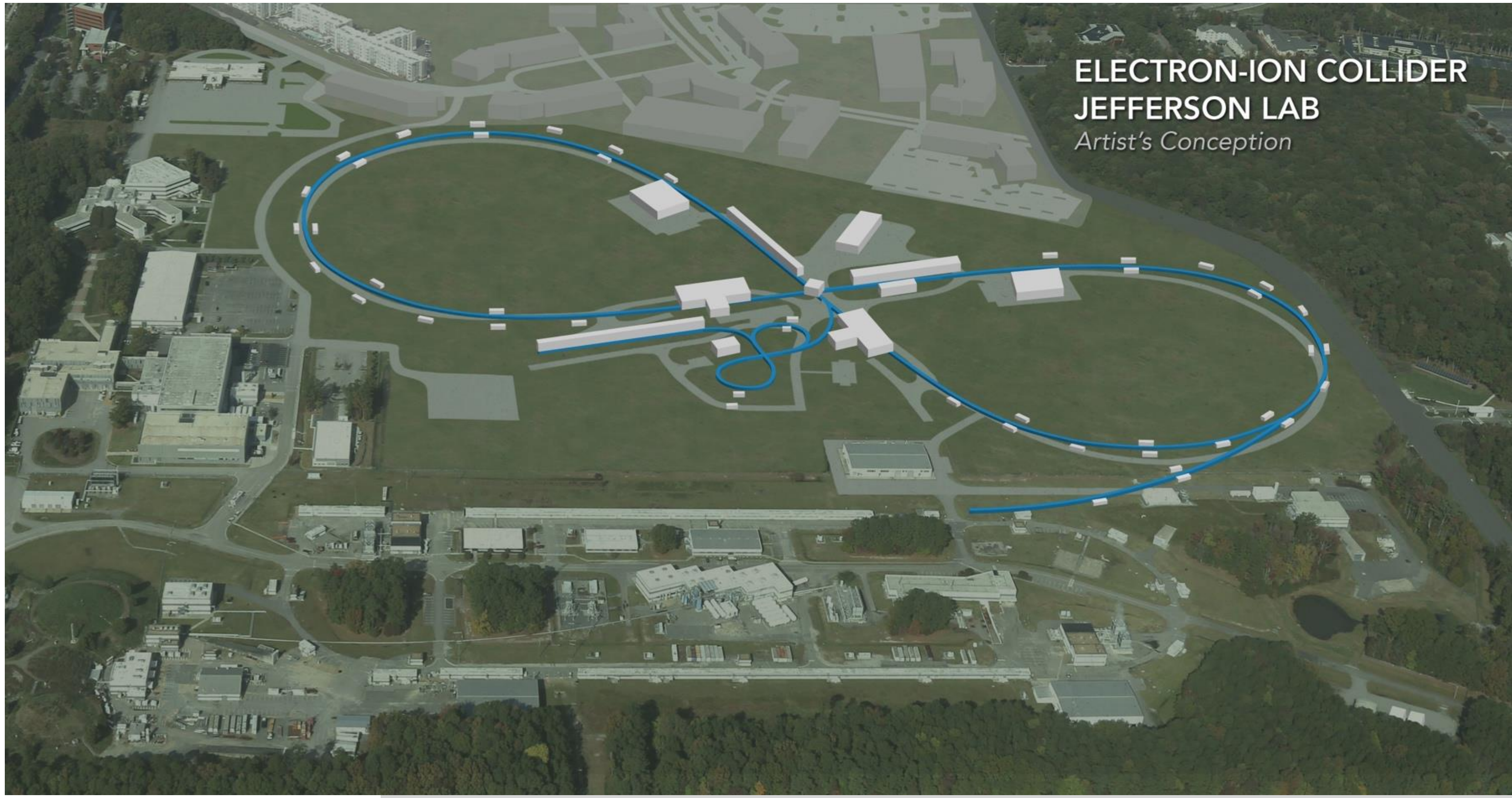
Beam-Beam dynamics with gear changing

Yves Roblin, Jefferson Lab

He Zhang, Jefferson Lab

Balsa Terzic, ODU



Jefferson Lab

OLD DOMINION UNIVERSITY IDEA FUSION

U.S. DEPARTMENT OF ENERGY Office of Science

JSA

Beam-Beam Dynamics with Gear Changing

- **Beam-Beam Dynamics with Gear Changing (row 4 – High A),JLAB FY17 Base R&D**

Description:

JLEIC has adopted an asymmetric colliding scheme in the baseline design (dubbed gear changing) where the harmonic numbers for the electron and ion rings are different. Thus, each bunch from one ring will collide with different bunches from the other ring in different turns. This allows for timing synchronization of the hadron with the electrons. It also provides a number of advantages for nuclear physics, namely a more accurate determination of the beam polarization. This scheme brings challenges to the machine design as it may lead to dynamic beam instabilities. There is no available simulation code that can currently simulate this gear changing scheme given the harmonic numbers involved. We are developing a new code, GHOST which is a beam-beam code implemented to take advantage of GPU enabled supercomputers in order to perform these calculations.

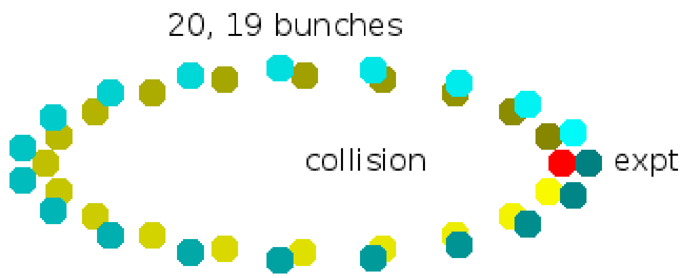
Goals:

- Deliver a beam-beam simulation code able to simulate gear-changing effects.
- Benchmark code relative to existing code for regular collisions and small number of bunches.

