Swiss Accelerator
Research and
Technology

Short update on python framework activities

Félix Carlier

Table of contents

1. Overview of functionalities
2. Lattice manager
3. Next steps

Progression of functionalities of framework

1. Successfully import and export of lattices from/to MAD-X, pyAT, SAD, Bmad, ..
2. Include calculations of orbit and optical functions. Interface → Engine independent
3. Include tracking simulations with MAD-X, pyAT, SAD, Sixtracklib, ...
4. Further developments.

Thorough testing required after each step!

Progression of functionalities of framework

1. **Successfully import and export of lattices from/to MAD-X, pyAT, SAD, Bmad, ..**
2. Include calculations of orbit and optical functions. Interface → Engine independent
3. Include tracking simulations with MAD-X, pyAT, SAD, Sixtracklib, ...
4. Further developments.

Thorough testing required after each step!

Solid lattice description crucial for codes

Ensuring stability of lattice between different codes:

- ensures consistency when moving between codes.
- validates comparative simulation studies of different codes.
- allows alternative lattice exploration: thin/thick, alternative slicing, ..

Test and validate stability of lattice after multiple engine changes

→ Is lattice unchanged after: MAD-X → SAD → pyAT → … → MAD-X

Current developments already highlight different conventions in modelling of accelerators:

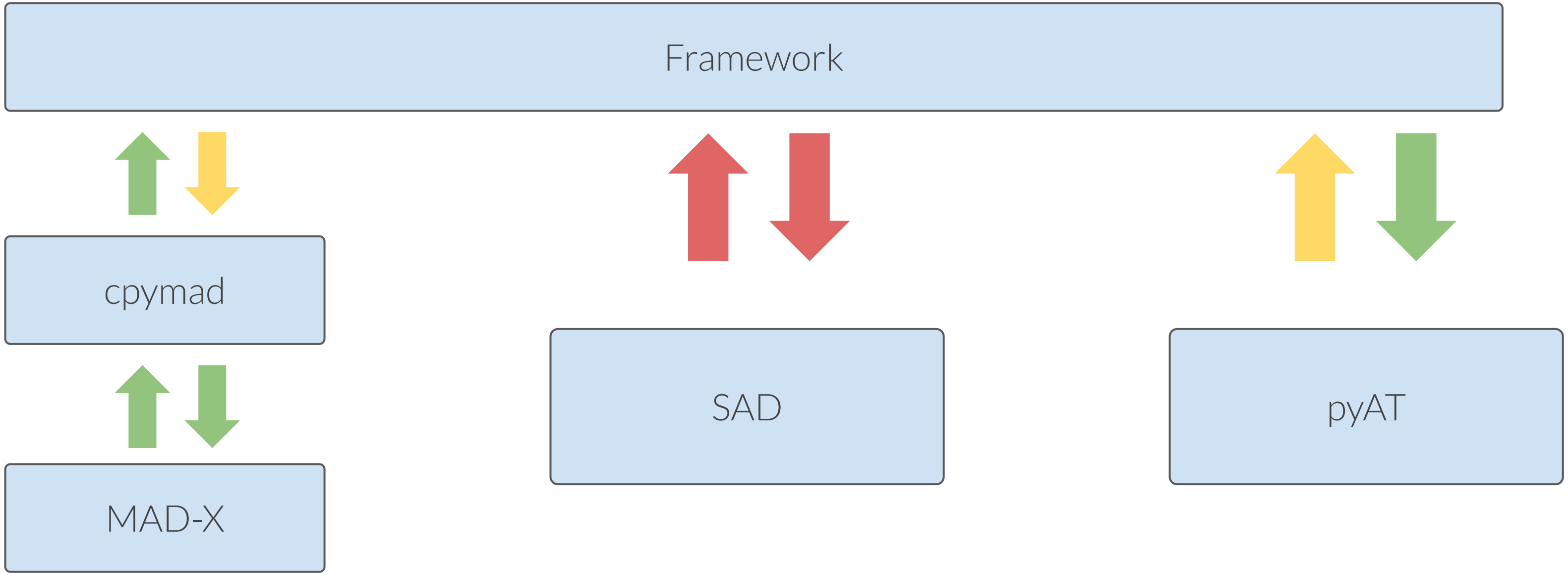
- Dipole vs. rbend/sbend
- Sequence vs. line representation
- Magnetic coefficients to arbitrary order, vs. limited representation
- RF cavities with different frequencies not accepted everywhere
- …

Import and export of lattices between codes

Only for FCC-ee lattices for the moment:

- Very basic and minimal element representations, but sufficient for FCC-ee.

