# ICFA Mini-Workshop on Slow Extraction, 2022

*KEK/J-PARC  
24-28 January 2022*

Accelerator codes for SX

For reference: <https://en.wikipedia.org/wiki/Accelerator_physics_codes>

To contact Kevin Brown, BNL: brownk@bnl.gov

*Instructions:*

*Consider this a fluid document – add and remove as you like. It is meant to be community built. Add your names/contact info if you don’t mind.*

**Contact info**

Cristopher Cortés, [Cristopher.CortesGarcia@med.uni-heidelberg.de](mailto:Cristopher.CortesGarcia@med.uni-heidelberg.de), HIT (Heidelberg)

*This initial word doc is not the best format. I will try to get it into latex and into overleaf, or if someone wants to volunteer to put into a better online collaborative format, they are welcome.*

## What do people use for SX and related simulations? (if your institute isn’t shown, please add it)

**BNL**

Madx

Zgoubi

Vsim

CST studio

Opera3D

BLonD

Bmad

*Local (provide website or technote, etc. for ref.)*

Raytrace – tracking with field maps, satisfies Maxwells Equations

Tltrack – pseudo element to element symplectic tracking

<https://technotes.bnl.gov/PDF?publicationId=32127>

<https://technotes.bnl.gov/PDF?publicationId=221761>

**CERN**

**CNAO**

**FNAL**

**GSI**

**HIT (Heidelberg)**

MAD-X

**IHEP**

**INFN**

**KEK/JPARC**

**Korea Univ.**

**Kyoto, IIRNS**

**Max Plank Inst.**

**MedAustron**

**MIT**

**National Institutes for Quantum Sciences …**

**Petersburg Nucl Phy Inst.**

**RAL**

**SEEIIST Assoc.**

→ see CERN-NIMMS

## Features needed for SX

1. Fast tracking (as many as millions of turns for thousands or particles)
2. Transit time tracking (what exists?)
3. Spill simulation (e.g., including vacuum chamber attenuation, instrument response, etc.)  
   1. What features do people want?
4. More realistic beam loss tracking (tracking particles after they hit something)  
   1. GEANT, MARS (CERN Yellow Report), FLUKA
   2. BDSim – GEANT + beamline tracking
   3. G4Beamline (commercial)
5. Integration between field simulations and tracking (tracking through ESS, at Magnetic Septa, etc)
6. Collimation and beam shaping (uniform beams)
7. Sophisticated raster scanning with scattering and shaping
8. Space charge (particularly for lower energy beams)
9. Fast collective effects (during debunching, electron cloud buildup, etc.)
10. Beam loading response and compensation during SX
11. Integrating crystal channeling with more global tracking code
12. Integrating with simulations of instruments (IPM, beam screen, etc.)
13. Regulation process
14. SAD – 6D description
15. Improved documentation
16. Community: bests practices and tricks of the trade  
    1. Python – track in c.
    2. Cython
    3. pyorbit (c and c++ libs) – Oak Ridge => CERN
    4. cpymad

## Software for Operations

OpenAiGym, BobyQA, COSE

OpenCMW – Open Common Middle-Ware library

GSI public GitHub coordinates are: <https://github.com/fair-acc/>

and for OpenCMW in particular: <https://github.com/fair-acc/opencmw-java>  
<https://github.com/fair-acc/opencmw-cpp>

**User Interface/Code Integration**

CERN-NIMMS

python notebook = sloexlab

Works with madx and maptrack

pyNAFF for fast Fourier transform

**General (mostly transverse) codes**

Madx: <https://madx.web.cern.ch/madx/>

(does anyone use the one-turn mapping translation? Needs to convert to thin lens.)

python binding of MAD-X called cpymad:<https://github.com/hibtc/cpymad>

Maptrack or MADX/PTC for the tracking: <https://gitlab.cern.ch/abt-optics-and-code-repository/simulation-codes/maptrack>

SixTrack (??) 6D tracking: <http://sixtrack.web.cern.ch/SixTrack/>

SAD (mostly KEK/JPARC ?): <https://acc-physics.kek.jp/SAD/>

Zgoubi (Francois Meot): <https://sourceforge.net/projects/zgoubi/files/>

mad-ng (under dev, any users yet?)

Elegant (pelegant): <https://www.aps.anl.gov/Accelerator-Operations-Physics/Software>

BNL local codes

Tltrack – c++ pseudo element by element

Raytrace (Nick Tsoupas)

(noname) Hamiltonian tracker in python (SageMath)

Bmad (growing, but not easy to setup and get started): <https://www.classe.cornell.edu/bmad/>

**For space charge**

pyOrbit: <https://github.com/PyORBIT-Collaboration>

Synergia2: <https://bitbucket.org/fnalacceleratormodeling/synergia2/src/devel3/>

Impact (anyone?)

Bmad

**Longitudinal**

SixTrack

BLonD – longitudinal simulations: [https://blond.web.cern.ch](https://blond.web.cern.ch/)