Status: Completed

"Gear Change" with GHOST

- "Gear change" provides beam synchronization for JLEIC
  - Non-pair-wise collisions of beams with different number of bunches (N_1, N_2) in each collider ring (for JLEIC $N_2 = N_1$-2 ~ 3420)
    - Simplifies detection and polarimetry
    - 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)



- The load can be alleviated by implementation on GPUs
  - The information for all bunches is stored: huge memory load!

- Approach
  - Implement single-bunch collision right and fast
  - Collide multiple bunches on a predetermined schedule
    - N_{bunch} different pairs of collisions on each turn

Milestones

1. Finish the development and testing of a GPU-parallelized code symplectic and non- symplectic particle tracking only.
2. Demonstrate the long-term symplectic tracking for a sample collider lattice.
3. Write up and submit for publication (tracking feature only).
4. Integrate into the code of a slice-by-slice beam collision algorithm based on Bassetti-Erskine approximation and benchmark.
5. Optimize the code including both (non-)symplectic tracking and a slice-by-slice collision on a single GPU.
6. Optimize the code including both (non-)symplectic tracking and a slice-by-slice collision on a multiple GPUs.
7. Benchmark the code on bunch by bunch mode for kinematics relevant to JLEIC.
8. Benchmark the code on multi-bunch mode (gear-changing) for a small number of bunches.

Beam-Beam Dynamics with Gear Changing

- Budget

	FY'17	Totals
a) Funds allocated	$179,000	$179,000
b) Actual costs to date	$179,000	$179,000

- Deliverables and schedule

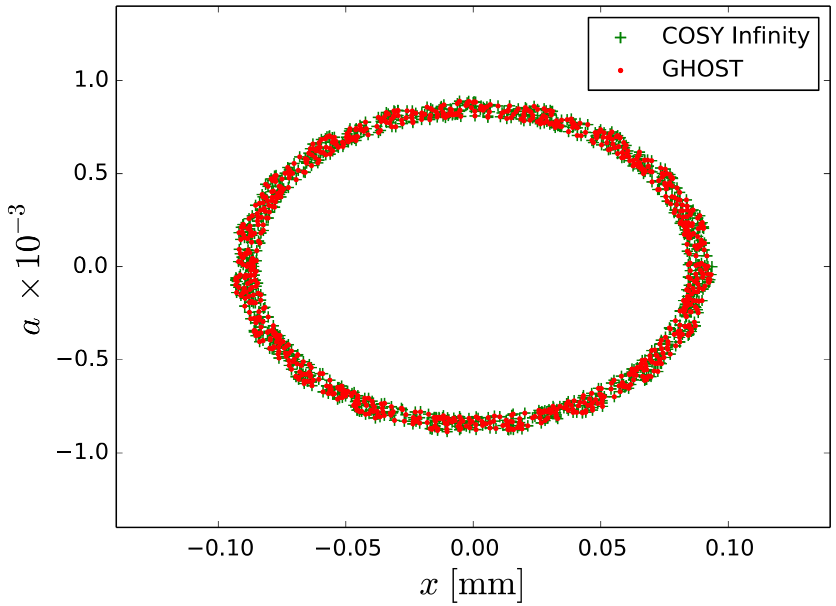
Task	FY'17 Q1	FY'17 Q2	FY'17 Q3	FY'17 Q4
Demonstrate symplectic tracking 1,2,3	*X*			
Implement slice-by-slice beam-beam collisions 4		X	X	X
Optimize code on single and multiple GPUs 5,6				X
Benchmark code on JLEIC kinematics 7,8				X

- The project corresponds to Line 4 "Beam-Beam Dynamics with Gear Changing", Priority High-A of the Jones' Panel report

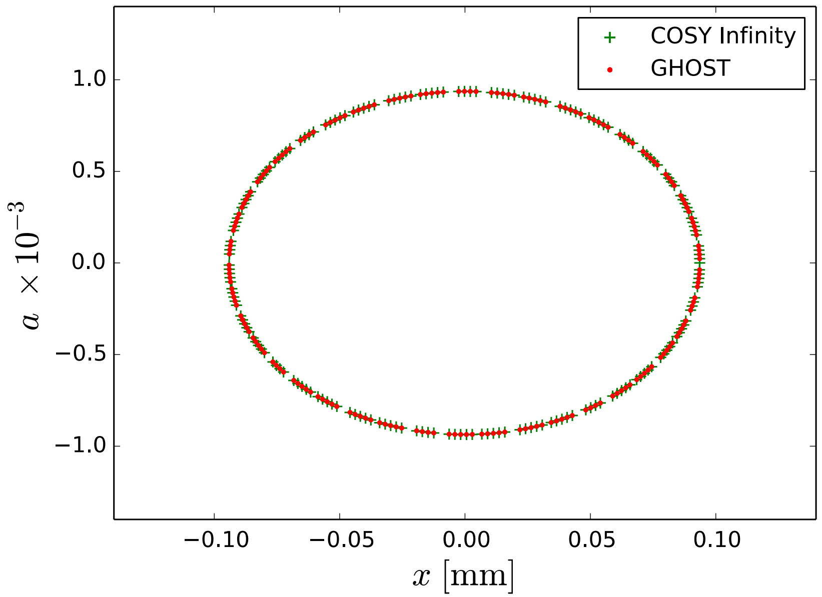
Symplectic Tracking With GHOST (milestone 1)

- Symplectic tracking in GHOST is the same as in COSY Infinity
  (Makino & Berz 1999)

Non-Sympletic Tracking 3rd order map
COSY GHOST 100,000 turns



Sympletic Tracking 3rd order map
COSY GHOST 100,000 turns



Perfect agreement!

