CERN

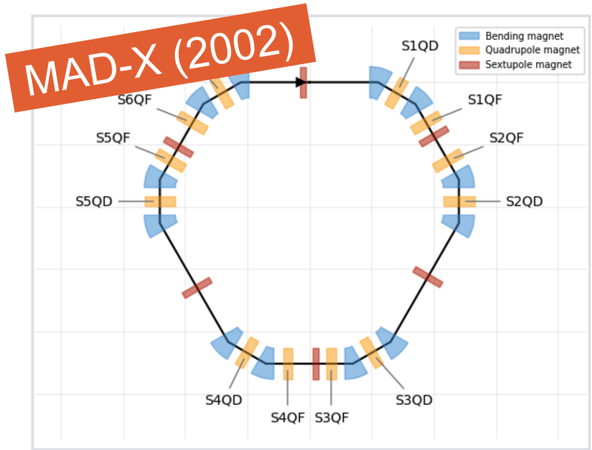
Beam Physics/Simulation Software Evolution: Challenges, Opportunities

Szymon Łopaciuk, Giovanni Iadarola, Riccardo De Maria, CERN, Geneva, Switzerland

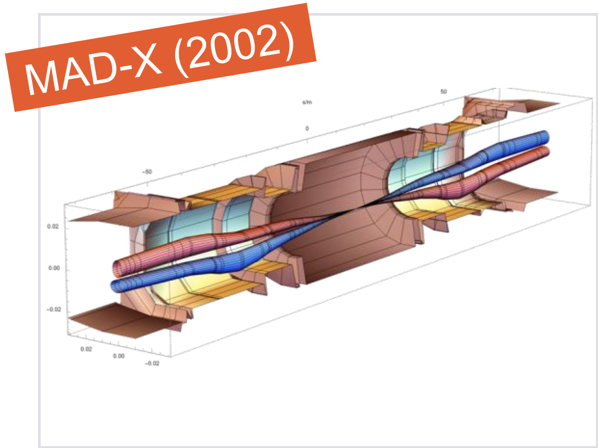
We sincerely thank the Xsuite users and contributors for their enthusiasm and invaluable input. Special thanks to Yannis Papaphilippou and the ABP management for their guidance and for supporting these developments.

What is ‘software for beam dynamics’?

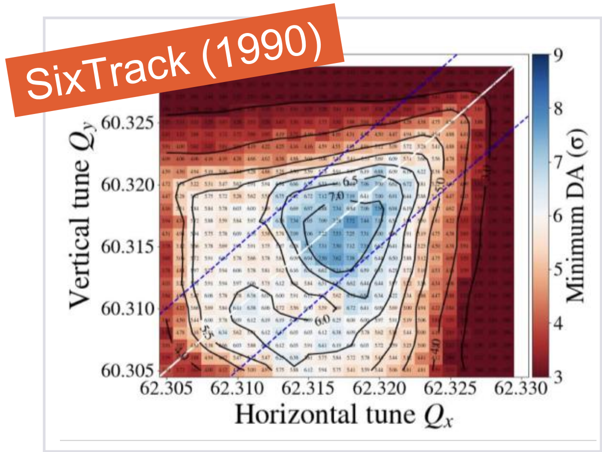
CERN has a long tradition of powerful software tools for beam physics applications, typical examples (ABP):



