

Request !31

Track iteration cleanup

- Focused on the *TrackIteration* object and unit tests
- No functional changes, focused on type hinting and PEP8 compatibility
- Not backwards compatible, but also not used by any “operational” code (Kostis started using it for the assembler)

Request !32 - Overview

Turn counter object

- Adds a new object *TurnCounter*
- The intention is to create an independent counter that defines the current turn and section
- Objects that require turn/section information would typically refer to the default counter, for advanced cases multiple counters can be created

Request !32 - Motivation

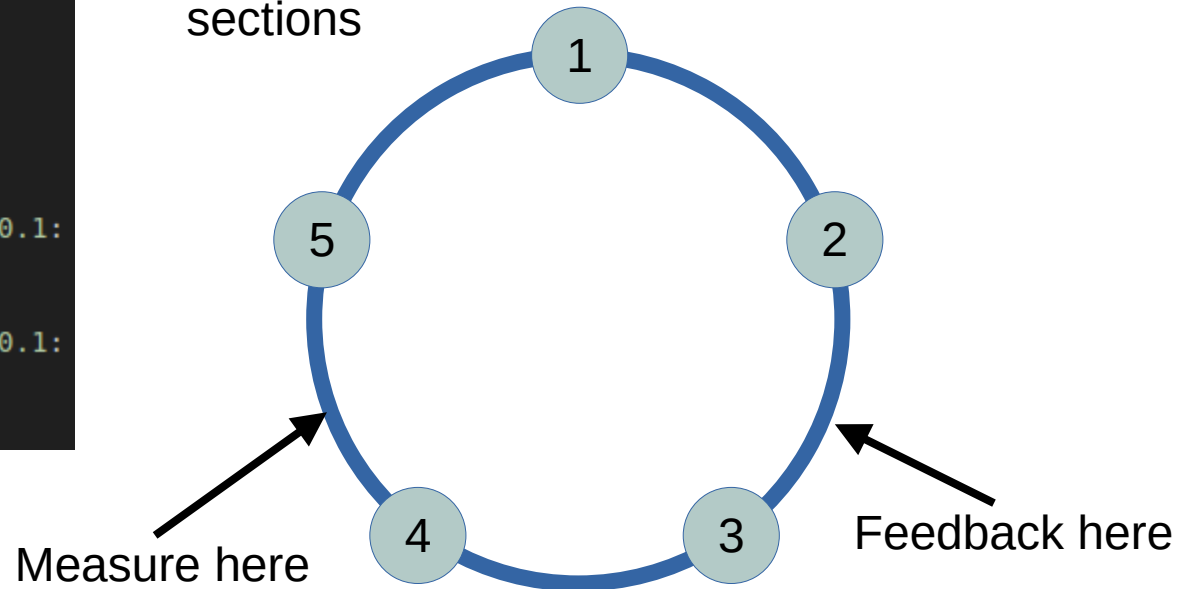
```
map1 = [full_ring1, profile1, multi_inj1]
map2 = [full_ring2, profile2, multi_inj2]

for i in range(n_inj1):
    for m in map1:
        m.track()

for i in range(n_inj2):
    for m in map2:
        m.track()

for i in range(1000):
    if ring1.cycle_time[counter1.current_turn] < 0.1:
        for m in map1:
            m.track()
    if ring2.cycle_time[counter2.current_turn] < 0.1:
        for m in map2:
            m.track()
```

- When tracking in stages, keeping track of the “current” turn number can be difficult
- At present, the *FullRingAndRF.track()* method automatically tracks through all sections



Request !38

Akima interpolation

- Adds the interpolation mode 'akima' to *RingOptions*, no other functional change
- Enables interpolation with **Akima splines**, a form of piecewise polynomial, defined in **SciPy**
- Structural change: Interpolation methods moved to private functions of *RingOptions* to improve readability of *RingOptions.preprocess*

Request !41

Multi turn injection

- New trackable object *MultiTurnInjection* that holds *Beam* objects to be added to the simulation, adds them on the correct turns
- Used for multi turn injection, might be worth adapting to allow batch injections e.g. on SPS/LHC long flat bottom
- Performance improvement to *Beam.add_beam*
- Adds type hints to *Beam* and makes *Beam.ratio* and *Beam.intensity* into mutually updating properties

Request !42

RF waveform in RFStation

- Adds *RFStation.compute_voltage_waveform* method
- Simple convenience function to calculate the waveform provided by a given RFStation at specified turn and return an array of [time, amplitude]

Request !47 - Overview

GPU `simple_kick` fix

- Imperfect fix to critical bug that arises with multiple RF systems with GPU
- Array slices are passed as a reference to the first element of the slice, not a subset of the array
- Indexing in CUDA kernel does not match indexing in Python

Request !47 - Details

2D array as defined

1	1	2	2	5
1	4	4	4	4

Slice requested in Python

2	4
---	---

slice[0]

slice[1]

2D array in memory

1	1	2	2	5	1	4	4	4	4
---	---	---	---	---	---	---	---	---	---

slice[0]

CUDA: slice[0]+1

Python: slice[1]