High-Fidelity Simulations of Long-Term Beam-Beam Dynamics on GPUs

Balša Terzić

Department of Physics, Old Dominion University
Center for Accelerator Science, Old Dominion University

International Computational Accelerator Physics Conference, Shanghai, 13 October 2015

Collaborators

Center for Accelerator Science at Old Dominion University:

Professors:

Physics: Alexander Godunov

Computer Science: Mohammad Zubair, Desh Ranjan

Students:

Physics: Chris Cotnoir, Mark Stefani

Computer Science: Kamesh Arumugam

Jefferson Lab (CASA) Collaborators:

Vasiliy Morozov, He Zhang, Fanglei Lin, Yves Roblin

Interdisciplinary collaboration

Outline

- Background
  - Motivation
  - Computational requirements and challenges
- GHOST: New GPU-Optimized Beam-Beam Code
  - Outline of the algorithm
    - Particle tracking
    - Beam collision
  - Benchmarks
  - Present and future capabilities
- Outlook and Conclusion

Motivation

- Design and performance of particle colliders depend crucially on their long-term dynamics
- Beam-beam effect has been particularly limiting to the long-term stability and high luminosity reach
- Extracting long-term behavior from a short-term simulation does not provide the necessary level of confidence
- Need to simulate the dynamics for intervals which are comparable to the beam lifetime
  - Hundreds of millions to billions of turns
- Until recently such long-term simulations have been prohibitive
  - Parallel computation on GPUs is changing this

Computational Requirements

- Perspective: At the current layout of the MEIC
  1 hour of machine operation time ≈ 400 million turns

- Requirements for long-term beam-beam simulations
  ① Accurate and efficient particle tracking
  ② Efficient beam collision simulation

- We meet these requirements by
  ① High-order symplectic tracking
    - One-turn maps + symplectic correction
  ② Approximate beam-beam collisions by generalizing strong-strong Bassetti-Erskine approximation
    - Poisson solvers are much more expensive

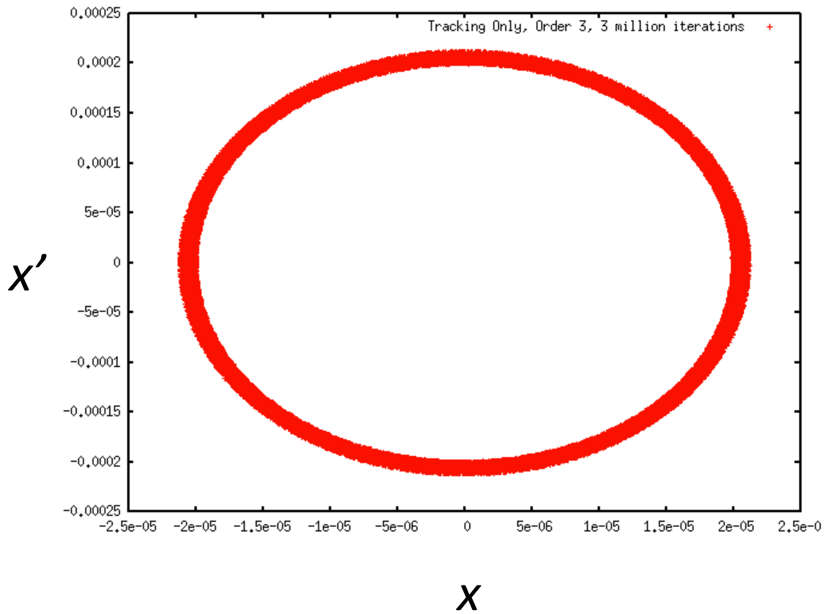
GHOST: Outline

- GHOST: Gpu-accelerated High-Order Symplectic Tracking
- Resolve computational bottlenecks by
  - Employing approximations (Bassetti-Erskine for collisions)
  - Implementing the code on a massively-parallel GPU platform
- GPU implementation yields best returns when:
  - The same instruction for multiple data (particle tracking)
  - No communication among threads (particle tracking)
  - Done in close collaboration with field experts
    - Physicists → proof of concept, CS → implementation
- Two main parts:
  - Particle tracking
  - Beam collisions