Methodology[1,2]

Assuming a dynamic system, the initial and final coordinates of which are $(\boldsymbol{q}_i, \boldsymbol{p}_i)$ and $(\boldsymbol{q}_f, \boldsymbol{p}_f)$, the symplecticity is preserved if the coordinates satisfied a nonlinear implicit partial differential equation as follows:

$$
\begin{aligned}
(\vec{p}_i, \vec{p}_f) &= (\vec{\nabla}_{q_i} F_1, -\vec{\nabla}_{q_f} F_1) \\
(\vec{p}_i, \vec{q}_f) &= (\vec{\nabla}_{q_i} F_2, \vec{\nabla}_{P_f} F_2) \\
(\vec{q}_i, \vec{p}_f) &= (-\vec{\nabla}_{p_i} F_3, -\vec{\nabla}_{q_f} F_3) \\
(\vec{q}_i, \vec{q}_f) &= (-\vec{\nabla}_{p_i} F_4, \vec{\nabla}_{p_f} F_4).
\end{aligned}
$$

Here $F_1(\boldsymbol{q}_i, \boldsymbol{q}_f), F_2(\boldsymbol{q}_i, \boldsymbol{p}_f), F_3(\boldsymbol{p}_i, \boldsymbol{q}_f), F_4\,(\boldsymbol{p}_i, \boldsymbol{p}_f)$ are the four most commonly used **generating functions** (GF) in mixed variables.

The generating function and the respective PDEs can be constructed from a truncated map $\boldsymbol{M}$ of the dynamic system using **differential algebra** (DA).

Methodology[1,2]

Symplectic Tracking:

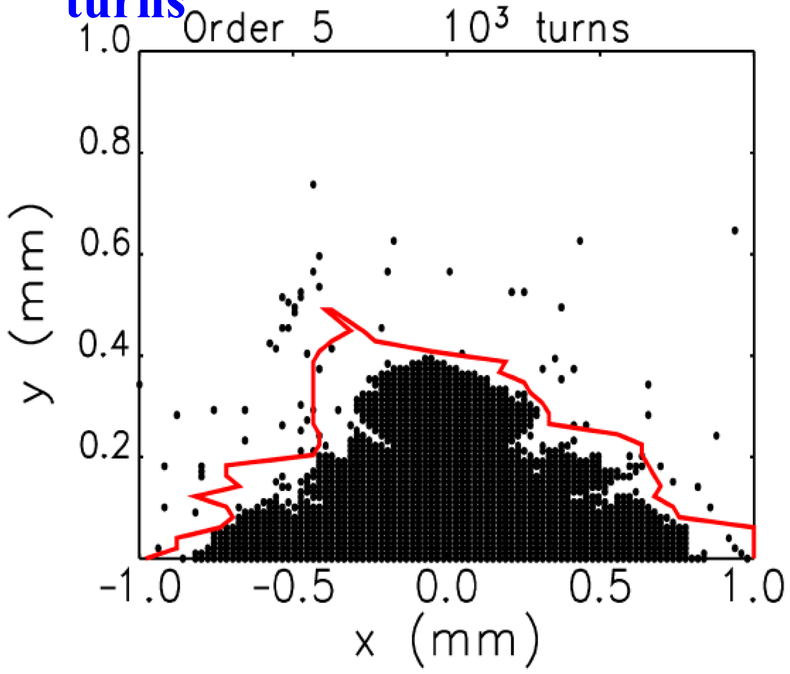
1. Use the truncated map to construct the GF
2. Construct the PDEs by taking derivatives of the GF.
3. Calculate $\boldsymbol{X}_f = \boldsymbol{M} \circ \boldsymbol{X}_i$, where $\boldsymbol{M}$ is the truncated map, $\boldsymbol{X}_i$ is the initial coordinates.
4. Use $\boldsymbol{X}_i$ and $\boldsymbol{X}_f$ as initial guess to solve the PDEs iteratively. Since $\boldsymbol{M}$ is very close to the real symplectic map, the solution should converge very fast in just a few iterations.

[1] Modern Map Methods in Particle Beam Physics, page 293
M. Berz, Academic Press, 1999, ISBN 0-12-014750-5
[2] Symplectic Tracking in Circular Accelerators with High Order Maps
M. Berz, in: *"Nonlinear Problems in Future Particle Accelerators"* (1991) 288-296, World Scientific

Long term tracking checks (milestone 2)

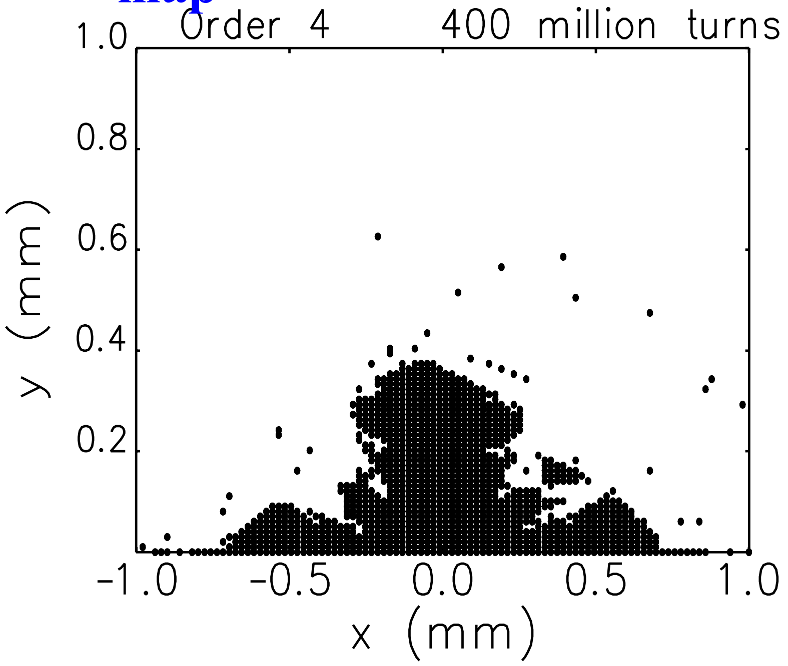
- Dynamic aperture comparison to Elegant (Borland 2000)
- 400 million turn simulation (truly long-term)