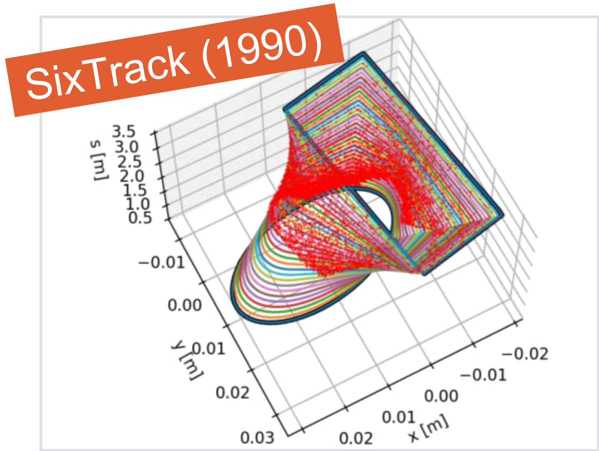
Lattice Design



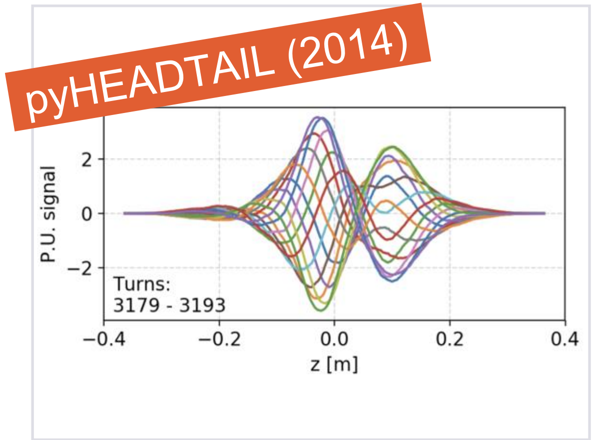
Optics (calculation & design)



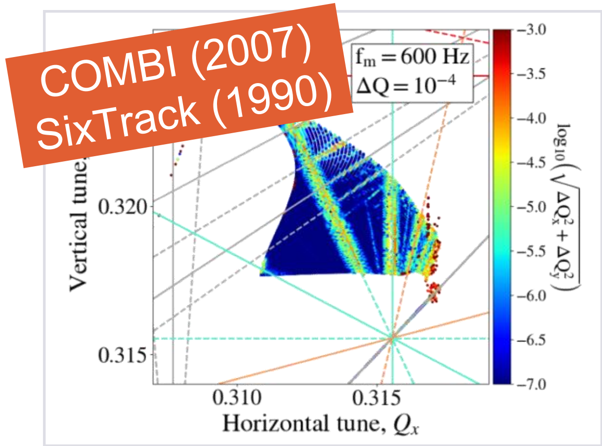
Tracking (dynamic aperture)



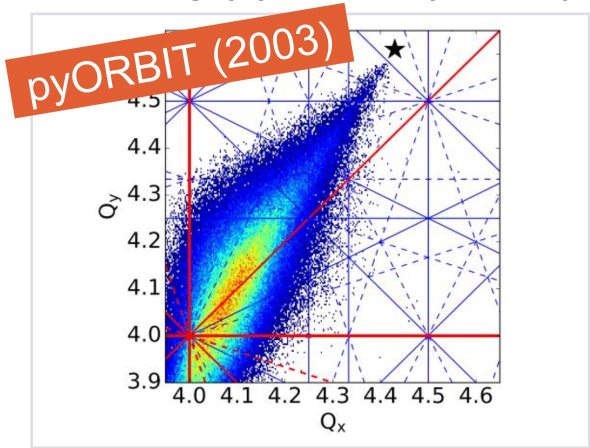
Collimation



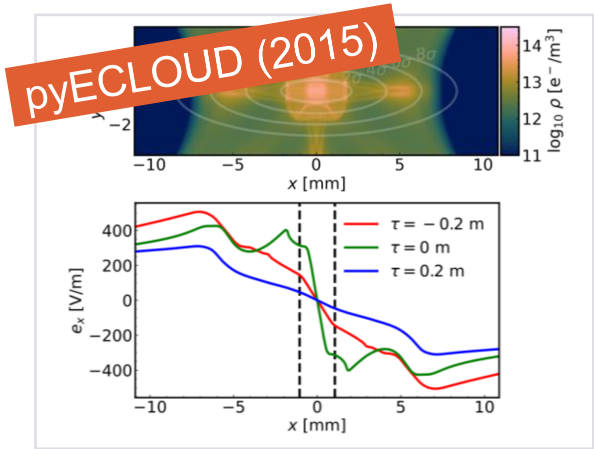
Impedances



Beam-Beam Effects



Space Charge

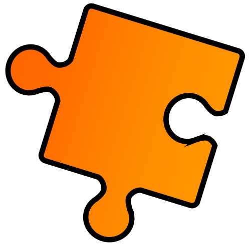


Electron/Ion Cloud

Need to do better

This configuration with legacy tools has several limitations:

