BE-ABP: Beam Dynamics on the GPU

SixTrackLib + PyHEADTAIL:

Summary of Day 3

Riccardo de Maria, Lotta Mether, Adrian Oeftiger, Martin Schwinzerl

CERN – Openlab E4 Hackathon

15-17 April 2019

Goal

Starting point:

- **two existing codes:**
  SixTrackLib (templated C) and PyHEADTAIL (Python + PyCUDA)

⟶ **merge functionality:** single-particle tracking + multi-particle dynamics

- abstracted PyHEADTAIL in CuPy: jupyter notebook ↗ on github ↗

Goals:

- extend this python script based on PyHEADTAIL to

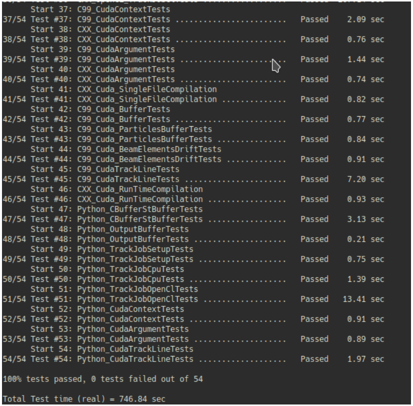
works!: prepare the accelerator optics in SixTrackLib in chunks (not SixTrackLib's usual one-turn behaviour)

works!: share the macro-particle coordinates / memory from PyHEADTAIL with SixTrackLib

works!: call to SixTrackLib to track through chunks of the optics lattice before returning to a PyHEADTAIL multi-particle interaction module

⟹ everything runs in CuPy!

What Did we Do?



```
      Start 37: C99_CudaContextTests
37/54 Test #37: C99_CudaContextTests .........................   Passed    2.09 sec
      Start 38: CXX_CudaContextTests
38/54 Test #38: CXX_CudaContextTests .........................   Passed    0.76 sec
      Start 39: C99_CudaArgumentTests
39/54 Test #39: C99_CudaArgumentTests ........................   Passed    1.44 sec
      Start 40: CXX_CudaArgumentTests
40/54 Test #40: CXX_CudaArgumentTests ........................   Passed    0.74 sec
      Start 41: CXX_Cuda_SingleFileCompilation
41/54 Test #41: CXX_Cuda_SingleFileCompilation ...............   Passed    0.82 sec
      Start 42: C99_Cuda_BufferTests
42/54 Test #42: C99_Cuda_BufferTests .........................   Passed    0.77 sec
      Start 43: C99_Cuda_ParticlesBufferTests
43/54 Test #43: C99_Cuda_ParticlesBufferTests ................   Passed    0.84 sec
      Start 44: C99_Cuda_BeamElementsDriftTests
44/54 Test #44: C99_Cuda_BeamElementsDriftTests ..............   Passed    0.91 sec
      Start 45: C99_CudaTrackLineTests
45/54 Test #45: C99_CudaTrackLineTests .......................   Passed    7.20 sec
      Start 46: CXX_Cuda_RunTimeCompilation
46/54 Test #46: CXX_Cuda_RunTimeCompilation ..................   Passed    0.93 sec
      Start 47: Python_CBufferStBufferTests
47/54 Test #47: Python_CBufferStBufferTests ..................   Passed    3.13 sec
      Start 48: Python_OutputBufferTests
48/54 Test #48: Python_OutputBufferTests .....................   Passed    0.21 sec
      Start 49: Python_TrackJobSetupTests
49/54 Test #49: Python_TrackJobSetupTests ....................   Passed    0.75 sec
      Start 50: Python_TrackJobCpuTests
50/54 Test #50: Python_TrackJobCpuTests ......................   Passed    1.39 sec
      Start 51: Python_TrackJobOpenClTests
51/54 Test #51: Python_TrackJobOpenClTests ...................   Passed   13.41 sec
      Start 52: Python_CudaContextTests
52/54 Test #52: Python_CudaContextTests ......................   Passed    0.91 sec
      Start 53: Python_CudaArgumentTests
53/54 Test #53: Python_CudaArgumentTests .....................   Passed    0.89 sec
      Start 54: Python_CudaTrackLineTests
54/54 Test #54: Python_CudaTrackLineTests ....................   Passed    1.97 sec

100% tests passed, 0 tests failed out of 54

Total Test time (real) = 746.84 sec
```

Figure: UnitTests for SixTrackLib chunk tracking

- unit tests for SixTrackLib implemented
- connected both codes with low level memory management in CuPy (not the way it is supposed to work but `cupy.asarray` did not work out – waiting for better solution from cupy team!)
- tried to connect with `nsight` to HPC node to assess profiling but did not succeed yet...
- talked with Miguel how to incorporate `cuDF` with `CuPy` to simply extend to multi-GPU! (remember FCC 100km study with 300GB memory needs)

How does it look like?

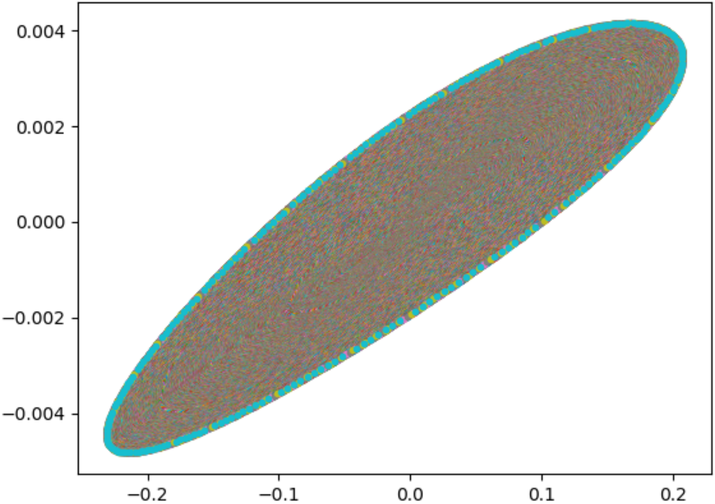
PyHEADTAIL

```
(... setting up SixTrackLib handlers ...)

beam = make_PyHEADTAIL_beam(...)

one_turn_map = [
    sixtrackelem1,
    pyheadtail_resonator,
    sixtrackelem2
]

with GPU(beam):
    for turn in range(n_turns):
        for m in one_turn_map:
            m.track(beam)
```

see https://nbviewer.jupyter.org/github/aoeftiger/PyHEADTAIL_concept_testing/blob/develop/Merging.ipynb

Example

5000 particles, 256 turns of full SPS lattice in SixTrackLib!


0.004
0.002
0.000
−0.002
−0.004
−0.2
−0.1
0.0
0.1
0.2