Can also directly define elements and lattice in Python



FCC-ee lattice import from cpymad

Currently:

- Successfully imported lattices of FCC-ee to framework.
- Directly calculates sequence and line representations (implicit drifts)
    - Currently with negative drift tolerance of 1e-9 m
- Successfully exported lattices to cpymad (some details remain, next slide...)

Needs to be extended further:

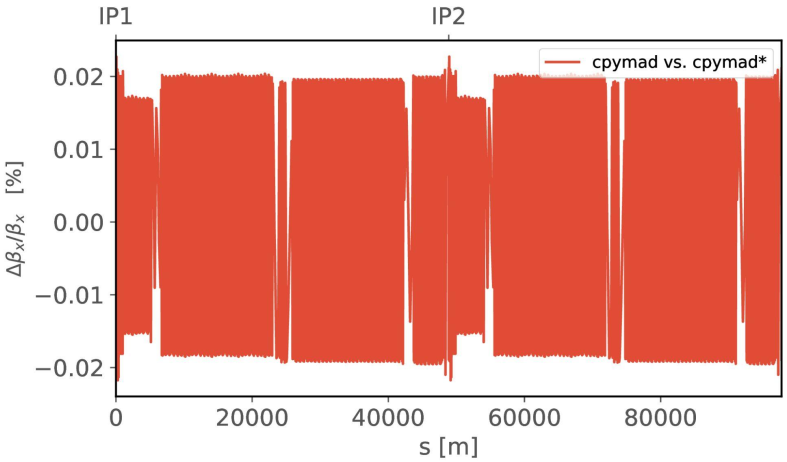
- More complicated lattices need to be tested. FCC-ee lattice does not have wide range of element types.
    - No collimators, octupoles, solenoid, beam-beam, kickers, …
- More complicated element parameters need to be tested:
    - Errors, rotations, apertures, ...

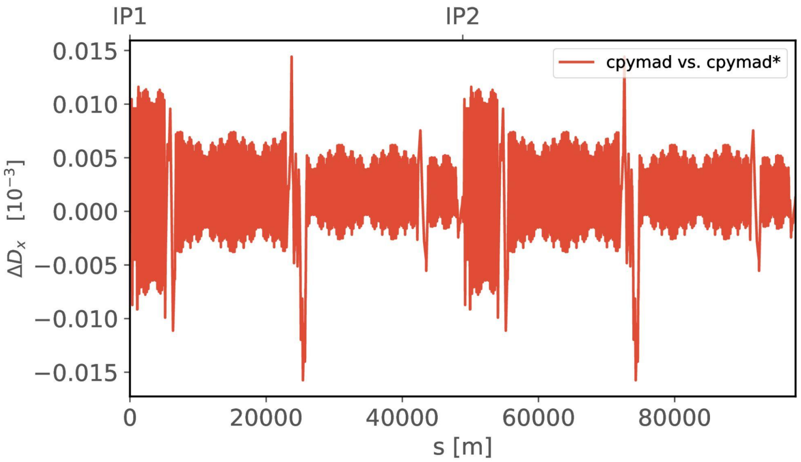
Optics not yet perfectly reproduced!

Cpymad → Framework → cpymad

Discrepancy under investigation!

