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

B. Terzić[1,2], V.S. Morozov [1], Y.R. Roblin[1], F. Lin[1], H. Zhang[1], M. Aturban[2], D. Ranjan[2], M. Zubair[2]

[1]*Jefferson Lab*, [2]*Old Dominion University*

## 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 [1].

## Introduction

Beam-beam interaction is one of the most important dynamical factors limiting the collider's luminosity and therefore limiting its scientific efficiency.

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 approximations.

We generalize Bassetti-Erskine (BE) solution [2] of the Poisson equation which is exact for flat, infinitesimally short bunches with gaussian transverse distribution to other geometries (upright and flat), and to strong-strong mode. The resulting model optimized on a massively parallel GPU platform enables long-term simulations of beam-beam effects on the order of billion collisions.

## Parallelization on GPUs

We implemented the new beam-beam algorithm on a hybrid CPU/GPU platform, resulting in substantial overall speedup. We used an NVIDIA Tesla M2090 card with 512 cores. Each CPU hosts 4 GPUs.

Tracking algorithm results in a maximum speedup on a single GPU device of over 280. The speedup scales nearly linearly with multiple GPU devices.

We are currently benchmarking the full collision mode. Our preliminary results are encouraging and will be soon reported.

## Acknowledgements

Authored by Jefferson Science Associates, LLC under U.S. DOE Contract No. DE-AC05-06OR23177 and DE-AC02-06CH11357..

## Convergence

When the finite longitudinal size of particle bunches 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 (the computational load scales linearly with the number of slices *M*).

The differences between the *M* = 1 and *M* > 1 results are due to the hourglass effect. For cases where the hourglass effect is either minimal, as in Fig. 1, or moderate, simulations with *M* ≥ 3 yield virtually the same results, indicating rapid algorithm convergence.

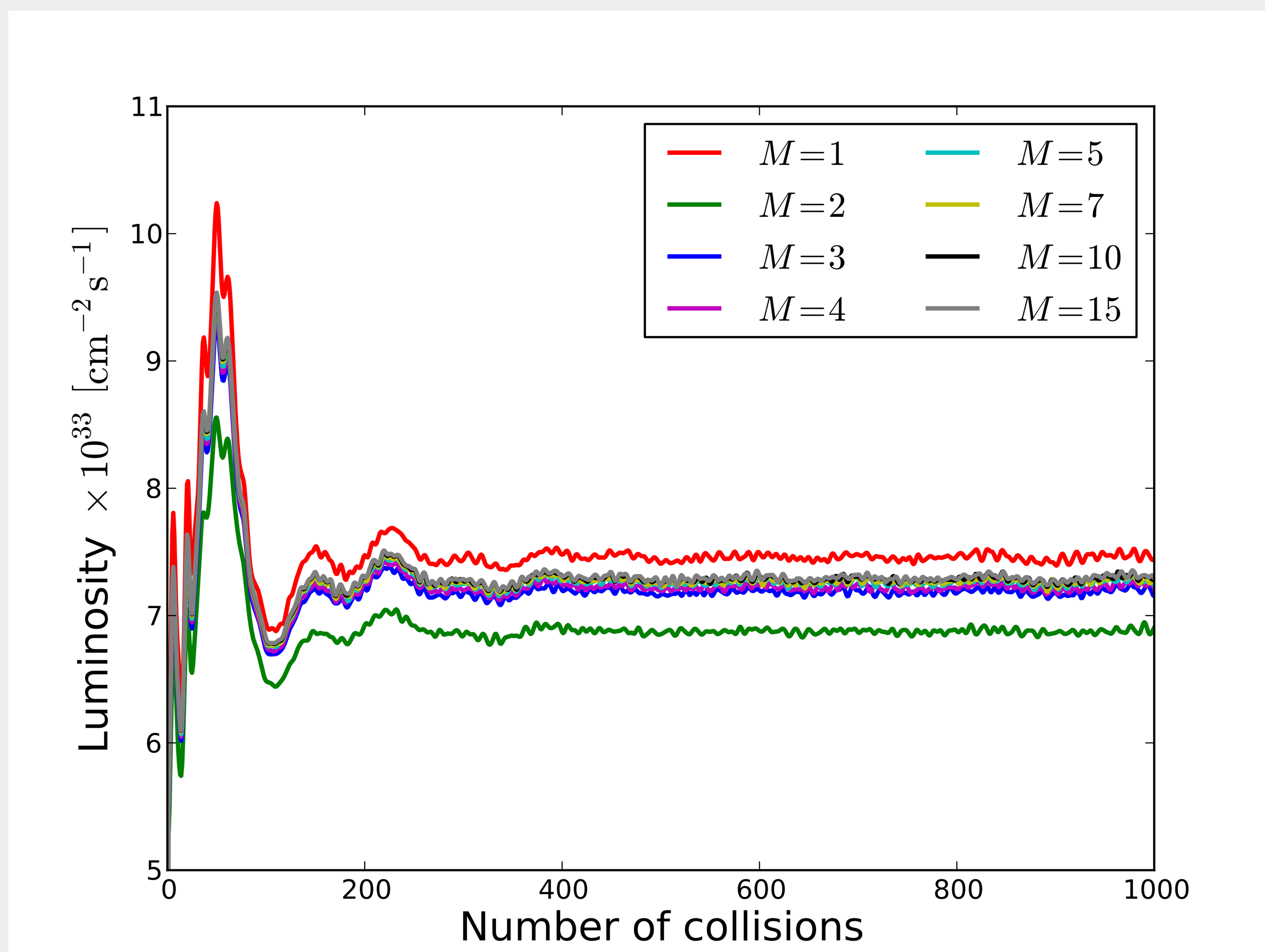


**Fig. 1**: MEIC lattice with *N*=40000 particles.

## Hourglass Effect

When the beam's length is on the order of the beta function at the IP ($\beta^*$), the luminosity experiences a geometric reduction known as the *hourglass effect* [3].