GHOST: Particle Tracking

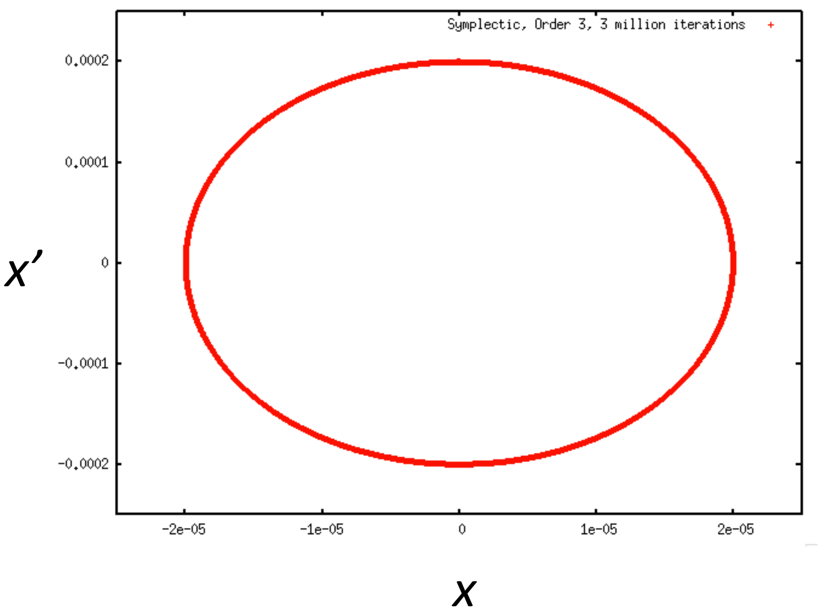
- Symplectic tracking is essential for long-term simulations

GHOST: Non-Sympletic Tracking
3 million iterations, 3rd order map



Energy not conserved
Particle will soon be lost

GHOST: Sympletic Tracking
3 million iterations, 3rd order map



Energy conserved

GHOST: Symplectic Particle Tracking

- Symplectic tracking in GHOST is the same as in COSY Infinity (Makino & Berz 1999)
- Start with a one-turn map

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

- Symplecticity criterion enforced at each turn

$$(\boldsymbol{q}_f, \boldsymbol{p}_i) = \mathbf{J}\nabla F_2(\boldsymbol{q}_i, \boldsymbol{p}_f) \qquad \mathbf{J} = \begin{bmatrix} 0 & -\mathbf{I} \\ \mathbf{I} & 0 \end{bmatrix}$$

Initial coordinates $(\boldsymbol{q}_i, \boldsymbol{p}_i)$ Final coordinates $(\boldsymbol{q}_f, \boldsymbol{p}_f)$

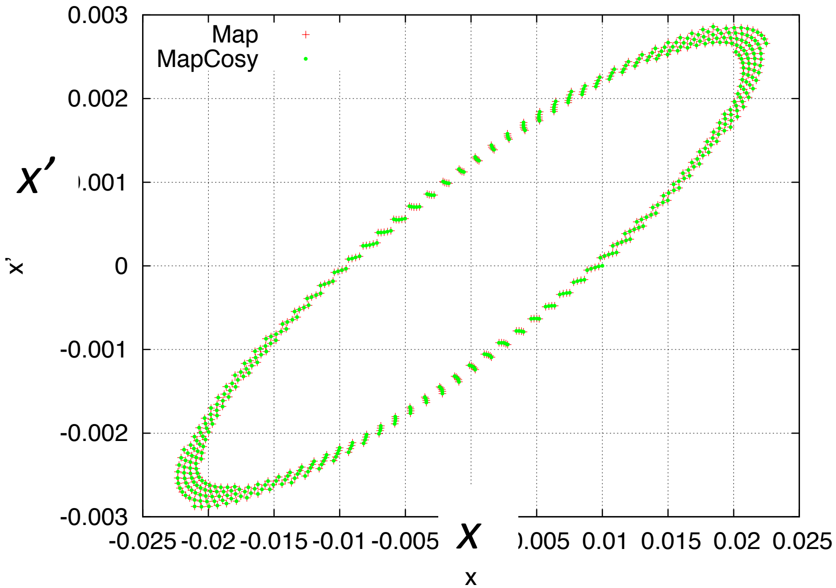
- Involves solving an implicit set of non-linear equations
  - Introduces a significant computational overhead

GHOST: Symplectic Particle Tracking

- Symplectic tracking is implemented as in COSY Infinity
  (Makino & Berz 1999)