GHOST **Elegant** **1,000 turns**



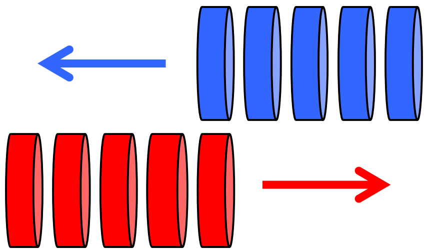
Excellent agreement!

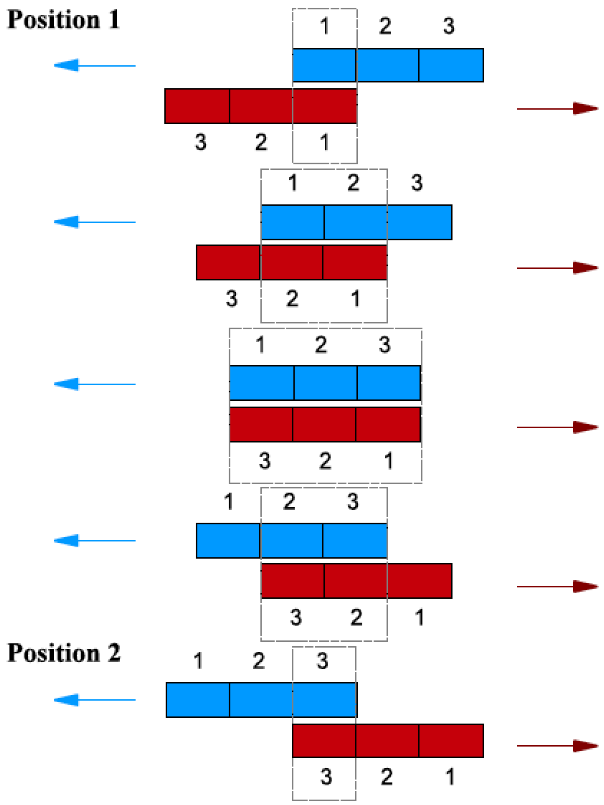
Sympletic Tracking 4th order map



GHOST: Beam Collisions (milestone 4)

- Bassetti-Erskine approximation
  - Beams as 2D transverse Gaussian slices
  - Poisson equation reduces to a complex error function
  - Finite length of beams simulated by using multiple slices



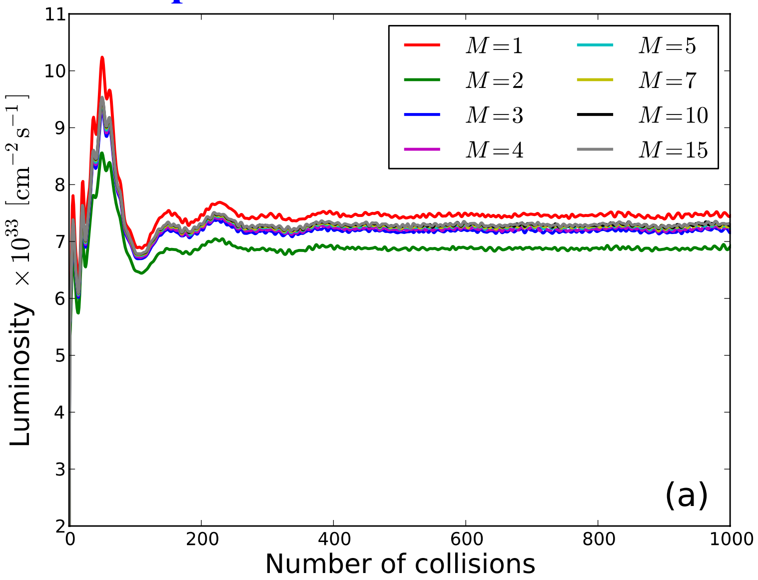


•We generalized a “weak-strong” formalism of Bassetti-Erskine
- Include “strong-strong” collisions (each beam evolves)
- Include various beam shapes (originally only flat beams)

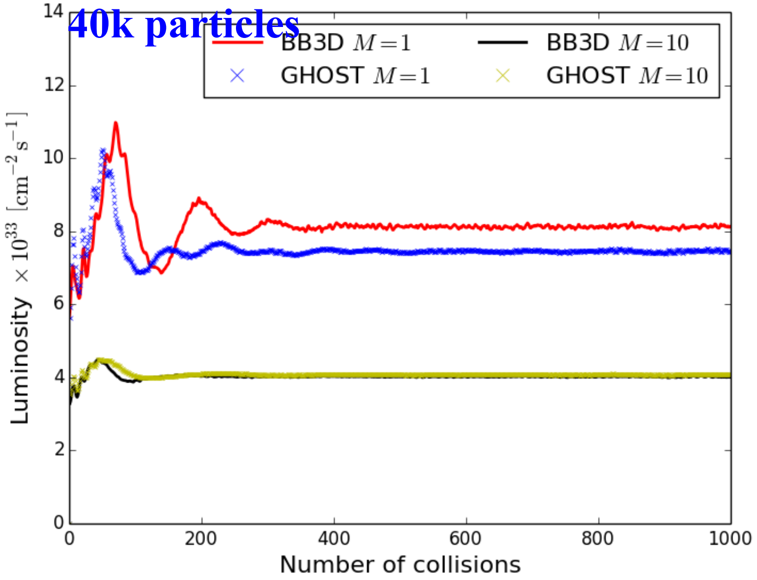
GHOST: Beam Collisions (milestones 4 and 5)