- **Integration issues**
  - Present studies often require heterogeneous studies with interplay of optics, non-linear tracking, collective effects, collimation, etc.
  - With legacy tools this means dealing with model translations, ad-hoc interfaces between codes, which is complicated, time-consuming, error prone, and slow.
- **Extendibility**
  - Very hard to extend legacy tools to present needs (notably FCC).
  - Many experts have left or retired; working on old Fortran code often difficult and not really appealing for young colleagues.
- **Outdated user interfaces – today users want Python**
  - Python is a standard in scientific computing, especially among younger researchers.
  - Many high-quality tools are available in the Python ecosystem also due to ML boom. Maintaining custom tools for plotting, scripting, etc. wastes effort, increases complexity, and lowers quality.
- **Performance**
  - GPU acceleration is a must-have in many applications, but often infeasible to retrofit.



Xsuite project launched in 2021 to address these issues using the know-how acquired in the development of the earlier tools.

Goals: 1st class Python, 1st class collective, 1st class GPU.

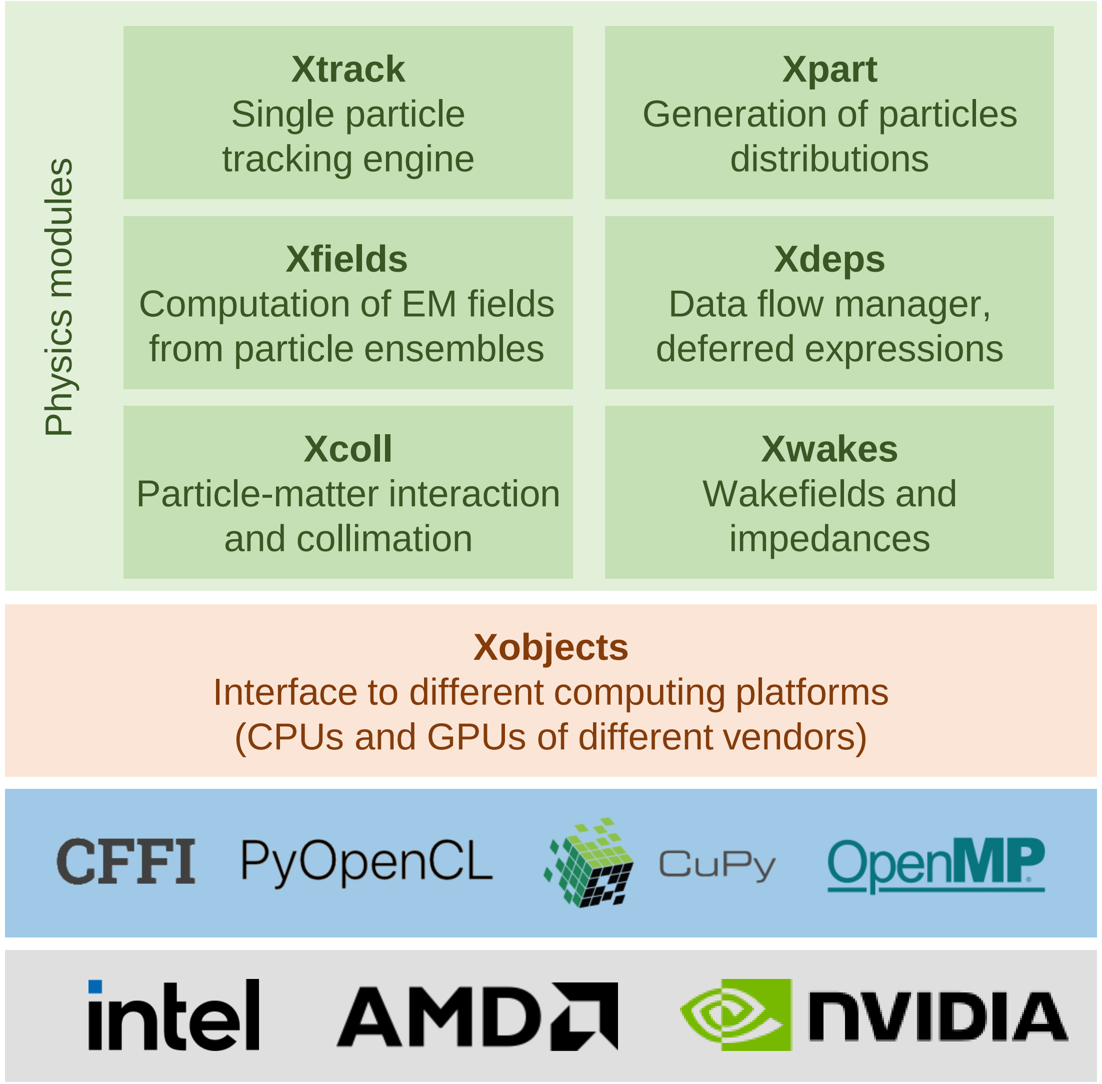
- One toolkit for many applications: from low-energy hadron rings to high-energy lepton colliders.
- Heterogeneous simulations made natural.
- Support different computing platforms – single- and multi-threaded CPU, GPU acceleration.