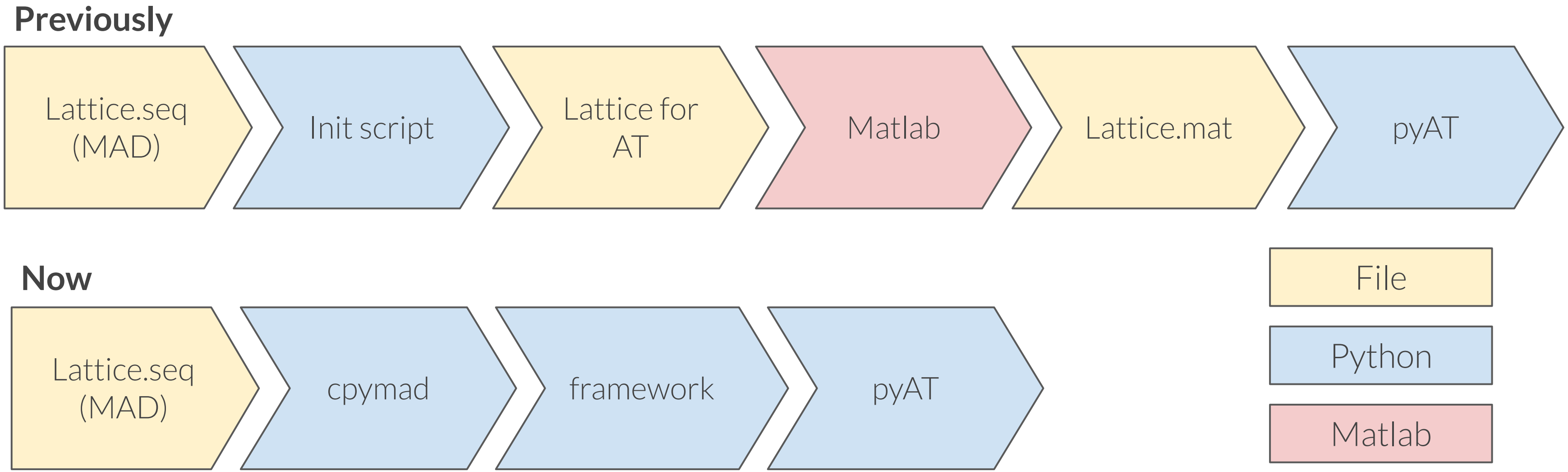
- Possible rounding errors? Drifts around sbend/rbend ?





Status of pyAT conversions

Lattice translation through framework now bypasses Matlab version of Accelerator Toolbox, and multiple file creations.



Inverse direction (pyAT → Framework) is currently being tested

Opening port to SAD

Access to SAD is currently not available and will be explored next.

Start by exploring current lattice conversion scripts between SAD and MAD-X.

Will probably have to rely on file creations and automatic script creations using templates for now.

Opening port to SAD

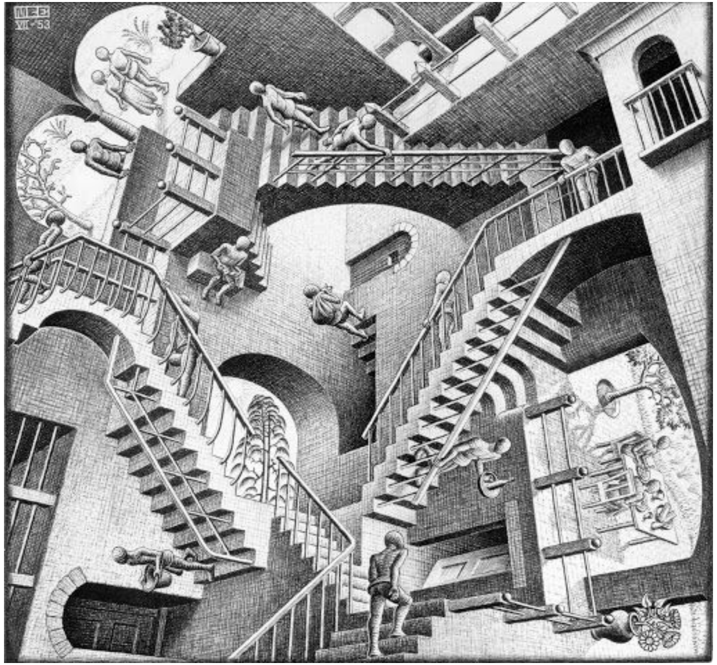
Access to SAD is currently not available and will be explored next.

Start by exploring current lattice conversion scripts between SAD and MAD-X.

Will probably have to rely on file creations and automatic script creations using templates for now.

But still need to learn a lot about SAD!

Current understanding →



Next steps

- **Develop tests** for lattice import/export and elements
  → Lattice should be unchanged after: MAD-X → SAD → pyAT → … → MAD-X
  → Are elements unchanged after: MAD-X → Framework → MAD-X ?

- Explore how to include SAD to framework

- Perform comparative simulations between codes: orbits, optics
  - w/ and wo/ radiation and tapering.

- Create basic simulation functions with engine independent interface
  - i.e. lattice.optics(engine = ‘madx’)