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

GHOST, 1 cm bunch
40k particles



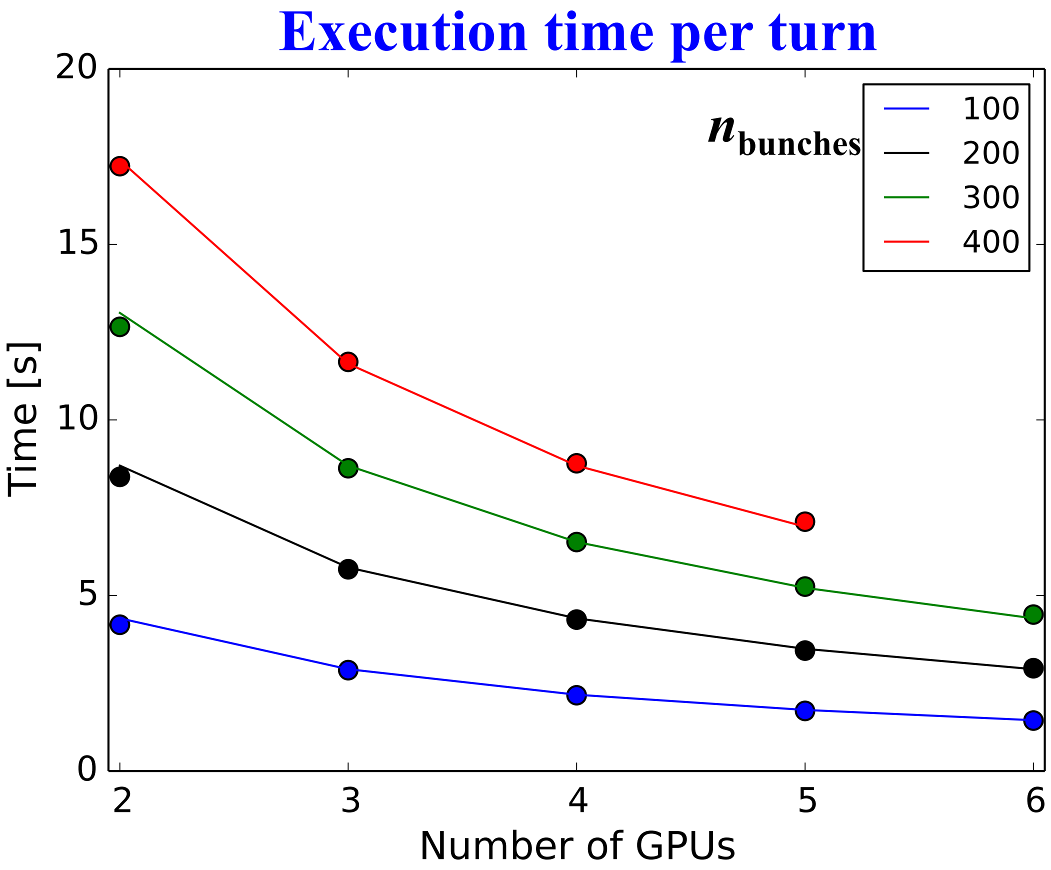
BeamBeam3D & GHOST, 10 cm bunch
40k particles



Finite bunch length accurately represented

Excellent agreement with BeamBeam3D

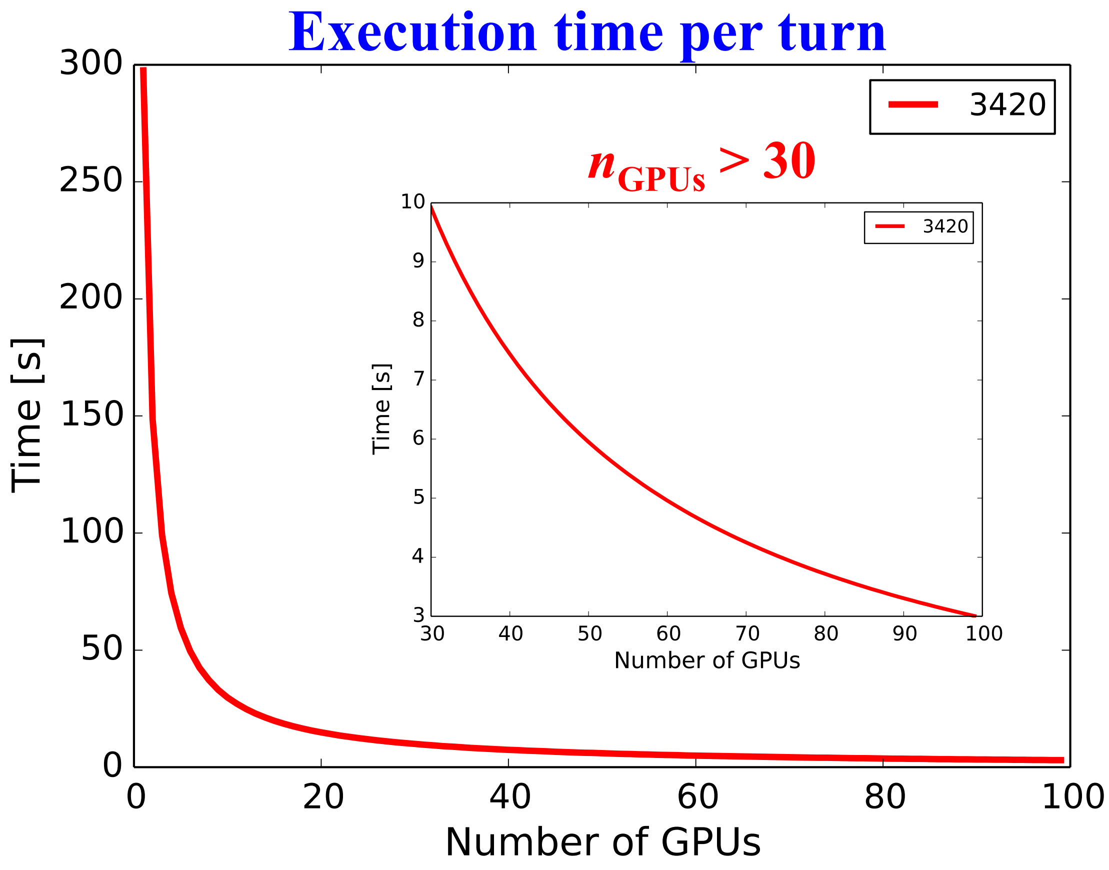
"Gear Change" with GHOST on multiple GPUS (milestone 6)