References for full list of features:

- *Documentation: xsuite.web.cern.ch.*
- *"Xsuite: An Integrated Beam Physics Simulation Framework," JACoW HB2023 (2024), TUA2I1.*
- *"Xsuite: a multiplatform toolbox for optics design, fast tracking, collimation and collective effects," ICAP'24.*

Xsuite approach

- **Orthogonal architecture:** split the software into independent functional blocks at all levels.
  - This means well-defined and cleaner interfaces, which lower overall complexity of the codebase.
  - Simpler codebase scales better and it is easier to add new features.
  - Lower learning curve for users & developers ⇒ users can (and do!) become developers.
- **Agile development:**
  - From the beginning big effort to support users: user feedback and involvement visibly increases the quality of the package.
  - Thanks to investment in continuous testing, we can afford a fast release cycle with new versions coming out multiple times a month, incorporating modifications/extensions based on user feedback.



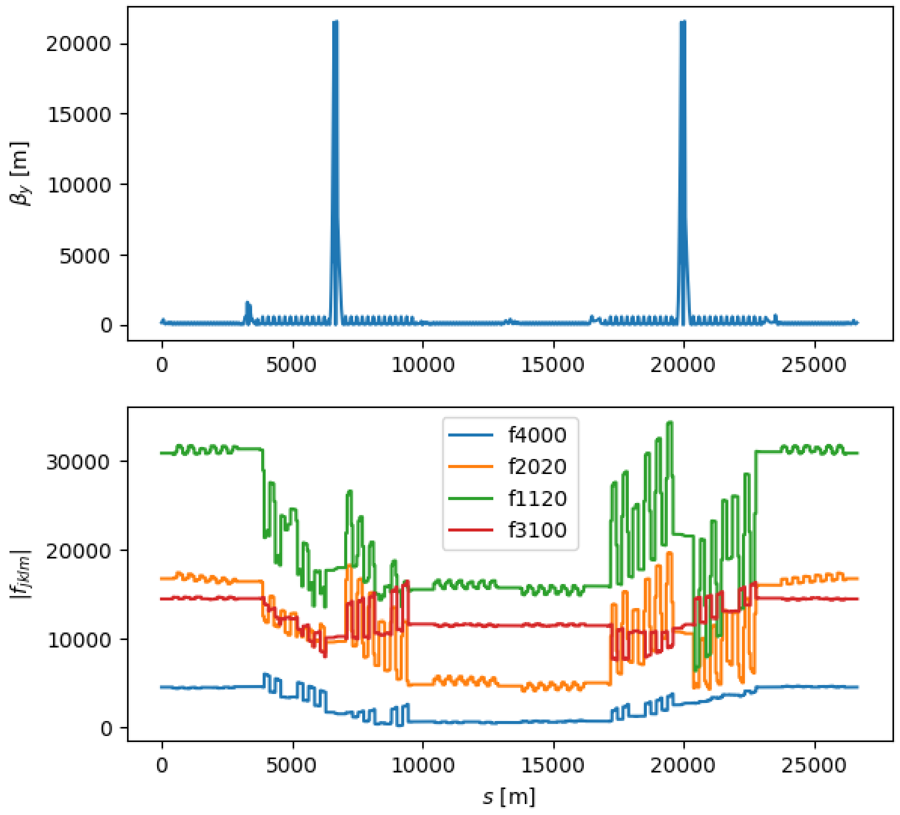
Interfaces to other tools

By design Xsuite is easy to interface and extend using other tools providing a Python API.