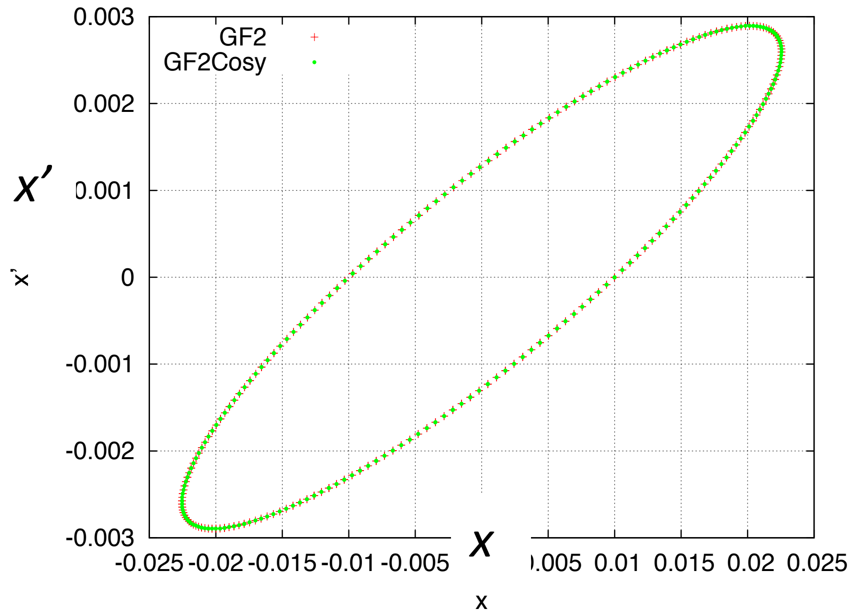
2D Non-Sympletic Tracking 3rd order map

COSY GHOST



2D Sympletic Tracking 3rd order map

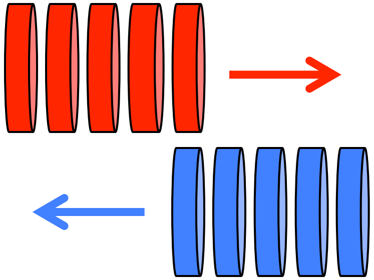
COSY GHOST



Perfect agreement!

GHOST: Beam Collisions

- Bassetti-Erskine approximation (Bassetti & Erskine 1980)
  - Beams treated as 2D transverse Gaussian slices (Good approximation for the MEIC)
  - Poisson equation reduces to a complex error function
  - Finite length of beams simulated by using multiple slices

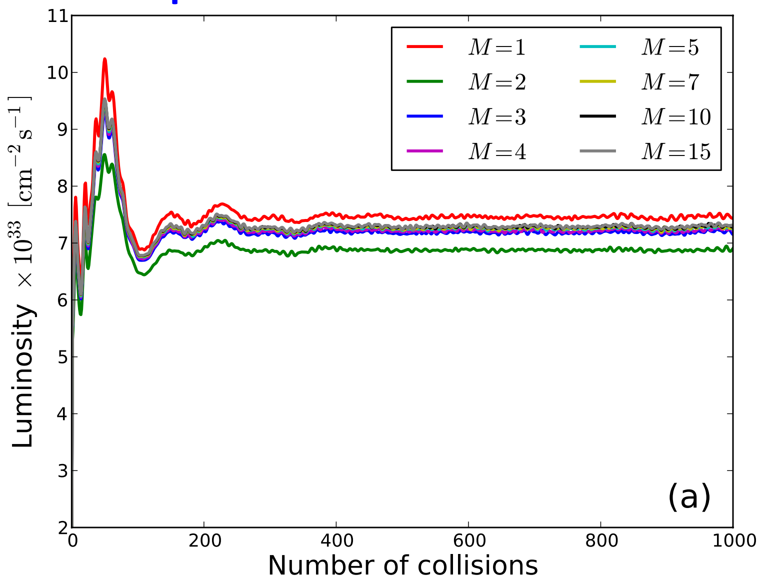


- We generalized a "weak-strong" formalism of Bassetti-Erskine
  - Include "strong-strong" collisions (each beam evolves)
  - Include various beam shapes (original only flat beams)