One turn on K40 GPUs: $t = 0.087 \frac{n_{\text{bunches}}}{n_{\text{GPUs}}} \text{ sec}$

Execution time scales as $1/n_{\text{GPUs}}$

“Gear Change” with GHOST: *n* x (*n*-1)

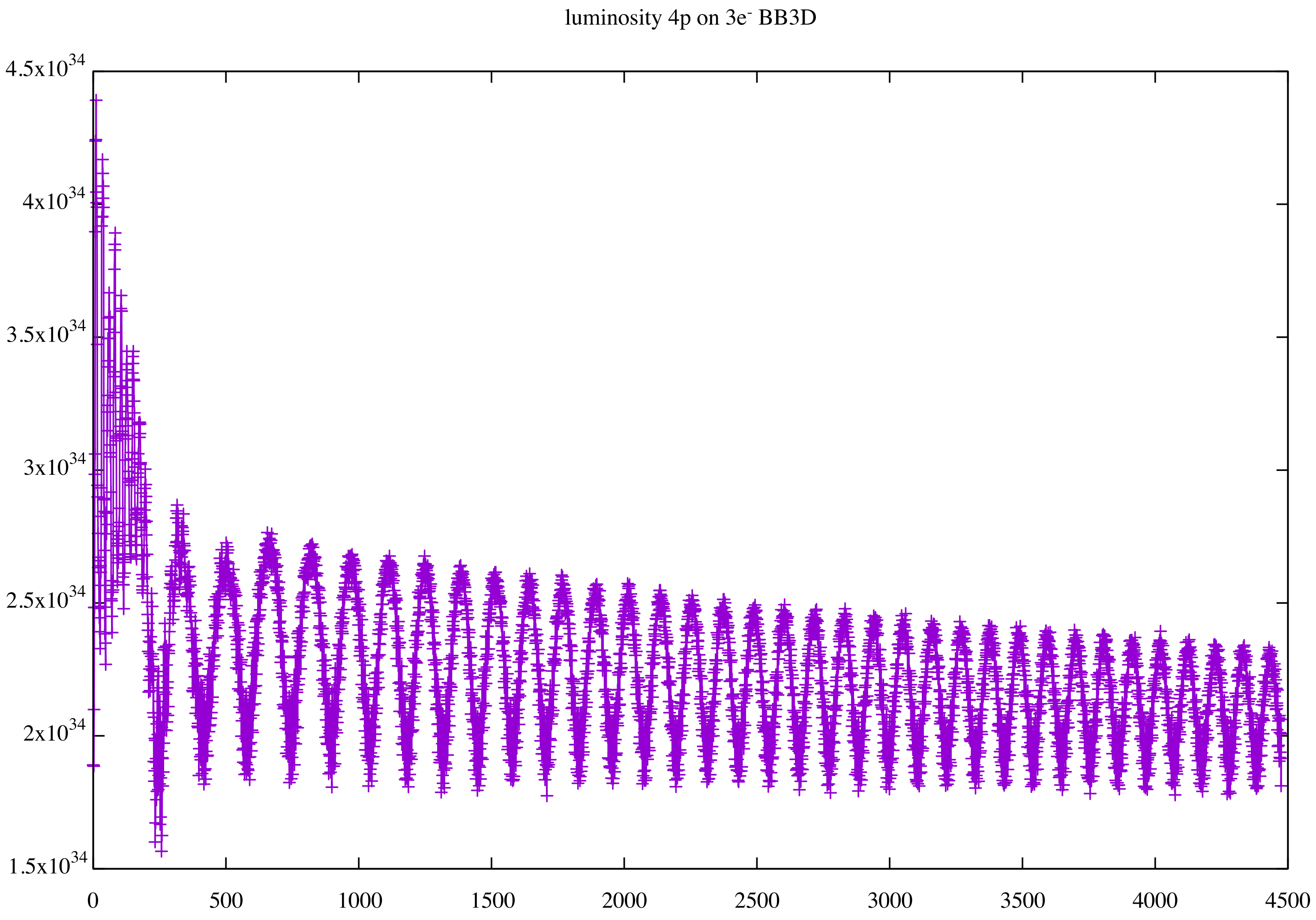


One turn on K40 GPUs: $t = 0.087 \frac{n_{\text{bunches}}}{n_{\text{GPUs}}}$ sec

Gear-changing simulations and benchmarking (milestone 8)