Several examples of interfaces providing Xsuite with features that are not natively available:

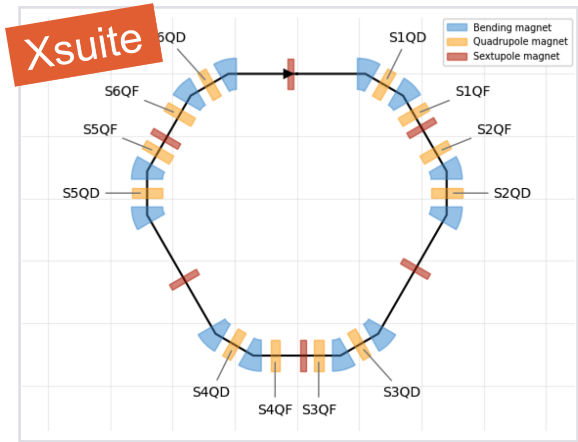
- **MAD-NG** – computation and optimisation of parametric nonlinear normal forms, resonant driving terms, automatic differentiation.
- **RF-Track** – particle tracking in complex field maps.
- **FLUKA** – simulation of particle-matter interaction (e.g. LHC collimation studies).
- **BDSIM-GEANT4** – particle-matter interaction (mostly for FCC).
- **BLonD** – advanced modelling for RF systems.

```python
rdts = ["f4000", "f3100", "f2020", "f1120"]
twiss_ng = hllhc15_line.madng_twiss(rdts=rdts)
```

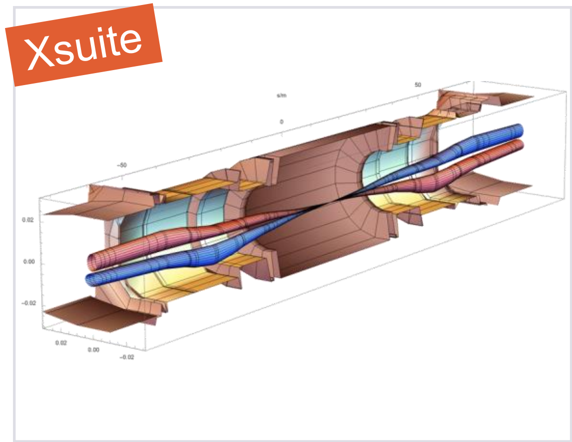


Xsuite adoption

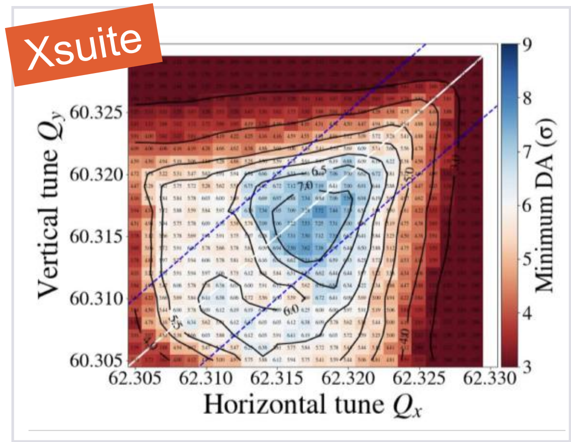
By now Xsuite became a production tool for many beam dynamics studies, allowing development discontinuation for many legacy tools.



