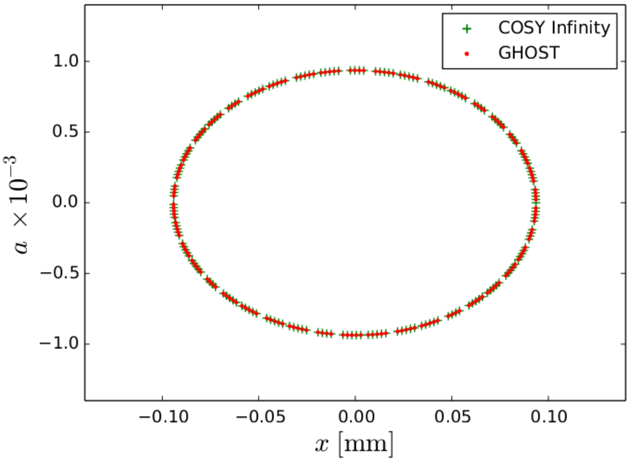

Figure 2: A particle tracked using 3rd order symplectic tracking for 2 million turns and recorded every 10000 turns with COSY Infinity (green crosses) and GHOST (red dots).

DISCUSSION

MEIC envisions a synchronization scheme which involves having different harmonic numbers for the electron and ion rings. This will give raise to gear changing effects which will have to be studied carefully to assess stability and proper damping of these resonances. GHOST will allow for an arbitrary pattern of bunches for both rings. As a benefit one also will be able to account for clearing gaps in the ion bunch pattern.

Additional features currently under development that will be a part of the next iteration include: synchrotron damping, cooling of the proton beam by a low-energy electron beam, intra-beam scattering and crabbing.

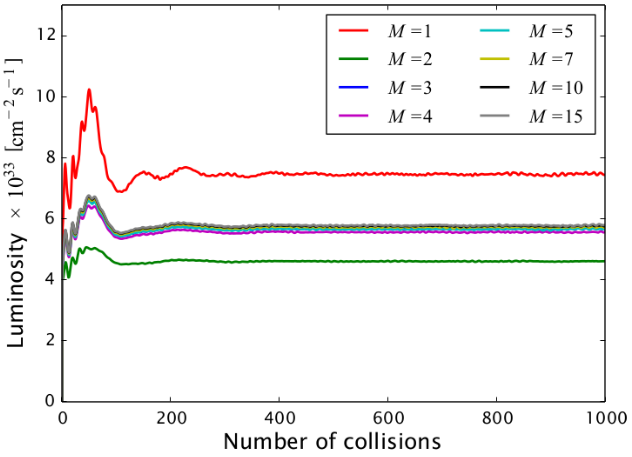

Figure 3: Luminosities computed with GHOST for different numbers of slices M with 40000 particles for the MEIC parameters [9] in which the beam bunches are three times longer than the nominal values, with the reduction factor due to hourglass of 23% (the third point from the left on the red curve in Fig. 4).

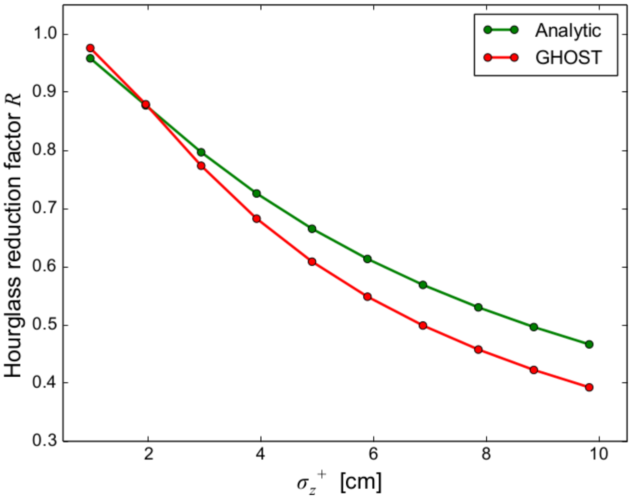

Figure 4: The hourglass effect as a function of the initial beam size, computed in a simulation with 128000 particles and 10 slices in each beam using GHOST (red line) and the analytic expression [8, 10] (green line). The layout used is that of the MEIC [9], with the constant ratio σ_z^+/σ_z^-.

ACKNOWLEDGMENT

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. This paper is authored by Jefferson Science Associates, LLC under U.S. Department of Energy (DOE) Contract No. DE-AC05-06OR23177.

REFERENCES

[1] J. Qiang, R. Ryne and M. Furman, PR STAB 5, 104402 (2002).

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

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

[4] M. Borland, Proceedings of ICAP (1998).

[5] K. Makino and M. Berz, *COSY INFINITY version 8*, Nuclear Instruments and Methods, A427:338–343 (1999).

[6] M. Berz, *Nonlinear Problems in Future Particle Accelerators*, World Scientific Publishing, Hackensack, NJ (1991).

[7] K. Hirata, H. Moshammer and F. Ruggerio, Particle Accelerators 40, 205 (1993).

[8] A. W. Chao and M. Tigner, *Handbook of Accelerator Physics and Engineering*, World Scientific Publishing, Hackensack, NJ (2009).

[9] Y. Roblin, V. Morozov, B. Terzić, M. Aturban, D. Ranjan and M. Zubair, Proceedings of IPAC (2013).

[10] M. Furman, Proceedings of PAC (1991).

ISBN 978-3-95450-136-6