- Configured BB3D to perform simulations for a small number of bunches
- Compared with GHOST for various scenarios up to 7x6
- No attempt to optimize for good luminosity

Gear changing simulations 4x3 luminosity, BB3D



Gear changing simulations (GHOST)

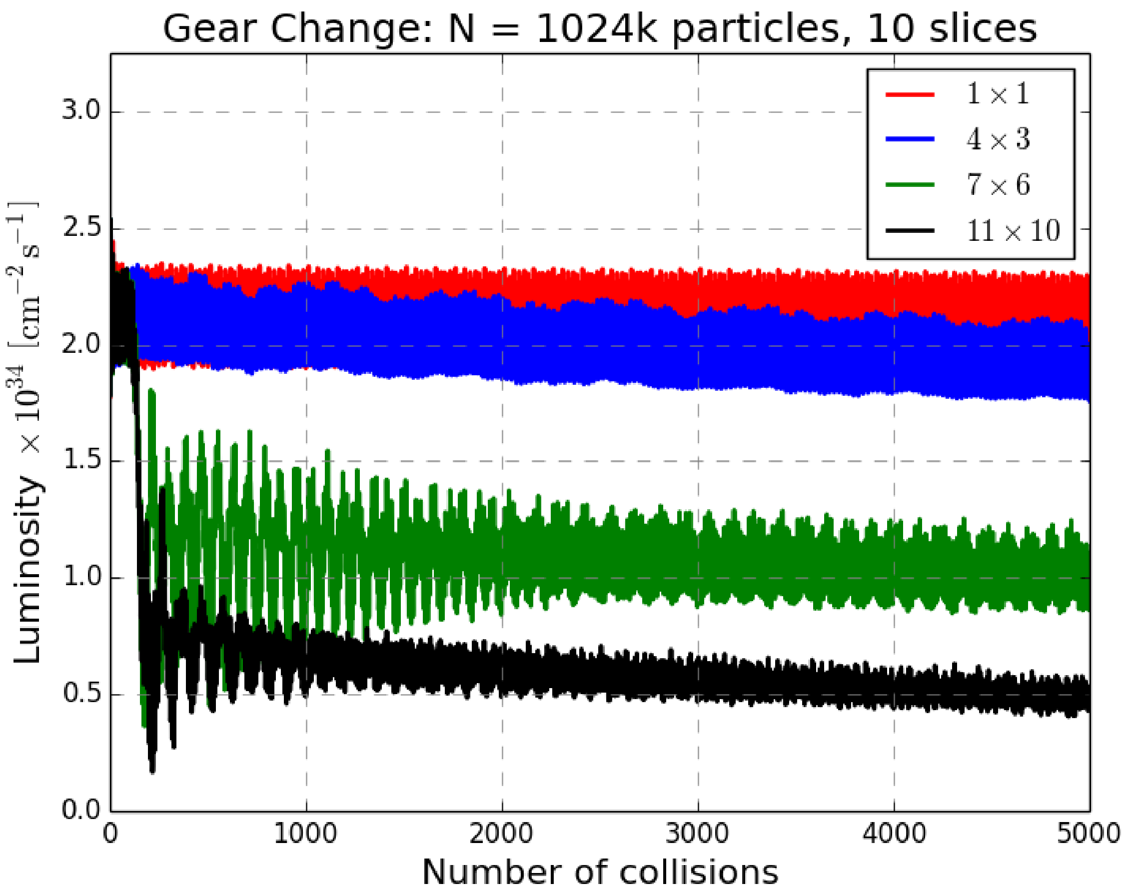