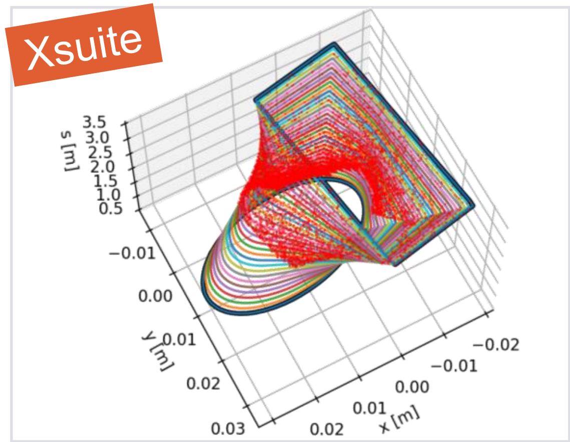
Lattice Design



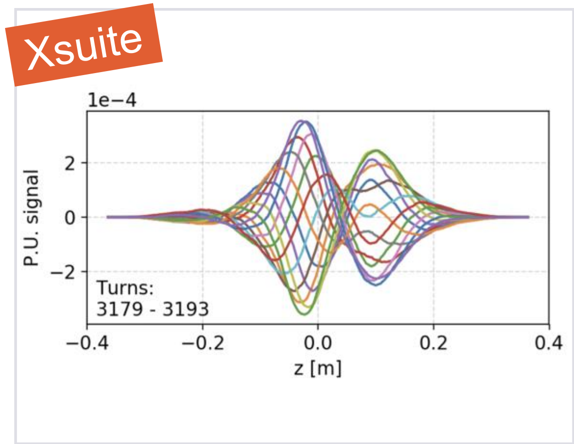
Optics (calculation & design)



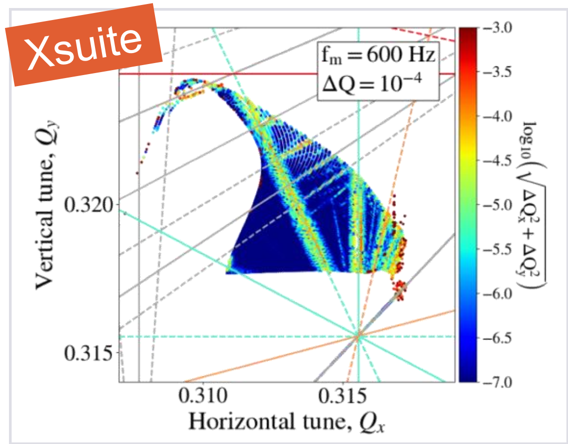
Tracking (dynamic aperture)



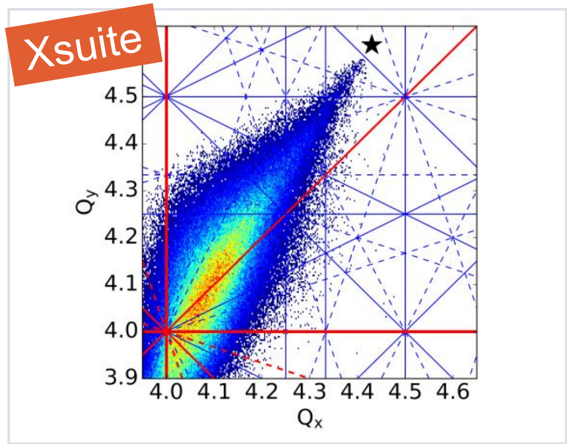
Collimation



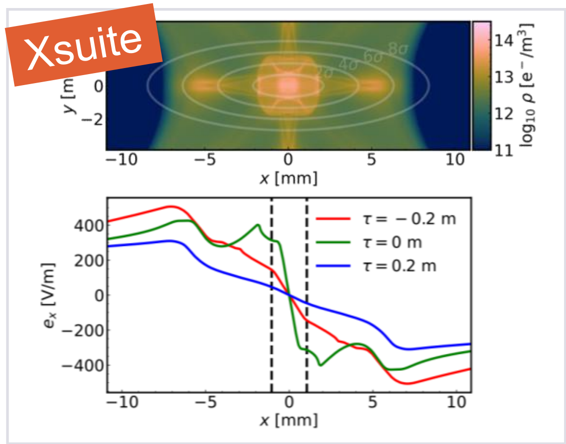
Impedances



Beam-Beam Effects



Space Charge



Electron/Ion Cloud

Lively user base

Users' and developers' response well beyond our expectations!

>30 contributors
>100 active users world-wide
>200 Xsuite mentions in IPAC24 proceedings
Accelerator schools (USPAS, CAS) used Xsuite for tutorials in 2024.

Critical mass matters ⇒ can afford a reasonably-sized team and professional workflows.