We compute the hourglass effect in the MEIC design and compare it to the analytic approximation [3]. The agreement is excellent.

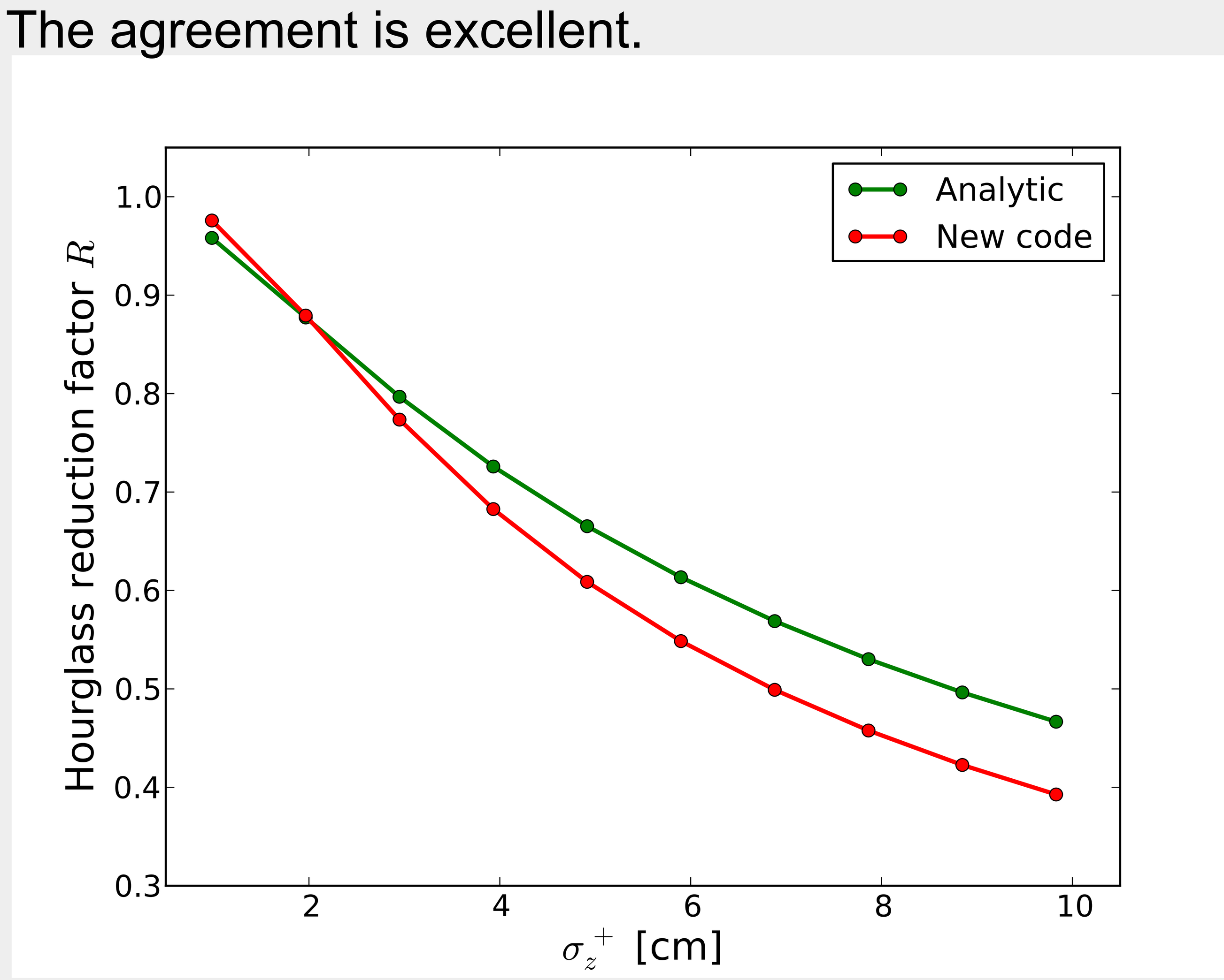


**Fig. 2:** *N*=128000 particles, *M*=10 slices, MEIC lattice scaled (nominal design is given by $\sigma_z^+$= 1 cm).

## Future Work

A number of additional features are being developed and will be included in the next iteration, including:

- synchrotron damping
- cooling of the proton beam by an electron beam
- intrabeam scattering

The new code will be used for high-fidelity, long-term simulations of the MEIC design.

## Particle Tracking

The particle transport through the ring is carried out using an arbitrary-order Taylor map ***M*** generated by COSY Infinity [4].

Symplectic tracking option is implemented using the generating function $F_2$ [5]:

$$(q_f, p_i) = J \cdot \nabla F_2(q_i, p_f) \quad (1)$$

with

$$J = \begin{pmatrix} 0 & -I \\ I & 0 \end{pmatrix}.$$

Given the generating function $F_2$ and the corresponding truncated map ***M***, we first calculate $(q_f', p_f')$ by applying ***M*** to $(q_i, p_i)$, and then use $(q_i, p_i, q_f', p_f')$ as a starting point for solving Eq. (1) numerically. Because $(q_f', p_f')$ is very close to $(q_f, p_f)$, Eq. (1) can be solved to machine accuracy in a few iterations.

## Particle Collision

The generalized BE formalism applies only to an infinitely short bunch. Finite length is modeled by dividing each 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.

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.*

## References

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

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

[3] M. Furman, Proc. PAC 1991

[4] K. Makino and M. Berz, NIM A427, 338 (1999)

[5] M. Berz, in *Nonlinear Problems in Future Particle Acclerators,* 288 (1991)