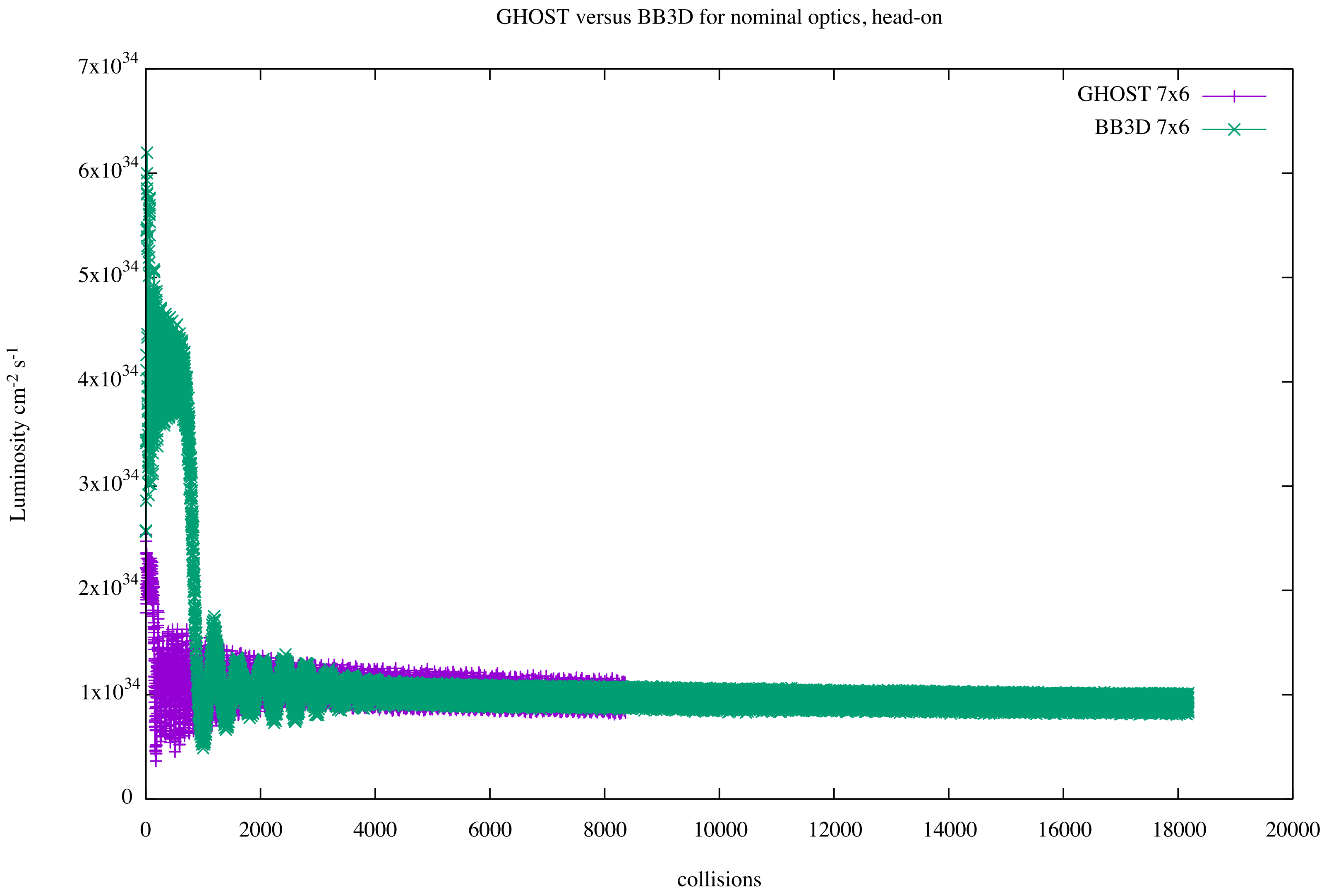
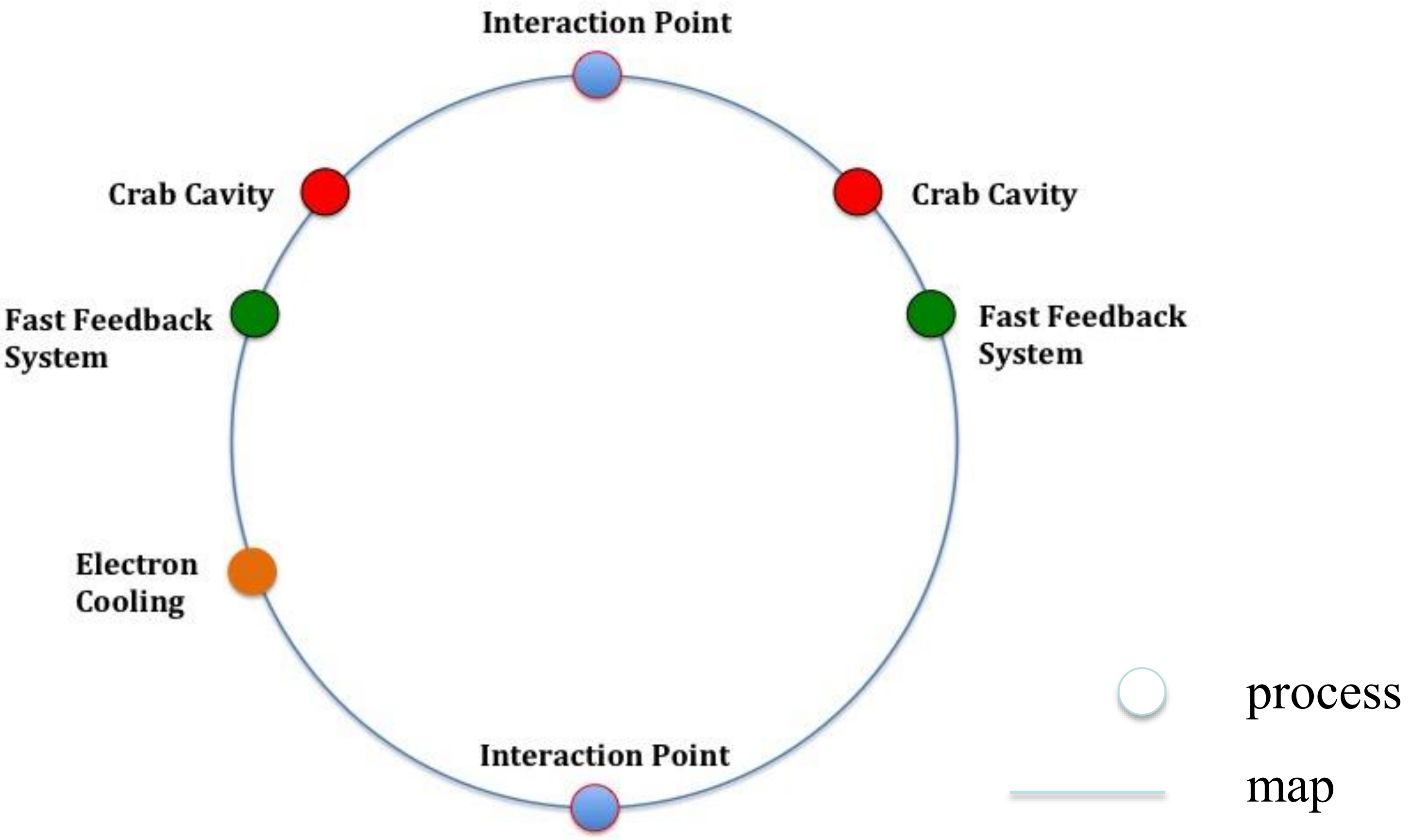


Figure 8:Initial tests of gear changing with GHOST

Gear changing 7x6 BB3D versus GHOST (10 slices)



Current/Future Efforts: Other Functionalities



GHOST's modular nature allows for flexibility in simulations