CERN
– AD
– ELENA
– LEIR
– PSB
– PS
– SPS, TI2, TI8
– LHC
– FCC-ee, FCC-hh
– Muon Collider

Fermilab
– Main injector
– Recycler
– Booster
– IOTA

GSI
– SIS-18
– SIS-100

BNL
– RHIC
– Booster
– EIC

J-PARC
– Main Ring
– KEK
– SuperKEKB

Medical facilities
– HIT (Heidelberg)
– MEDAUSTRON
– PIMMS
– NIMMS

Light sources/damping rings:
– PETRA
– DESY injector ring
– ELETTRA
– BESSY III
– PSI SLS 2.0
– Canadian Light Source
– CLIC-DR

and more...

Xsuite in action

Xsuite allows loading, exploring, and manipulating accelerator models directly in a Python environment, and launching advanced simulations.

Usage is very dynamic and interactive, as we illustrate in the following example.

Demo code available at https://github.com/xsuite/tutorial_jap24.

jupyter notebook_00_ip_knob Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

[ ]:
```
%matplotlib widget
```

[ ]:
```
import xtrack as xt
import matplotlib.pyplot as plt
```

Load LHC model from acc-models-lhc

[ ]:
```
# Load the sequence and the optics from the acc-model repository
lhc = xt.Multiline.from_json('acc-models-lhc/xsuite/lhc.json')
lhc.vars.load_madx("acc-models-lhc/strengths/ATS_Nominal/2024/ats_30cm.madx")

# Define reference particles
particle_ref = xt.Particles(energy0=6.8e12, mass0=xt.PROTON_MASS_EV)
lhc.b1.particle_ref = particle_ref
lhc.b2.particle_ref = particle_ref

# Cycle starting point to IP7
lhc.b1.cycle('ip7')
lhc.b2.cycle('ip7')

# Twiss default for B2 (use clockwise reference frame)
lhc.b2.twiss_default['reverse'] = True
```

Switch off all orbit bumps

[ ]:
```
lhc['on_disp'] = 0.
lhc['on_xx1_v'] = 0.
lhc['on_x1_v'] = 0.
lhc['on_sep1_h'] = 0.
lhc['on_sep2h'] = 0.
lhc['on_x2v'] = 0.
lhc['on_xx5_h'] = 0.
```

SPS Example

Having the beam dynamics code in the Python ecosystem makes it very natural to combine it with other tools, to manipulate measurement data, interact with the control system, perform numerical optimisation, data analyses, signal processing, etc.

In the following we show an example in which Xsuite was used online during an SPS MD.

Credit to K. Paraschou and H. Bartosik.

Demo code available at https://github.com/xsuite/tutorial_jap24.

jupyter notebook_01_sps_1st_turn_pandas Last Checkpoint: 22 hours ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

[ ]:
```
%matplotlib widget
```

[ ]:
```
import xtrack as xt
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Load and plot SPS BPM data

[ ]:
```
# Use pandas to import YASP file
yasp_file_cols = ['name', 'plane', 'beam', 'pos', 'rms', 'sum', 'hw-status', 'status', 'status-tag']

injection_trajectory_data = pd.read_csv(
    './YASP_injectionTrajectory.txt',
    sep='\s+', skiprows=30, skipfooter=218, engine='python',
    names=yasp_file_cols, index_col='name')

closed_orbit_data = pd.read_csv(
    './YASP_injectionOrbitReference.txt',
    sep='\s+', skiprows=31, skipfooter=218, engine='python',
    names=yasp_file_cols, index_col='name'
)

assert (injection_trajectory_data.index == closed_orbit_data.index).all()