GHOST: Beam Collisions

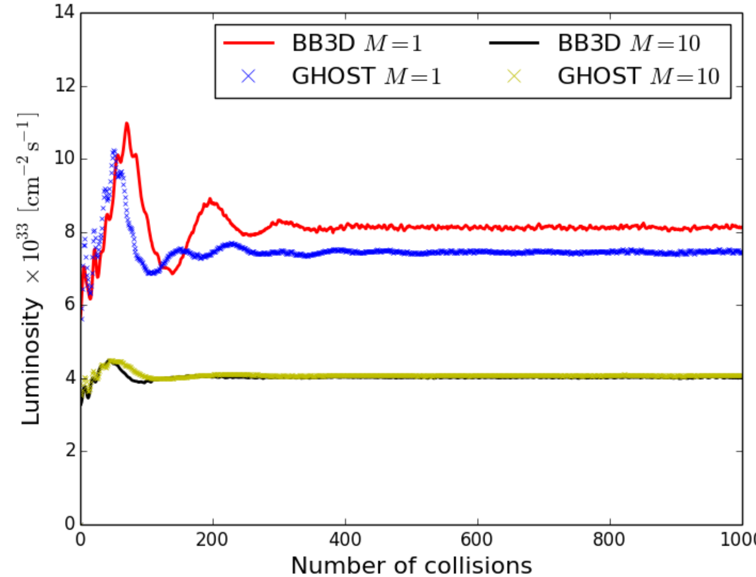
- Code calibration and benchmarking
  - Convergence with increasing number of slices *N*
  - Comparison to BeamBeam3D (Qiang, Ryne & Furman 2002)

GHOST, 1 cm bunch
40k particles



Finite bunch length accurately represented

BeamBeam3D & GHOST, 10 cm bunch
40k particles

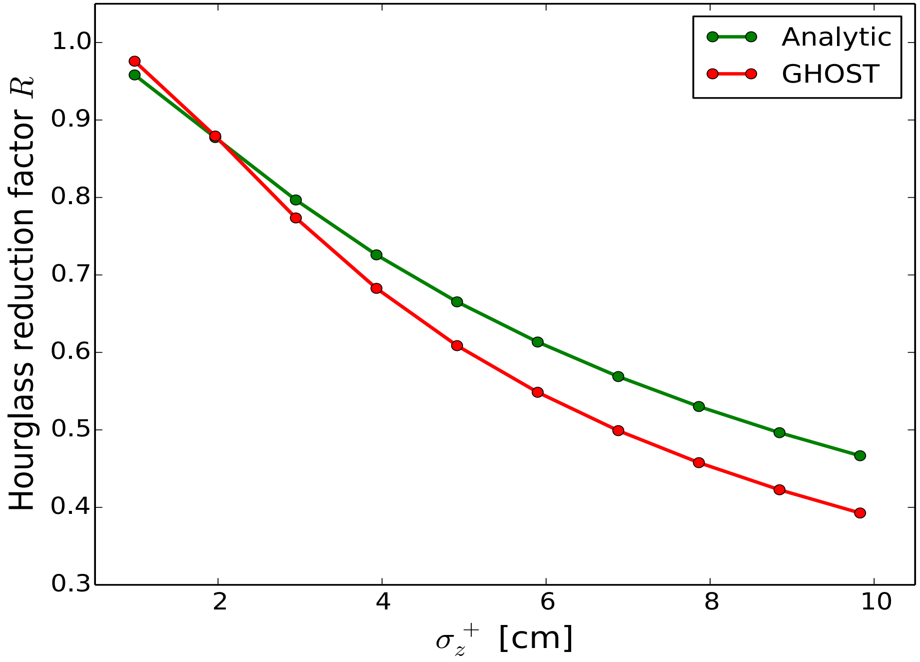


Excellent agreement with BeamBeam3D

GHOST Benchmarking: Hourglass Effect

- When the bunch length $\sigma_z \approx \beta^*$at the IP, it experiences a geometric reduction in luminosity – the *hourglass effect* (Furman 1991)

GHOST, 128k particles, 10 slices

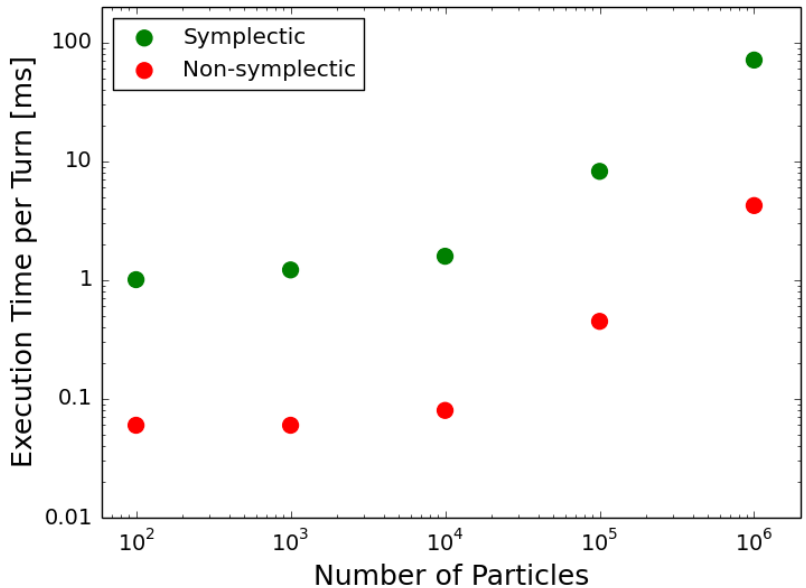


Excellent agreement with theory

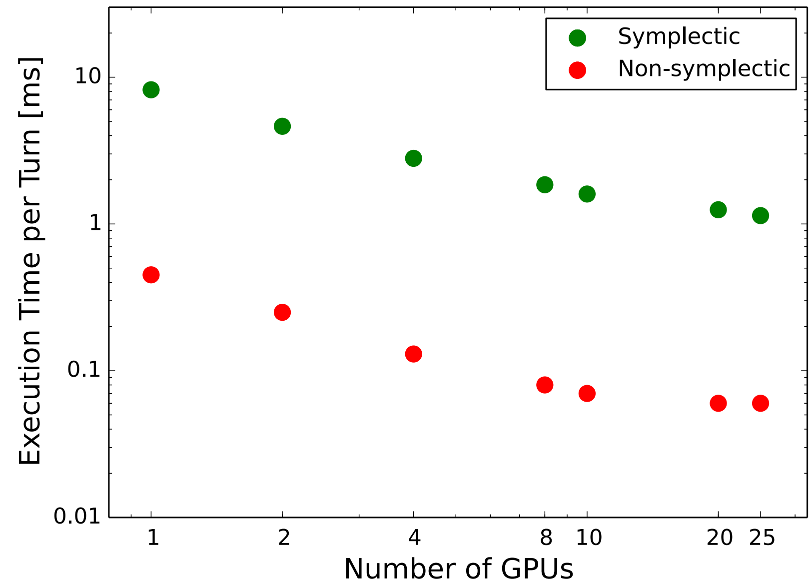
GHOST GPU Implementation

GHOST: 3rd order tracking

1 GPU, varying # of particles



100k particles, varying # of GPUs



400 million turns in an MEIC ring for a bunch with 100k particles:

> 7 hours for non-symplectic tracking

~ 4.5 days for symplectic tracking

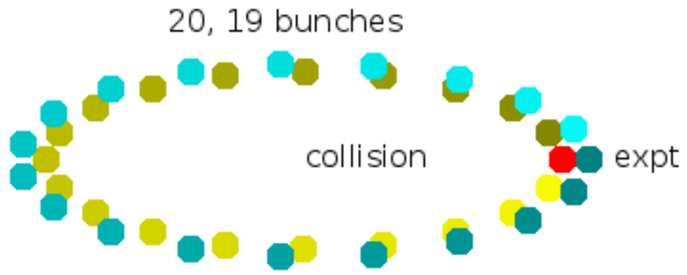
Current and Future Efforts

- Beam-beam collisions on GPUs
  - Finish implementation and optimize

- Other effects to be considered and implemented
  - Synchrotron damping
  - Cooling of the proton beam by an electron beam
  - IBS
  - Space charge
  - Other options for collisions? (fast multipole)
  - Beam synchronization
    (arbitrary arrangement of colliding bunches)

Future Challenges: Beam Synchronization

- MEIC design has to deal with beam synchronization
  - Non-pair-wise collisions of beams with different number of bunches (N_1, N_2) in each collider ring ("gear-change")
    - Simplifies detection and polarimetry
    - Beam-beam collisions precess
    - If N_1 and N_2 are incommensurate, all combinations of bunches collide
    - Can create linear and non-linear instabilities (Hirata & Keil 1990; Hao *et al*. 2014)



- Gear-change requires many collisions per crossing (~3420)
  - The load can be alleviated by implementation on GPUs
  - The information for all bunches stored: huge memory load
  - More interesting computer science problem: truly parallel!