# Keep only horizontal BPMs
injection_trajectory_data = injection_trajectory_data[injection_trajectory_data.plane == 'H']
closed_orbit_data = closed_orbit_data[closed_orbit_data.plane == 'H']
```

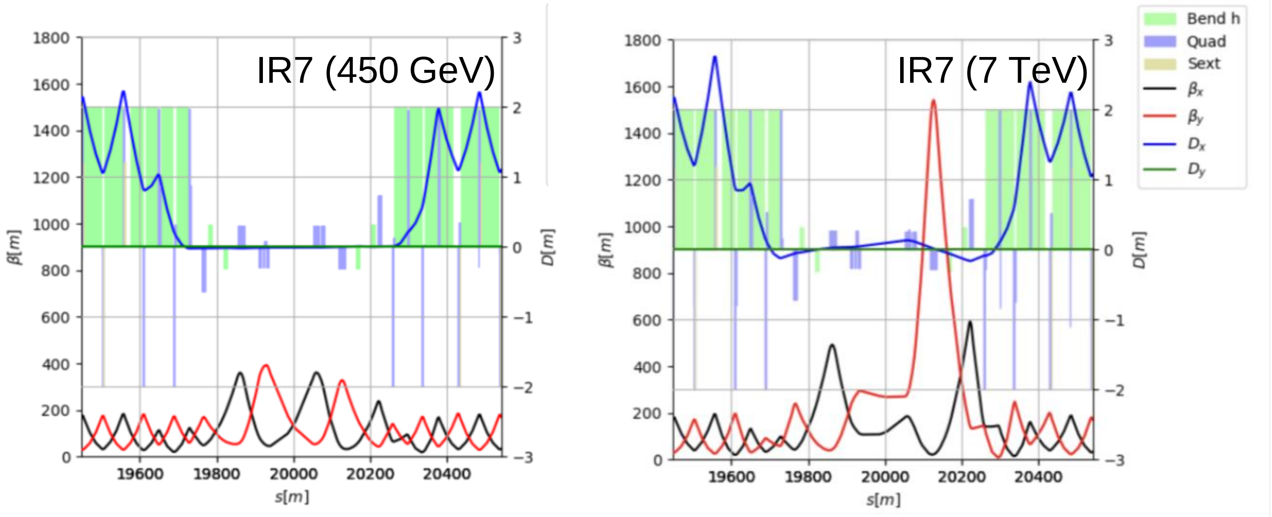
[ ]:
```
# Take difference (1st turn - closed orbit)
bpm_meas = pd.DataFrame({'x_meas': 1e-6 * (injection_trajectory_data['pos'].values - closed_orbit_data['pos'].values)},
                        index=[nn.lower() for nn in injection_trajectory_data.index.values])
```

Xsuite towards Operations

Today operation of the CERN complex (accelerators, transfer lines, experimental areas) uses MAD-X as standard tool for lattice modelling, survey, optics calculations, knob generation, etc.

First tests of Xsuite also in this context went very smoothly.

- Notably, the first full LHC cycle was designed entirely with Xsuite in 2024 and tested in MDs.
- Including optics matching, squeeze functions design, knob design (cross./sep. bumps, tune, chromaticity, coupling, etc.), link with LSA.



Courtesy of R. De Maria and B. Lindstrom

Moving forward: Xsuite towards Operations

Going forward we want to **move to Xsuite for operational tasks**, aiming at not depending on MAD-X for critical applications by the start of Run 4.

Several benefits:

- Advantageous to use the same tool for the present complex and design studies for future machines – development synergies, using same tools makes it more natural for people to work in both environments.
- Overcome shortcomings of MAD-X that today are worked around:
  - e.g., modelling of optics distortion from the PSB injection chicane, reference frame curvature kept independent of bend powering, general handling of tilted magnets.
- Economy of resources: high-quality support for MAD-X is expensive.
  - Requires a very specific skill-set training, not beneficial in other contexts.
  - Plan to reinvest these resources to support users during the migration.

Moving forward: Xsuite towards Operations

Preliminary discussions with main stakeholders (BE-EA, BE-OP, SY-ABT) already took place.

- **Response so far was very positive**
  - General perception that advantages of modern and supported tools outweigh the cost of transition.
- **We will continue engaging with the user community** to get an accurate and complete overview of needs and wishes.
  - Don't hesitate to give Xsuite a try for your applications and reach out to us with feedback.
  - Early feedback is important to ensure that all relevant needs are correctly incorporated in the design.
- The migration of the **'acc-models' repositories** for all accelerators and lines will be an important milestone. BE-ABP is committed to work on this task during LS3 in collaboration with repository owners.

Thank you!

xsuite.web.cern.ch