Conclusion

- GHOST: code for long-term beam-beam simulations
  - Efficiency for long-term simulations achieved by
    - GPU implementation in CUDA C
    - Beam-beam kicks modeled with Bassetti-Erskine approximation
  - Comparison with existing codes instills confidence
    - Symplectic and non-symplectic tracking equivalent to that of COSY Infinity
    - Beam collision mode is in excellent agreement with BeamBeam3D
  - SDDS-compliant (Borland 1998)

- GHOST is a modular platform for beam-beam simulations
  - Easy implementation of new modules and functionalities
  - Particular challenge: beam synchronization for MEIC

Backup Slides

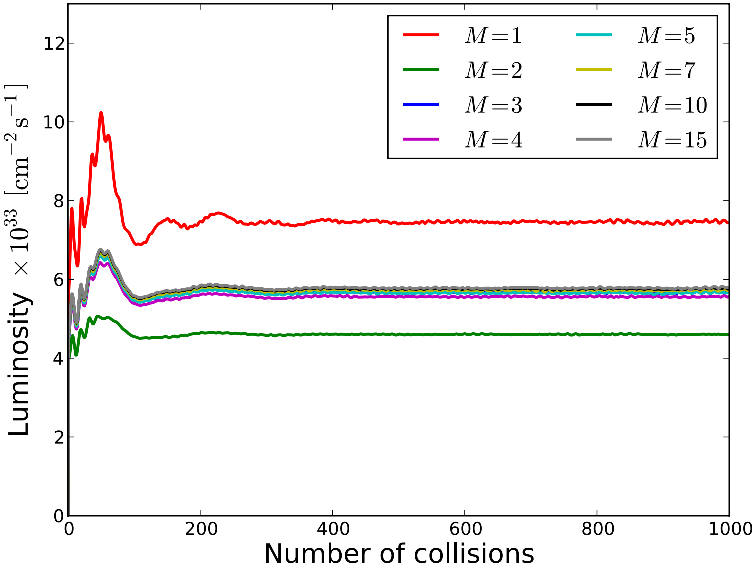
MEIC Design Parameters Used

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.1 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	$\text{cm}^{-2}\text{s}^{-1}$	0.562×10^{34}	
Reduction (hourglass)		0.957	
Peak luminosity with hourglass effect	$\text{cm}^{-2}\text{s}^{-1}$	0.538×10^{34}	

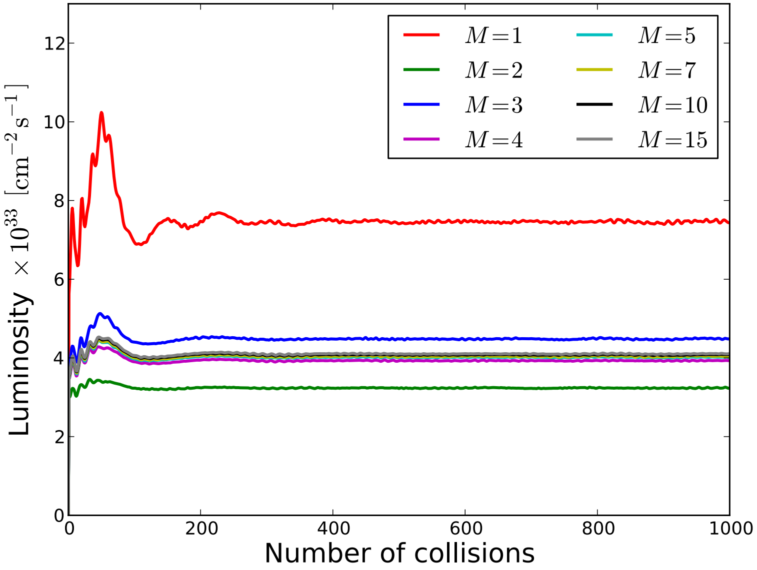
GHOST: Beam Collisions

- Code calibration and benchmarking
  - Convergence with increasing number of slices *N*

GHOST, 3 cm bunch
40k particles



GHOST, 10 cm bunch
40k particles



Convergence confirmed

GHOST GPU Implementation

GHOST Tracking on 1 GPU